

Приложение А «Диаграмма вариантов использования».....	27
Приложение Б «Структура игрового приложения».....	29
Приложение В «Диаграмма последовательности».....	31
Приложение Г «Диаграмма деятельности».....	33
Приложение Д «Функциональная модель».....	35
Приложение Ж «Модель данных».....	37
Приложение И «Диаграмма классов».....	39
Приложение К «Диаграмма объектов».....	41
Приложение Л «UX и UI прототипирование».....	43
Приложение М «Тест-кейсы».....	45

## Введение

Целью данного техно рабочего проекта по предмету ТРПО является разработка и проектирование программного продукта – игрового приложения «Elven Illusion of Time». Данный программный продукт позволит игрокам осознать значение времени и научиться эффективнее его управлять. Игра покажет, как важно бережно относиться ко времени, использовать его с умом и планировать свою жизнь грамотно. Этот игровой опыт поможет развить навыки тайм-менеджмента, сделает пользователей более дисциплинированными и организованными. В процессе игры игроки познакомятся с различными методами планирования, а также узнают много нового о принципах управления временем.

Далее приведено краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем можно ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, функциональные и нефункциональные требования к программному продукту. Все входные, выходные и условно-постоянные данные тоже будут описаны в первом разделе. В подразделе «Инструменты разработки» рассмотрена среда, в которой создается данный программный продукт. Здесь также установлены минимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей. В анализе задачи описана диаграмма вариантов использования, процесс выбора модели жизненного цикла программного обеспечения.

В разделе «Проектирование задачи» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе четко описан пользовательский интерфейс с помощью UX дизайна.

«Реализация» – это третий раздел пояснительной записки, в котором описываются элементы и объекты, которые использованы при реализации данного приложения. В этом разделе четко описаны функции пользователя и их структура.

Четвертый раздел – «Тестирование». В нем описано функциональное тестирование данной программы, смоделированы все основные действия пользователя при работе с программой.

В разделе «Руководство пользователя» описано назначение, область применения, среда функционирования разработанной программы. Также в нем описано использование справочной системы.

«Заключение» содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В разделе «Список использованных источников» приведен список используемых при разработке источников.

В приложении А предоставлена диаграмма вариантов использования.

В приложении Б предоставлена структура игрового приложения.

В приложении В предоставлена диаграмма последовательности.

В приложении Г предоставлена диаграмма деятельности.

В приложении Д предоставлена функциональная модель.

В приложении Ж предоставлена модель данных.

В приложении И предоставлена диаграмма классов.

В приложении К предоставлена диаграмма объектов.

В приложении Л предоставлены UX и UI прототипирование.

В приложении М предоставлены тест-кейсы.

					ТЗ 2-40 01 01.35.40.12.24	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

## **1 Анализ задачи**

### **1.1 Постановка задачи**

#### **1.1.1 Организационно-экономическая сущность задачи**

Наименование задачи: Игровое приложение «Elven Illusion of Time»

Цель разработки: Создание видеоигры для учения ценить время.

Назначение: ПП предназначен для пользователей, желающих осознать ценность времени и научиться его эффективно использовать.

Источники и способы получения данных: Данные будут получаться от пользователей через внутриигровые действия, их взаимодействие с миром и обратную связь.

На данный момент абсолютных аналогов данного ПП не найдено.

#### **1.1.2 Функциональные требования**

- Гость имеет данные права доступа:
- Просмотр главной страницы и обзора игры;
- просмотр туториалов и обучающих материалов;
- регистрация в игре;
- просмотр отзывов и рейтингов игры;
- просмотр информации о разработчиках игры и их истории.
- Пользователь имеет данные права доступа:
- Вход в учетную запись;
- доступ к игровому процессу и выполнение заданий;
- право давать комментарии на форуме;
- просмотр раздела “Часто задаваемые вопросы”;
- получение новостей и обновлений игры;
- отправка обратной связи и предложений;
- отслеживание своего прогресса и достижений;
- выход из учетной записи;
- имеет права «Гость».
- Администратор данные права доступа:
- Вход в административный интерфейс;
- ответы на обратную связь от игроков;
- публикация новостей;
- имеет права «Пользователь».

