

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Основы алгоритмизации и программирования»

«К защите допустить»
Руководитель курсового проекта
ассистент кафедры информатики
_____ М.В. Ганусевич
_____._____.2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему:
«СОЗДАНИЕ ТЕКСТОВОЙ RPG (АНАЛОГ DUNGEONS & DRAGONS)»

БГУИР КП 6-05 0612 02 12 ПЗ

Выполнил студент группы 353502
Козелло Владислав Александрович

(подпись студента)

Курсовой проект представлен на
проверку _____._____.2024

(подпись студента)

Минск 2024

СОДЕРЖАНИЕ

Введение.....	5
1 Обзор существующих аналогов.....	6
1.1 Обзор жанра RPG и его особенностей.....	6
1.2 Анализ игры “Dungeons & Dragons”	8
1.2.1 История и концепции игры.....	8
1.2.2 Основная механика игры	9
2 Архитектура игры "Dungeons & Data"	13
2.1 Описание общей структуры игры	13
2.2 Компоненты, обеспечивающие работоспособность программы	13
2.2.1 Модель	13
2.2.2 Контроллер	15
2.2.3 Представление.....	15
3 Теоретическое обоснование разработки программного продукта.....	18
3.1 Обоснование необходимости разработки.....	18
3.2 Технологии и инструменты	19
3.2.1 Язык программирования	19
3.2.2 Библиотеки и фреймворки	21
3.2.3 Среда разработки	23
4 Проектирование функциональных возможностей программы	24
4.1 Анализ алгоритмов	24
4.1.1 Алгоритм управления системой боя.....	24
4.1.2 Алгоритм генерации случайных событий.....	24
4.1.3 Алгоритм обработки выборов и последствий.....	24
4.1.4 Алгоритм прогрессии персонажей.....	24
4.1.5 Алгоритм взаимодействия с неигровыми персонажами	24
4.2 Разработка игрового контента	24
Заключение	25
Список использованных источников	26
Приложение А (обязательное) Листинг программного кода	27

ВВЕДЕНИЕ

В современном мире компьютерные игры являются неотъемлемой частью развлекательной культуры, привлекая к себе миллионы людей. В рамках её ролевые игры (RPG) занимают особое место, предлагая уникальный игровой опыт, позволяющий игрокам погрузиться в виртуальные миры и взаимодействовать с ними.

Целью данного курсового проекта является разработка текстовой RPG, основой которой является настольная ролевая игра "Dungeons & Dragons", предоставляющая игрокам возможность исследовать фантастический мир, вступать в бои с монстрами, развивать своего персонажа и выполнять разнообразные задания. Формат RPG будет текстовый, чтобы уделить особое внимание глубине игрового сюжета, а также стимулированию воображения игроков, как и в оригинальной настольной игре.

Выбор системы "Dungeons & Dragons" в качестве основы обусловлен рядом причин, которые делают её особенно привлекательной для разработки проекта. Наиболее важной особенностью являются правила, которые отличаются гибкостью и расширяемостью, позволяя разработчику адаптировать их под свои нужды и внести собственные изменения, чтобы создать уникальный игровой опыт.

Для достижения поставленной цели предусмотрено следующее:

- 1 Анализ существующих RPG, изучение правил и механик системы "Dungeons & Dragons".
- 2 Проектирование игрового мира, определяя основные его характеристики и особенности.
- 3 Проектирование алгоритмов и механик, реализующих возможности программы.
- 4 Разработка контента для обеспечения максимального погружения в игровой мир, включая диалоги и различные описания.
- 5 Тестирование и анализ результатов курсовой работы.

В результате выполнения данного курсового проекта ожидается создание полноценной игровой среды, которая будет в состоянии захватить внимание игроков и предоставить им уникальный опыт.

Пояснительная записка оформлена в соответствии с СТП 01-2017 [1].

1 ОБЗОР СУЩЕСТВУЮЩИХ АНАЛОГОВ

1.1 Обзор жанра RPG и его особенностей

Ролевая игра (RPG, от английского "Role-Playing Game") – знаменитый жанр компьютерных и настольных игр, где основой игрового процесса является отыгрывание определенной роли. Игрок берет под контроль персонажа, который обладает некоторым набором стандартных характеристик и улучшает их по ходу игры [2].

Многие компьютерные RPG берут своё начало в настольных ролевых играх (НРИ), позаимствовав оттуда во многом терминологию и игровую механику. В НРИ игроки взаимодействуют воображаемыми персонажами в созданном ведущим (мастером игры) мире. Обычно игра проходит за столом, отсюда и название. Мир, в котором происходит игра, история, персонажи и сюжет создаются мастером, который руководит ходом игры и предоставляет игрокам различные сценарии и задания.

Настольные ролевые игры могут быть разнообразными по жанру и тематике. От классических фэнтези и научной фантастики до детективов, приключений в стиле пиратов или постапокалипсиса – выбор огромен. Кроме того, некоторые из них предлагают различные подходы к игровому процессу, например, более сфокусированные на отыгрыше роли или на тактических сражениях.

Основой НРИ являются правила, которые определяют, как происходит взаимодействие игроков с окружающим миром, как решаются конфликты, и как развиваются персонажи. Правила могут быть простыми или сложными, зависит от конкретной игры, но в любом случае, важнее всего – это воображение игроков [3].

Компьютерные ролевые игры – это те же настольные, только роль мастера игры выполняет аппаратное обеспечение вашего компьютера, а воображение игроков чаще всего заменяется на графику. Одни из первых ролевых видеоигр появились в середине 1970-х и представляли из себя текстовые RPG, то есть вся информация, что получал игрок, была в виде текста. Однако несмотря на столь давний срок появления, они актуальны и по сей день. Сейчас найти текстовую игру, оформленную красивыми статичными или анимированными картинками довольно просто (рисунок 1.1) [4].

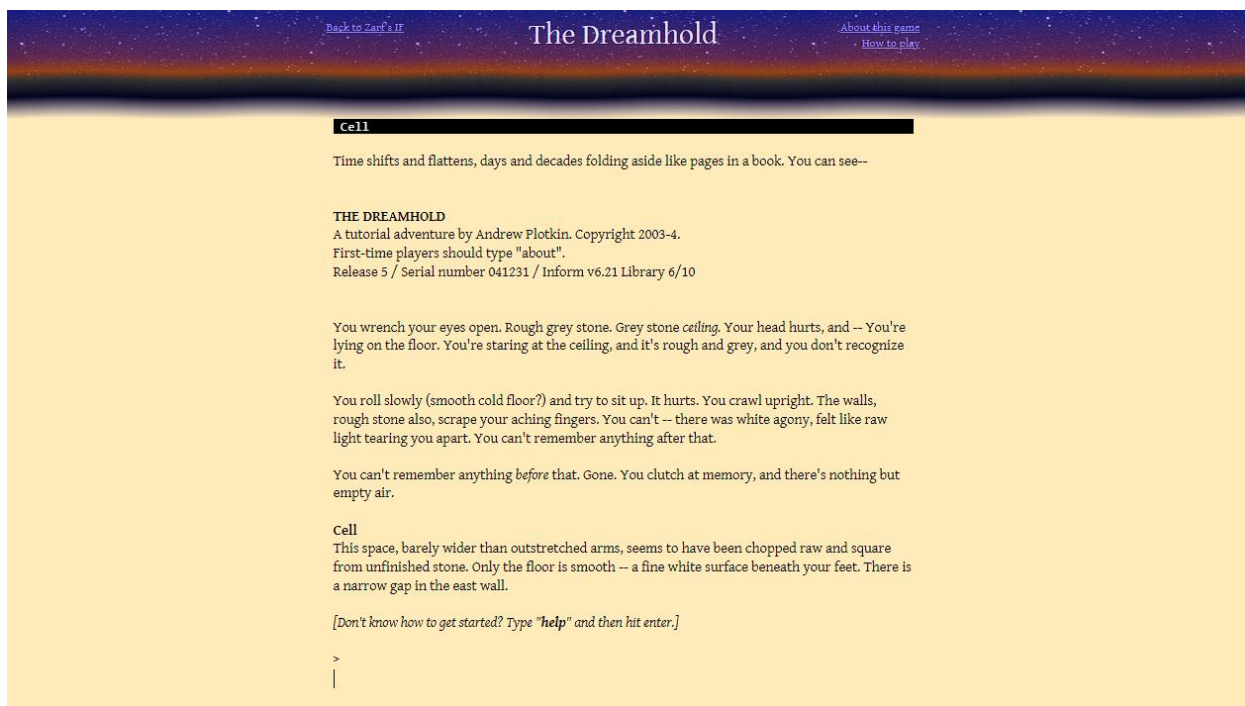


Рисунок 1.1 – Пример текстовой игры (The Dreamhold)

На протяжении всей истории компьютерных ролевых игр возникло достаточно много различных новых поджанров, которые обладают своими особенностями:

1 Настоящая РПГ (CRPG – Computer Role-Playing Game) – это ролевая игра с достаточно большим объёмом диалогов, с хорошо проработанными ролевой системой и игровым миром, свободой действий, а также захватывающим сюжетом.

2 Action RPG (Экшен РПГ) – это игра, где основой игрового процесса выступает не ролевая составляющая, а боевая. Она сложнее, чем в простых RPG, так как чаще всего требует от игрока хорошей реакции или владение определенной боевой системой.

3 Тактические РПГ – RPG, разбавленное стратегией, где игрок контролирует группу (отряд). Развитие всех членов отряда осуществляется по классической для данного жанра схеме.

4 ММОРПГ (Массовая многопользовательская ролевая онлайн-игра) – как и в большинстве ролевых игр, игрок выбирает персонажа, который развивается, выполняя задания и продвигаясь по сюжету. В отличие от одиночной, в ММОРПГ играет очень много игроков одновременно.

5 Roguelike – это популярный поджанр ролевых игр, основой которого является случайная генерация игрового мира, а также необратимая смерть персонажа [2].

Жанр компьютерных ролевых игр сильно укрепился в современном обществе: каждый год выходят десятки различных игр (Baldur's Gate 3, Elden Ring, Starfield и др.), которые вносят новые уникальные механики, графику и атмосферу.

1.2 Анализ игры “Dungeons & Dragons”

1.2.1 История и концепции игры

“Подземелья и Драконы” (Dungeons & Dragons, D&D) – первая коммерческая настольная ролевая игра, созданная Гэри Гигаксом и Дэйвом Арнесоном в США в 1974 году, и до сих пор являющаяся самой лучшей в своей категории. Почти все современные электронные игры чем-либо обязаны ей, в особенности игры жанра RPG. К примеру, “Baldur's Gate” – это компьютерная RPG, мотивом которой стала одноименная книга, написанная для проведения кампании в “Подземелья и Драконы” (рисунок 1.2).

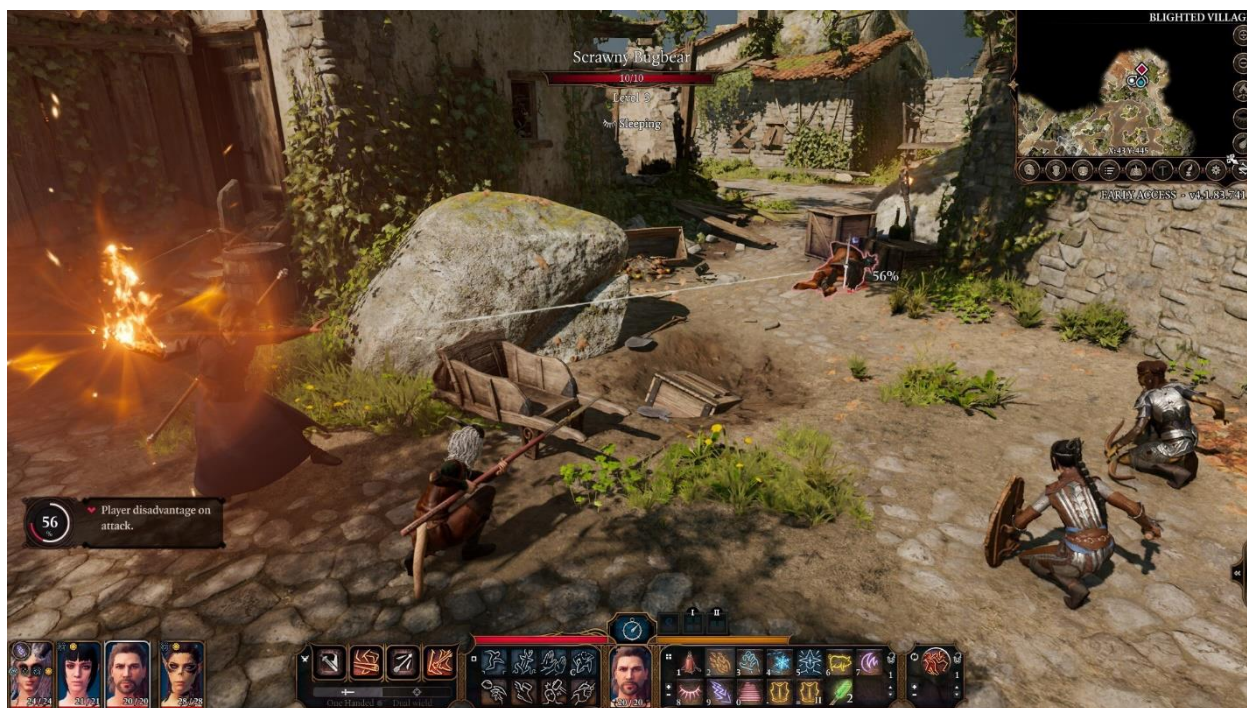


Рисунок 1.2 – Скриншот из компьютерной игры “Baldur's Gate 3”.