### 1.1.3 Описание входной, выходной и условно-постоянной информации

Данные пользователей: возраст, пол, предпочтения, уровень сложности, выбранный пользователем, история игры пользователя. Эти данные необходимы для персонализации игрового опыта и адаптации сложности игры под каждого пользователя. Профиль пользователя и настройки игры: информация о профиле пользователя, включая персональные данные, настройки игры, такие как предпочтительный режим игры, графические настройки и звуковые параметры. Это помогает улучшить пользовательский опыт и сделать игру более комфортной. Пользователи и разработчики игр: информация о пользователях, которые играют в игру, а также о разработчиках, которые поддерживают и развивают проект.

Эта информация представлена в различных формах:

В форме регистрации: пользователи предоставляют свои данные (возраст, пол, предпочтения) при регистрации на игровой платформе. Регистрация необходима для создания персональных аккаунтов и доступа к игровому контенту. В форме профиля пользователя и настроек игры: пользователи могут настраивать свой профиль и параметры игры, чтобы сделать игровой процесс более удобным и персонализированным.

Комментарии: комментарии могут быть оставлены пользователями к различным постам, задачам или событиям в игре. Комментарии помогают обсуждать достижения, делиться опытом и предлагать идеи по улучшению управления временем. Блоки с последними новостями: это секции на игровой платформе, где отображаются последние новости, связанные с игрой и платформой. Здесь могут быть представлены обновления игры, анонсы новых функций, события или другая актуальная информация, связанная с темой ценности времени. Новостные ленты: новостные ленты могут содержать разнообразную информацию о событиях в игре, актуальных темах и новостях, которые могут быть интересны пользователям. В этих лентах пользователи могут следить за своим прогрессом и узнавать о новых возможностях игры.

Перечень условно-постоянной информации: правила игры, описание уровней, параметры персонажей.

### 1.1.4 Нефункциональные (эксплуатационные) требования

– Требования к применению: игра предоставляет возможность быстро начать игру, сохранить прогресс и получить обратную связь от системы, а также возможность узнать о статусе текущих заданий и достижений;

– требования к реализации: игра будет разработана с использованием современных технологий, таких как Unity для обеспечения высокого качества графики и производительности. Продукт поддерживает однопользовательский режим;

– требования к надежности: игра будет стабильной и надежной, с минимальными перебоями в работе. Будет предусмотрена возможность резервного копирования и восстановления данных игрока;

– требования к интерфейсу: игра будет обладать интуитивно понятным и удобным интерфейсом. Будут предоставлены различные настройки для адаптации игры под предпочтения пользователя. Игра будет поддерживать различные языки интерфейса. Продукт будет иметь возможность получения помощи или подсказок внутри игры.

					ТЗ 2-40 01 01.35.40.12.24	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

## 1.2 Диаграмма вариантов использования

Диаграмма вариантов использования – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

В разрабатываемом проекте можно выделить трех актёров: гость, пользователь, администратор. Для гостя доступны следующие варианты использования: просмотр главной страницы и обзоров игры, просмотр tutorиалов и обучающих материалов, регистрация в игре, просмотр отзывов и рейтингов в игре, просмотр информации о разработчиках игры и их истории.

Для пользователя доступны следующие варианты использования: вход в учетную запись, доступ к игровому процессу и выполнение заданий, просмотр раздела «Часто задаваемые вопросы», получение новостей и обновлений игры отправка обратной связи и предложений, отслеживание своего прогресса и достижений, выход из учетной записи, а также все права имеющиеся у «Гостя».

Для администратора доступны следующие варианты использования: Вход в административный интерфейс, ответы на обратную связь, управление игровым контентом, управление учетными записями пользователей, отслеживание статистики игры и поведения игроков, разрешение технических проблем и обслуживание серверов, обновление игры и внедрение новых функций.

Разработанная диаграмма вариантов использования представлена в приложении А.

## 1.3 Выбор модели, метода и подхода разработки программы.

С помощью нескольких таблиц и ответов на вопросы возможно выяснить модель жизненного цикла программного обеспечения.