D&D основана на правилах, которые обеспечивают рамки для взаимодействия игроков с окружающим миром и друг с другом, но это не значит, что нужно максимально им следовать. Система “Подземелий и драконов” отличается от других игр подобного жанра своей гибкостью:

каждый игрок вправе дополнить или урезать официальные правила (рисунок 1.3) под свои конкретные нужды.



Рисунок 1.3 – Официальный трехтомник правил игры “Dungeons & Dragons”.

Общая концепция представляет собой открытость действий игроков и влияние их последствий на вымышленный фэнтезийный мир, которым управляет мастер игры. "Подземелья" несут в себе свободу действий, отыгрыш роли и полёт фантазии от детских игр в воображаемых героев и импровизационного театра. Но в то же время D&D подобно компьютерным ролевым играм структурирует происходящее правилами для определения результатов действий.

1.2.2 Основная механика игры

Человек, играющий партию в D&D может быть гейм-мастером или обычным игроком. Гейм-мастер ведёт приключение: подготавливает игровой мир, решает его законы и что может произойти в сценарии. Остальные члены группы выбирают себе персонажей и по ходу приключения исследуют мир, борются с монстрами, решают загадки и общаются с другими персонажами. Мастер решает, что происходит в игровом мире в ответ на их действия, а также отыгрывает всех остальных обитателей вселенной. Он действует как арбитр для правил и направляет историю, но сюжет развивается совместными усилиями гейм-мастера и игроков.

Первоначально игроку нужно создать персонажа, которого он будет отыгрывать. Создание заключается в заполнении специального листа, где

будут описаны: класс, раса, характеристики, очки здоровья, способности, умения, инвентарь, мотивация, цель, слабости и др. В процессе игры лист будет дополняться новыми сведениями.

Вся механика D&D строится на случайных исходах бросков игровых костей. В рамках игры существует 6 их разновидностей: к4, к6, к8, к10, к12 и к20. Каждая кость выполняет свою конкретную роль:

1 К20 – основная игровая кость, которая определяет исход заявок на действия от игроков, а также исход попадания атакой во врага. Он имеет 20 граней, на которых изображены числа от 1 до 20.

2 К12 – кость этого типа используется в основном для определения урона от оружия или навыков соответствующих классов персонажей. На гранях кубика находятся числа от 1 до 12.

3 К10 – эта кость позволяет определить результаты, подразумевающие десятичную систему, такие как определение критических попаданий при атаке или урон при накладывании заклинания. Он содержит числа от 0 до 9, которые могут быть умножены на 10 в случае броска двух таких костей, чтобы имитировать бросок к100 (100-гранной кости).

4 К8 – используется для определения урона от определенных навыков и заклинаний, а также для урона от сильного оружия. Кость имеет числа в диапазоне от 1 до 8.

5 К6 – шестигранные кубики используются в D&D наиболее часто. Они участвуют в определении урона от оружия, навыков и заклинаний, а также для определения некоторых результатов и случайных событий. На гранях кубика находятся числа от 1 до 6.

6 К4 – Четырехгранная кость используется в основном для определения незначительных результатов или мелких действий, таких как урон от импровизированного оружия. Кость содержит числа от 1 до 4.

Кубики в Dungeons & Dragons – это не просто инструменты для определения случайных чисел (рисунок 1.4). Они символизируют неопределенность и приключения, которые составляют основу игры. Результаты бросков кубиков могут влиять на ход сюжета и развитие персонажей. Удачные или неудачные броски могут открыть новые возможности или создать препятствия, с которыми игроки должны справиться. Они стали символом самой игры. Игроки могут развивать свои собственные традиции и ритуалы, связанные с бросками кубиков, делая их частью общей культуры D&D.



Рисунок 1.4 – Игральные кости, используемые в системе D&D.

Сражения происходят по правилу "step by step" ("шаг за шагом"). Это означает, что персонажи ходят по очереди, начиная с того, у кого больше выпало значение на к20. За ход игрок может сделать какое-либо основное действие, бонусное действие, перемещение и малое взаимодействие. Ходы персонажей объединяются в раунды, которые длятся 6 секунд игрового времени. Мастер вправе самостоятельно определить, какая заявка от игрока относится к определенному действию, но следует прислушиваться к следующим правилам:

- 1 Основное действие. Действие, которое требует приложения значительных мыслительных или физических усилий, но укладывающееся в 6 секунд. Например, атака, наложение заклинания, помощь, использование предмета и др.

- 2 Бонусное действие. Действие, право на которое дают определенные

умения вашего класса.

3 Перемещение. Этим действием персонаж может переместиться в пространстве на расстояние, равное значению его скорости в футах в секунду умноженное на 6.

4 Малое взаимодействие (часто опускается). Какое-либо незначительное действие, которое можно совершить не задумываясь, и которое не требует больших затрат времени. Например, достать меч из ножен, надеть рюкзак.

Как правило, сражения происходят на специальных боевых полях (рисунок 1.5), но какие-то маленькие стычки гейм-мастер может отыграть и как обычные сцены, где нет четкой последовательности ходов.

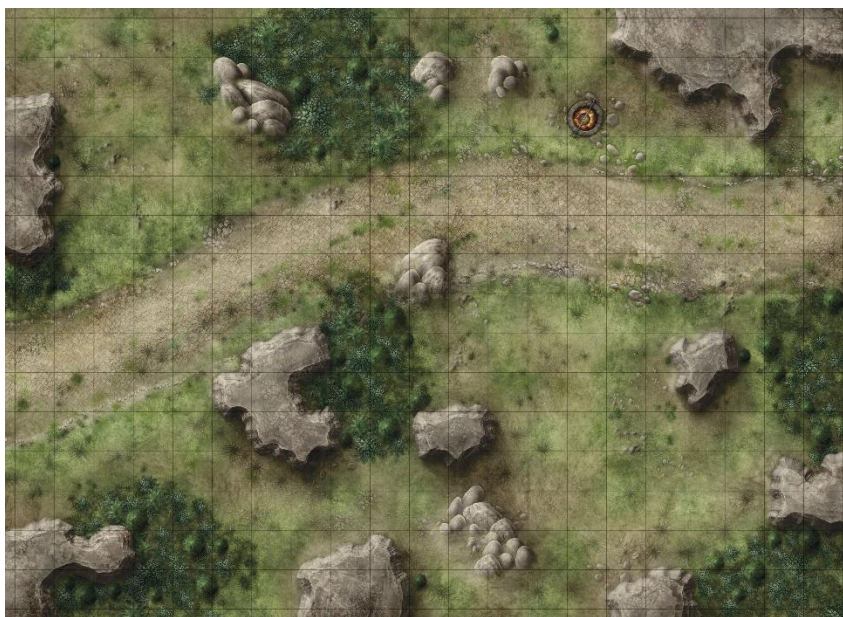


Рисунок 1.5 – Пример боевого поля для игры в D&D.

Вне боевой ситуации персонажи вправе вольно совершать действия, без какой-либо четкой последовательности. В список основных действий, которые могут совершать персонажи, можно включить:

- общение с неигровыми персонажами;
- исследование мира;
- решение головоломок;
- покупка новых предметов и продажа старых.

После заявки на действие от игрока мастер устанавливает сложность броска двадцатигранника (значение от 1 до 20), и если игрок выбросил значение больше заявленного или равное ему, то действие считается успешным [5].

2 АРХИТЕКТУРА ИГРЫ "DUNGEONS & DATA"

2.1 Описание общей структуры игры

"Dungeons & Data" – это текстовая компьютерная ролевая игра, разработанная в качестве данного курсового проекта. Она представляет собой упрощенный аналог описанной выше настольной ролевой игры "Dungeons & Dragons" с сохранением основных её механик, интерпретированных под одного игрока.

Структура приложения имеет вид, типичный для данного жанра компьютерных игр, а именно MVC. MVC (Model-View-Controller) – это архитектурный шаблон, который структурирует приложение вокруг трех основных компонентов: модель, представление и контроллер. Каждый из компонентов предоставляет свою функциональность:

1 Модель: отвечает за управление данными и логикой приложения. Она не зависит от других компонентов и обеспечивает независимый доступ к данным.

2 Представление: отображает данные модели пользователю и обрабатывает пользовательский ввод. Этот компонент получает данные от модели и передает их пользователю в удобном виде, а также передает пользовательский ввод контроллеру.

3 Контроллер: принимает пользовательский ввод из представления, интерпретирует его и обновляет модель и/или представление в соответствии с этим вводом. Контроллер является посредником между представлением и моделью, обеспечивая их взаимодействие.

MVC обеспечивает модульность и независимость компонентов, позволяя изменять каждый из них отдельно без влияния на остальные, что упрощает разработку, тестирование и поддержку приложения.

2.2 Компоненты, обеспечивающие работоспособность программы

2.2.1 Модель

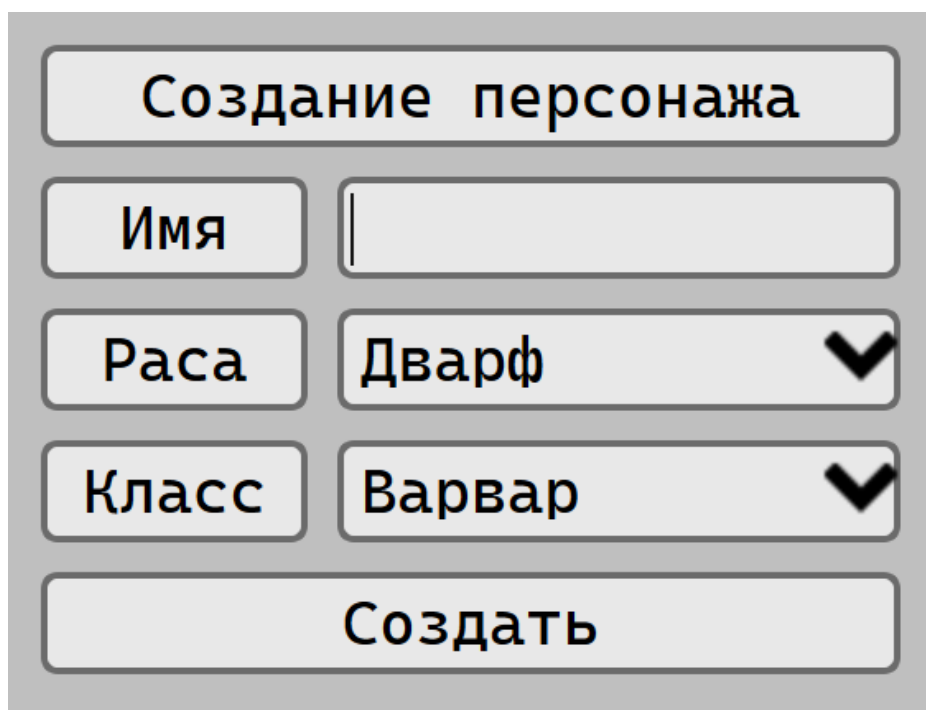
В качестве модели в игре "Dungeons & Data" предусмотрены несколько компонент, отвечающие за свой уникальный функционал:

- персонаж;
- неигровые персонажи;
- диалоги;
- предметы;

– игральные кости.

Персонаж представляет из себя набор данных, являющиеся его характеристиками, а также методами для их управления и изменения. Он является главной моделью, так как взаимодействие со всеми остальными компонентами приложения осуществляется именно через него.

Таким образом, игроки начинают свое знакомство с "Dungeons & Data" именно через форму создания своего персонажа (рисунок 2.1). Пользователь выбирает имя, расу и класс своего героя, тем самым неявно определяя его характеристики. Класс можно выбрать один из следующих: варвар, бард, жрец, друид, воин, монах, паладин, следопыт, плут, чародей, колдун, волшебник. Что касается расы, игроку дается на выбор: дварф, эльф, полурослик, человек, драконо-рожденный, гном, полуэльф, полуорк, тифлинг.



Создание персонажа	
Имя	<input type="text"/>
Раса	Дварф ▼
Класс	Варвар ▼
Создать	

Рисунок 2.1 – Форма создания персонажа.

Модель "Персонаж" обеспечивает весь необходимый функционал для взаимодействия с ним, предлагая все необходимые методы.