Выбор модели жизненного цикла на основе характеристик требований находится в таблице 1.

Таблица 1 – Выбор модели жизненного цикла на основе характеристик требований

Критерии категории требований	Каскадная	V-образная	RAD	Инкрементная	Быстропрототипирования	Эволюционная
1	2	3	4	5	6	7
Являются ли требования к проекту легко определяемыми и реализуемыми?	Да	Да	Да			
Могут ли требования быть сформулированы в начале ЖЦ?	Да	Да	Да	Да		

1	2	3	4	5	6	7
Часто ли будут изменяться требования на протяжении ЖЦ?	Нет	Нет	Нет	Нет		
Нужно ли демонстрировать требования с целью их определения?			Да		Да	Да
Требуется ли проверка концепции программного средства или системы?	Нет	Нет		Нет		
Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ?				Да	Да	Да
Нужно ли реализовать основные требования на ранних этапах разработки?			Да	Да	Да	Да
Итого	4	4	5	5	3	3

Вычисления: 4 за каскадную, 4 за V-образную, 5 за RAD, 5 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: на основе результатов заполнения таблицы 1 подходящей является RAD модель и инкрементная модель.

Выбор модели жизненного цикла на основе характеристик команды разработчиков находится в таблице 2.

Таблица 2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

Критерии категории команды разработчиков проекта	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7
Являются ли проблемы предметной области проекта новыми для большинства разработчиков?					Да	Да
Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков?	Да	Да				Да
Изменяются ли роли участников проекта на протяжении ЖЦ?				Да	Да	Да
Является ли структура процесса разработки более значимой для разработчиков, чем гибкость?	Да	Да		Да		
Важна ли легкость распределения человеческих ресурсов проекта?	Да	Да	Да	Да		
Приемлет ли команда разработчиков оценки, проверки, стадии разработки?	Да	Да		Да	Да	Да
Итого	4	4	1	4	3	4



Вычисления: 4 за каскадную, 4 за V-образную, 1 за RAD, 4 за инкрементную, 3 за быстрого прототипирования и 4 за эволюционную.

Итог: на основе результатов заполнения таблицы 2 подходящими являются каскадная, V-образная, инкрементная и эволюционная модели.

Выбор модели жизненного цикла на основе характеристик коллектива пользователей находится в таблице 3.

Таблица 3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

Критерии категории коллектива пользователей	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7
Будет ли присутствие пользователей ограничено в ЖЦ разработки?	Да	Да		Да		Да
Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки?	Нет	Нет	Нет			
Будут ли пользователи вовлечены во все фазы ЖЦ разработки?	Нет	Нет		Нет		Нет
Будет ли заказчик отслеживать ход выполнения проекта?					Да	Да
Итого	3	3	1	2	1	3

Вычисления: 3 за каскадную, 3 за V-образную, 1 за RAD, 2 за инкрементную, 1 за быстрого прототипирования и 3 за эволюционную.

Итог: на основе результатов заполнения таблицы 3 подходящей являются каскадная, V-образная и эволюционная модели.

Выбор модели жизненного цикла на основе характеристик типа проектов и рисков находится в таблице 4.

Таблица 4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

Критерии категории типов проекта и рисков	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7
Разрабатывается ли в проекте продукт нового для организации направления?				Да	Да	Да
Будет ли проект являться расширением существующей системы?					Нет	Нет

1	2	3	4	5	6	7
Будет ли проект крупно- или среднemasштабным?				Да	Да	Да
Ожидается ли длительная эксплуатация продукта?	Да	Да		Да		Да
Необходим ли высокий уровень надежности продукта проекта?		Да		Да		Да
Предполагается ли эволюция продукта проекта в течение ЖЦ?				Да	Да	Да
Велика ли вероятность изменения системы (продукта) на этапе сопровождения?	Нет	Нет	Нет			
Является ли график сжатым?			Да	Да	Да	Да
Предполагается ли повторное использование компонентов?	Нет	Нет				
Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)?	Нет	Нет	Нет	Нет		
Итого	4	5	3	7	5	7

Вычисления: 4 за каскадную, 5 за V-образную, 3 за RAD, 7 за инкрементную, 5 за быстрого прототипирования и 7 за эволюционную.