Неигровые персонажи (НПС) представляют из себя модели, имеющие практически такой же набор характеристик, как и персонаж, созданный пользователем. Взаимодействие с ними не является прямым, игрок общается с текстовой моделью, представляющей из себя дерево диалога, которое имитирует разговор с вариантами выбора ответа. При решении напасть на НПС в бой вступает уже сама модель неигрового персонажа со своими

характеристиками.

Предметы – это модели вещей, с которыми может взаимодействовать персонаж. Они делятся на 4 вида: безделушки, броня, оружие и зелья. Каждый обеспечивает свой функционал, необходимый для игры.

Игральные кости в качестве модели имитируют бросок настоящих игральных костей, тем самым дают возможность случайного определения исхода действия игрока. Используется в сражениях и взаимодействии с НПС.

2.2.2 Контроллер

В роли контроллера в данном проекте выступают 2 компонента:

- игровой мир;
- боевая система.

Игровой мир – это главный контроллер в игре, который обеспечивает взаимодействие остальных компонент программы с пользовательским интерфейсом. Именно в нем обрабатываются все команды игрока, обновляются модели и вызываются необходимые методы. Весь сюжет игры, взаимодействия, генерация случайных событий, а также боевая система, хранятся в это контроллере.

Боевая система – компонент программы, предоставляющий игрокам возможность вступать в битву с НПС. Битва представляет из себя последовательность раундов, состоящих из ходов игрока и соперника. Ход – это бросок кости на попадание и на урон, который будет нанесен оппоненту. В случае, когда у кого-либо из сражающихся сущностей не останется здоровья, он проигрывает.

2.2.3 Представление

Представление в проекте выражено в виде пользовательского интерфейса. Пользовательский интерфейс – неотъемлемая часть любой компьютерной игры. Он позволяет пользователю взаимодействовать с компонентами приложения, а также получать эстетическое удовольствие от его реализации. Интерфейс игры "Dungeons & Data" включает в себя различные элементы взаимодействия с программой, оформленные в едином стиле, походящим на старину (рисунок 2.2). Среди них можно выделить: меню, кнопки, ярлыки, выпадающие списки, текстовые поля и таблицы. Данные элементы сгруппированы между собой и разбиты на своеобразные блоки, выполняющие определенные функции:

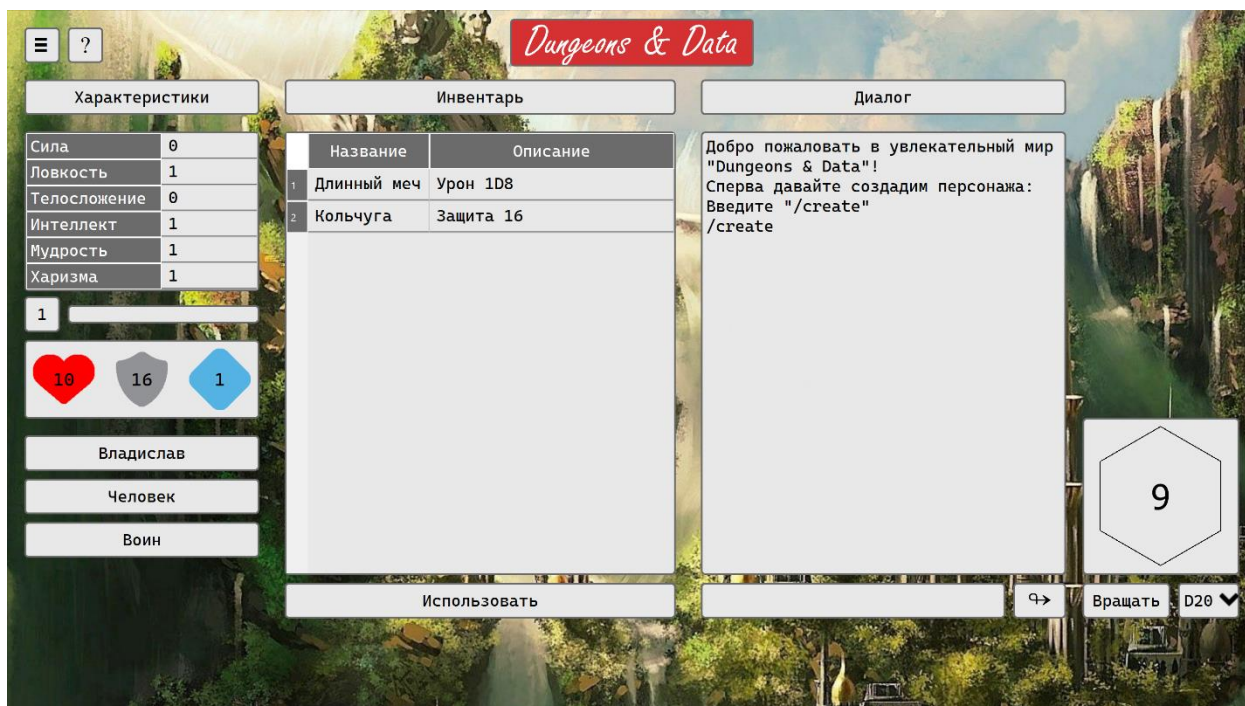


Рисунок 2.2 – Пользовательский интерфейс игры.

1 Меню. Имеет три кнопки, нажав на которые пользователь может выйти из игры, начать новую игру или же продолжить старую (рисунок 2.3).

2 Характеристики персонажа. Включает в себя всю необходимую для игры информацию о персонаже, обновляющуюся автоматически по мере необходимости.

3 Инвентарь. Имеет вид таблицы, где предоставлена информация об вещах, которыми персонаж обладает на данный момент. Также имеется кнопка, нажав на которую можно использовать выбранный предмет.

4 Диалог. Представляет из себя окно, в которое выводится сюжетный текст, а также описываются действия, совершаемые персонажем или НПС. Имеется строка ввода, предназначенная для ввода команды или текста пользователя, и кнопка отправки текста. Является главным источником информации для игрока.

5 Игральная кость. Данный блок состоит из окна вывода графики (имитация вращения кубика), кнопки "Вращать" и выпадающего списка, в котором можно выбрать необходимую кость.



Рисунок 2.3 – Меню игры.

Интерфейс разработан с учетом пользовательского удобства использования, придавая особое внимание созданию интуитивно понятной игровой среды. Предусмотрена кнопка "?", которая выполняет роль подсказки: выводит список всех команд, тем самым помогая пользователю, если тот что-либо забыл.

Также предусмотрен интерфейс улучшения характеристик персонажа, который появляется после повышения уровня (рисунок 2.4).

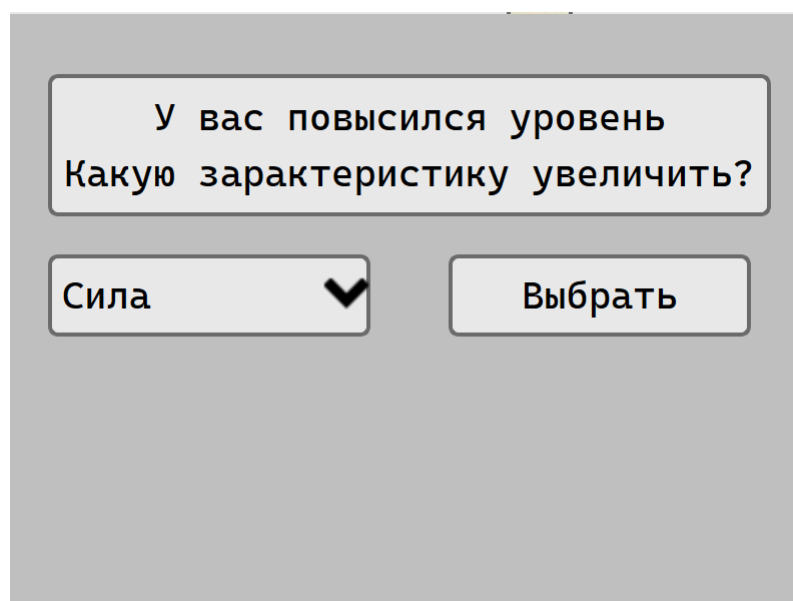


Рисунок 2.4 – Интерфейс улучшения характеристик персонажа.

3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА

3.1 Обоснование необходимости разработки

В наше время индустрия компьютерных игр развивается быстро и постоянно. Новые технологии и платформы приносят удивительные возможности, которые привлекают внимание как взрослых, так и детей, и обеспечивают захватывающий игровой опыт. Это стимулирует разработчиков следовать трендам и постоянно выпускать новые продукты, которые заслужат одобрение общества.

В свою очередь, создание текстовой RPG это явный способ подчеркнуть достоинство жанров компьютерных игр прошлого, которые отличались своей простотой визуализации и своеобразной механикой. А адаптация системы НРИ "Dungeons & Dragons" для одного игрока даст возможность погрузиться в игровой прогресс без вовлечения других людей.

Разработка игры – это креативный и трудоемкий процесс, который требует значительных усилий и ресурсов. В рамках курсового проекта, написание такой программы имеет множество практических преимуществ.

Были выделены следующие выгоды разработки программного продукта:

1 Проектирование игры позволяет применить знания, полученные во время учебы, в реальном проекте. Наряду с проектированием логики программы разработчик создает дизайн игрового процесса, что развивает его креативные способности.

2 Разработка игры является отличным способом улучшить навыки программирования. Изучая новые стези, можно попрактиковаться в разработке сложных программных проектов, требующими более расширенной базы знаний.

3 Разработчики могут ознакомиться с основами управления проектом, такими как планирование, управление ресурсами и сроками, а также контроль качества. Данные навыки очень востребованы в современном мире.

4 Текстовая RPG позволит пользователям получить незабываемый игровой опыт, расслабиться после важных дел, погрузившись в мир компьютерных ролевых игр.

В целом, разработка игры в качестве курсового проекта способствует повышению мотивации и интереса к учебе, поскольку она представляет собой захватывающий проект, который может вдохновить человека к творчеству и самореализации.

3.2 Технологии и инструменты

3.2.1 Язык программирования

Язык программирования, на котором будет реализована система, заслуживает большого внимания, так как погружение в него будет с начала конструирования программы и до самого конца. Поэтому нужно максимально ответственно подойти к его выбору.

Если язык хорошо знаком программистам, они работают более производительнее. Данные, полученные при помощи модели оценки Сосото II, показывают, что программисты, которые долго работают с определенным языком, примерно на 30% более продуктивны, чем программисты, обладающие аналогичным опытом, но на другом языке [6].

Отталкиваясь от этой информации и личных предпочтений, был выбран язык C++ (рисунок 3.1). C++ – компилируемый, статически и нестрого типизированный, высокоуровневый язык программирования. Рассмотрим подробнее каждую его характеристику:

1 Компилируемый язык программирования – язык программирования, исходный код которого преобразуется компилятором в машинный код. В результате компиляции получается исполняемый файл, который выполняется операционной системой.

2 Статическая типизация указывает на то, что тип данных привязывается к объекту при его создании и не может быть изменен после. Это снижает риск ошибки программиста и улучшает читаемость кода.

3 Нестрогая типизация допускает приведение одних типов данных к другим.

4 Высокоуровневый язык программирования – это язык программирования, который обладает высоким уровнем абстракции, что позволяет создавать компьютерные программы более быстро и просто. В таких языках используются конструкции, которые ближе к естественному языку человека, для описания структур данных и операций над ними.

Современный C++ может многое предложить. Многие современные функциональные возможности языка имеют практическую ценность. Они обеспечивают повышенную производительность и надежность, их легко понять и применить, трудно случайно использовать не по назначению [7].

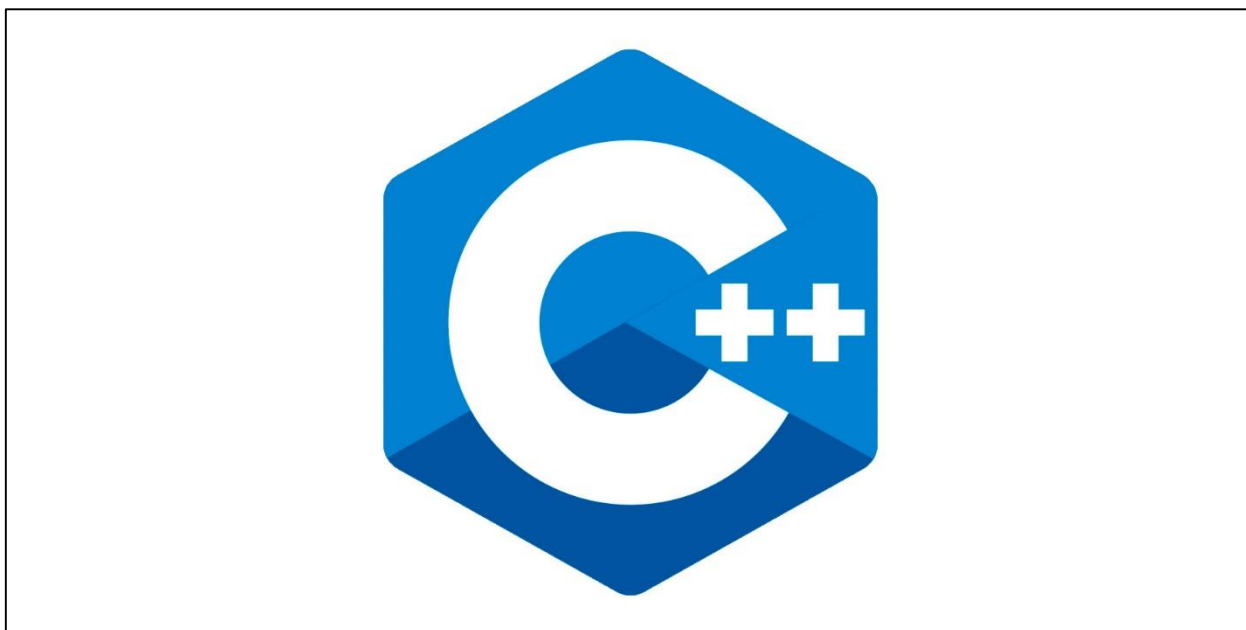


Рисунок 3.1 – Язык программирования C++.

Важной особенностью C++ в сравнении с другими языками программирования является возможность напрямую работать с памятью, что позволяет оптимизировать использование ресурсов и уменьшить накладные расходы, связанные с автоматическим управлением памятью, таким как сборка мусора. Это приводит к улучшению производительности приложения и увеличению скорости выполнения программы.

C++ в купе со всеми его функциональными возможностями является наилучшим выбором языка программирования в рамках данного курсового проекта. Рассмотрим основные причины, почему это так:

1 C++ является высокопроизводительным, следовательно, он позволяет создавать быстрые и эффективные приложения. В игровой разработке производительность играет ключевую роль, поскольку игры должны быть отзывчивыми.

2 C++ один из наиболее популярных и широко используемых языков программирования в игровой индустрии. Это означает, что существует обширное сообщество разработчиков, фреймворков и ресурсов, которые могут помочь в написании проекта.

3 C++ подходит для разработки крупных проектов благодаря своей модульности, возможности организации кода в библиотеки и поддержке пространств имен.

3.2.2 Библиотеки и фреймворки

Фреймворк – заготовка, готовая модель в программировании для быстрой разработки, на основе которой можно написать собственный код. Он задает структуру, определяет правила и предоставляет необходимый набор инструментов для создания проекта [8].

Для реализации игры "Dungeons & Data" был выбран фреймворк Qt версии 6.6.1. Он позволит облегчить работу с графикой и пользовательским интерфейсом, а также привнесет в проект свои особенности.

Qt – это кроссплатформенный фреймворк для разработки программного обеспечения с открытым исходным кодом, который широко используется для создания графических интерфейсов (GUI), приложений сетевого взаимодействия, игр и других программных продуктов (рисунок 3.2).



Рисунок 3.2 – Фреймворк Qt.

Однако не следует принимать Qt как фреймворк только для создания пользовательского интерфейса, так как в отличие от популярной библиотеки SFML он предоставляет большое количество новых возможностей для разработчика, не касающихся интерфейса. Можно выделить следующие особенности Qt:

1 Одним из главных преимуществ Qt является его кроссплатформенность. С помощью этого фреймворка можно создавать приложения, которые могут работать на различных операционных системах, таких как Windows, macOS, Linux, а также на мобильных устройствах под управлением Android и iOS.

2 Qt предоставляет мощные инструменты для создания графических пользовательских интерфейсов. Это включает в себя широкий набор виджетов, стилей и всевозможных средств для их проектирования.

3 Данный фреймворк построен на модульной архитектуре, что позволяет разработчику использовать только те компоненты, которые необходимы для конкретного проекта. Такой подход упрощает разработку и уменьшает размер итогового приложения [9].

Важно упомянуть, что в Qt существует своя иерархия классов, которая обеспечивает удобство и гибкость в разработке приложений. Иерархия базируется на концепции объектно-ориентированного программирования и включает в себя множество базовых классов, от которых наследуются более специфические классы для реализации конкретного функционала.

Одним из ключевых элементов этой иерархии является класс QObject, от которого наследуются практически все классы в Qt (рисунок 3.3). QObject предоставляет базовую функциональность для управления объектами в Qt, такую как система событий, сигналы и слоты, управление памятью и т.д.

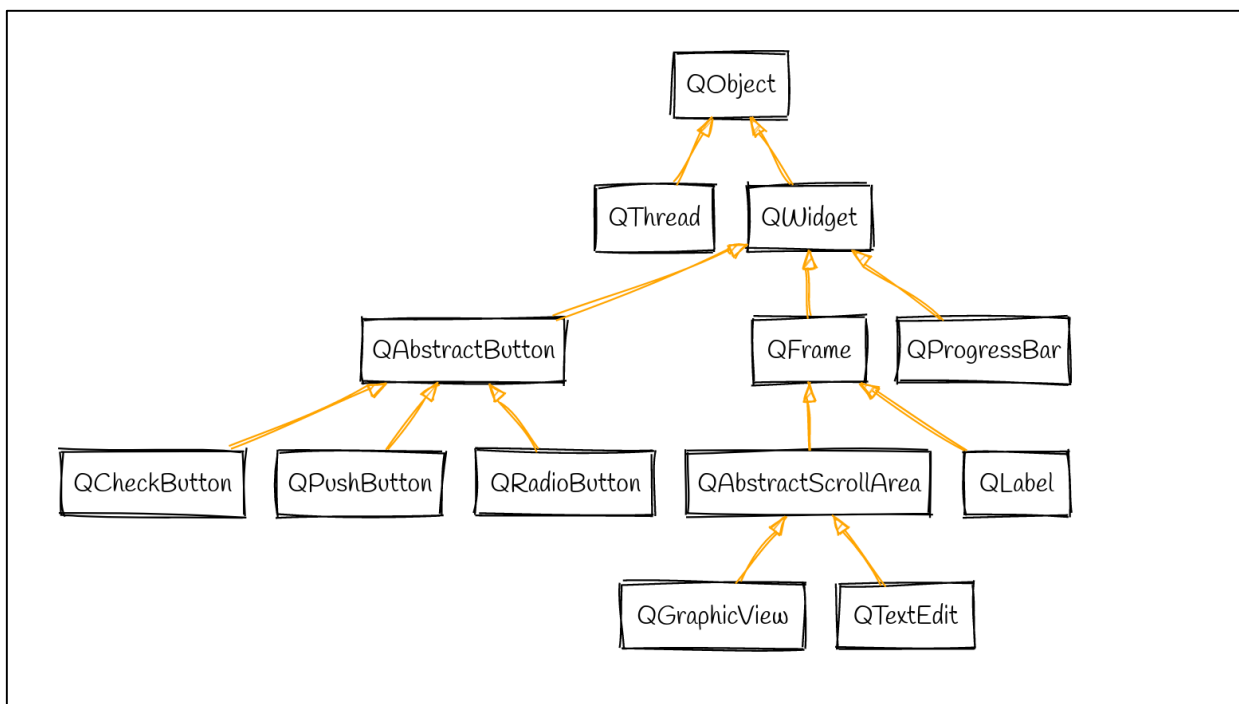


Рисунок 3.3 – Диаграмма наследования некоторых классов в Qt.

3.2.3 Среда разработки

Выбор интегрированной среды разработки (IDE) может существенно повлиять на эффективность и удобство разработки приложений. В связи с тем, что при создании проекта использовался фреймворк Qt, приоритет был отдан в сторону Qt creator.

Данная IDE разработана командой Qt и оптимизирована для работы с вышеупомянутым фреймворком. Это означает, что Qt creator обеспечивает прямую интеграцию с инструментами и библиотеками Qt, что упрощает создание и отладку приложений. Среди таких интеграций важно отметить встроенный дизайнер форм, который позволяет создавать графические интерфейсы приложений методом перетаскивания и редактирования (рисунок 3.4). Это во многом упрощает процесс создания пользовательского интерфейса и исключает необходимость вручную писать код разметки.

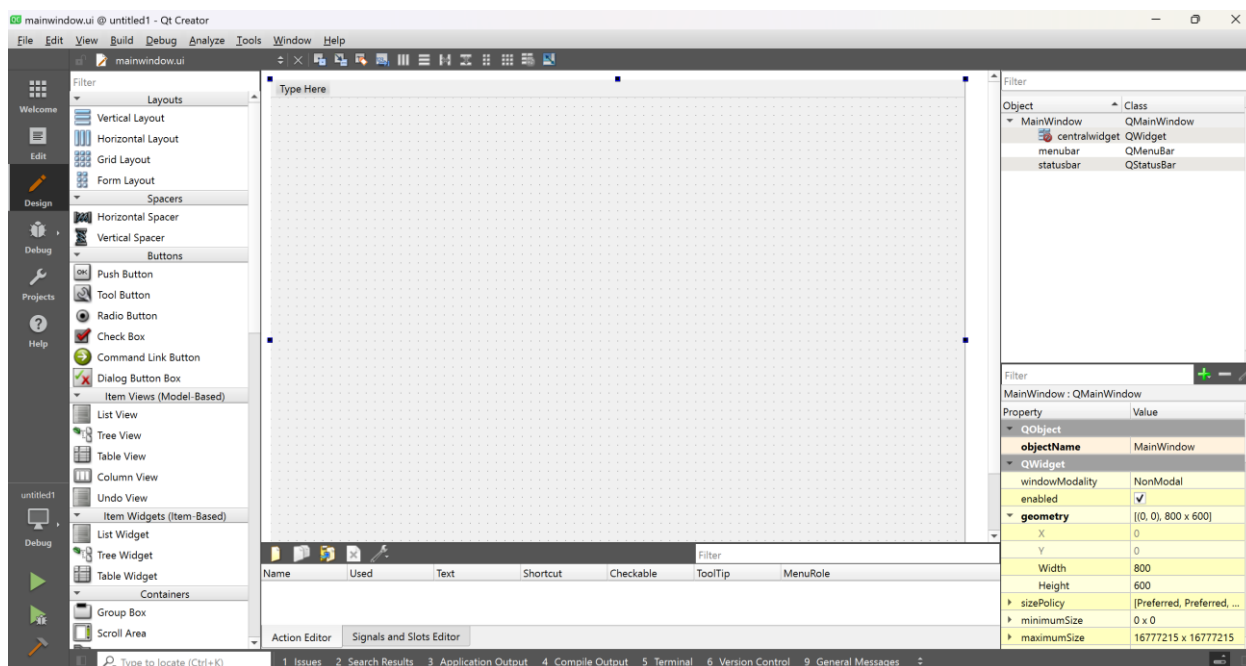


Рисунок 3.4 –Встроенный в Qt creator дизайнер форм.

Пользование Qt Creator облегчает поиск ошибок в коде, ведь он поставляется с мощным отладчиком, который поддерживает различные возможности проверки: точки останова, просмотр переменных, стека вызовов и другие [10].

4 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ПРОГРАММЫ

4.1 Анализ алгоритмов

4.1.1 Алгоритм управления системой боя

4.1.2 Алгоритм генерации случайных событий

4.1.3 Алгоритм обработки выборов и последствий

4.1.4 Алгоритм прогрессии персонажей

4.1.5 Алгоритм взаимодействия с неигровыми персонажами

4.2 Разработка игрового контента

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Доманов, А. Т. Стандарт предприятия / А. Т. Доманов, Н. И. Сорока. – Минск : БГУИР, 2017. – 167 с.
- [2] Что такое РПГ [Электронный ресурс]. – Режим доступа: <https://igrasan.ru/chto-takoe-rpg/>. – Дата доступа: 18.04.2024.
- [3] Настольные ролевые игры [Электронный ресурс]. – Режим доступа: https://gamesisart.ru/theory_nri_kk.html. – Дата доступа: 18.04.2024.
- [4] Текстовые игры [Электронный ресурс]. – Режим доступа: <https://cubiq.ru/luchshie-tekstovye-igry/>. – Дата доступа: 18.04.2024.
- [5] Книга игрока / Дж. Кроуфорд [и др.]. – США : Hasbro SA, 2014. –
- [6] Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – СПб. : БХВ, 2024. – 59 с.
- [7] Современный C++: безопасное использование / Дж. Лакос [и др.]. – М. : ДМК ПРЕСС, 2023. – 32 с.
- [8] Фреймворк: что это такое и для чего нужен [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/glossary/framework/>. – Дата доступа: 20.04.2024.
- [9] About Qt - Qt Wiki [Электронный ресурс]. – Режим доступа: https://wiki.qt.io/About_Qt. – Дата доступа: 20.04.2024.
- [10] Qt Creator [Электронный ресурс]. – Режим доступа: <https://www.qt.io/product/development-tools>. – Дата доступа: 20.04.2024.

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг программного кода

GitHub: <https://github.com/Vladk-jpg/Dnd>.