Итог: на основе результатов заполнения таблицы 4 подходящей является инкрементная и эволюционная модели.

Подведение итогов со всех предыдущих таблиц представлено в таблице 5.

Таблица 5 – Подведение итогов со всех предыдущих таблиц

№ таблиц	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	4	4	5	5	3	3
2	4	4	1	4	3	4
3	3	3	1	2	1	3
4	4	5	3	7	5	7
Итого	15	16	10	18	12	17

Общий итог: наиболее подходящей является инкрементная.

## 1.4 Инструменты разработки

Для разработки данного проекта была выбрана среда Unity, которая является межплатформенной средой для создания и разработки видеоигр и приложений, разработанной компанией Unity Technologies.

Разработка будет производиться на языке программирования C# – многофункциональный язык программирования, активно развивающийся на данный момент. Он часто используется как первый язык программирования начинающих программистов, так и для реализации крупных коммерческих проектов.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

- WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;

- Microsoft Office Word 2021 – для написания документации к программному продукту;

- Figma – будет использоваться для создания UX/UI макетов проекта;

- Xmind – для создания структуры сайта;

При разработке данного программного продукта был использован компьютер со следующими характеристиками:

- Процессор AMD Ryzen 7 5700H;

- видеокарта NVIDIA GEFORCE RTX 4050 6Gb;

- ОЗУ: 16Gb;

- память: HDD 512Gb;

- ОС: Windows 10.

## **2 Проектирование задачи**

### **2.1 Разработка системы меню**

Эффективная структура главного меню и продуманная система навигации являются основой удобства использования игрового ресурса и влияют на впечатление пользователей о проекте. Грамотная организация игровых меню и навигационных элементов помогает игрокам быстро находить нужную информацию, легко перемещаться между уровнями и заданиями, а также улучшает восприятие контента.

На этапе разработки структуры главного меню важно учитывать потребности целевой аудитории, принципы юзабилити, а также технические особенности проекта, чтобы создать логичную и интуитивно понятную навигацию. В этом разделе рассмотрены этапы проектирования структуры игры, основные подходы к организации игровых меню и варианты навигационных решений, направленных на улучшение взаимодействия игроков с игрой.

Полную структуру игрового приложения можно рассмотреть в приложении Б.

### **2.2 Разработка UML-диаграмм**

#### **2.2.1 Диаграмма последовательности**

В первом разделе «1.1 Постановка задачи» были описаны функциональные требования к проекту. На их основе построена диаграмма последовательности. Диаграмма последовательности – это диаграмма, предназначенная для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними. Графические диаграммы последовательности имеют два представления. Одно – слева направо – в виде вертикальных линий, которые соответствуют линии жизни отдельного участка взаимодействия. Второе – вертикальная временная ось, направленная сверху вниз. На диаграмме отображен процесс регистрации.

Диаграмму последовательности можно рассмотреть в приложении В.

#### **2.2.2 Диаграмма деятельности**

Диаграмма деятельности – это диаграмма, предназначенная для моделирования динамических аспектов системы, описывая последовательность выполняемых действий и поток управления между ними. Она помогает визуализировать логику выполнения процесса или сценария, показывая, как

действия связаны друг с другом и каким образом происходит переход от одного действия к другому. На диаграмме отображен процесс оформления заказа на аренду техники, включая шаги и возможные развилки в процессе.

Диаграмму деятельности рассмотрим в приложении Г.

### 2.2.3 Функциональная модель

Функциональная модель представляет собой описание функциональности системы. Она включает в себя множество диаграмм и моделей, таких как диаграммы потоков данных (DFD), которые показывают, как информация движется через систему и как она обрабатывается. Основная цель функциональной модели - показать, как система выполняет свои функции и взаимодействует с внешними компонентами и пользователями. Это помогает идентифицировать и устранить потенциальные проблемы на ранних стадиях разработки.

Функциональную модель рассмотрим в приложении Д.

### 2.2.4 Модель данных

Модель данных описывает, как данные организованы и хранятся в системе. Она включает схемы баз данных, связи между таблицами, а также правила, управляющие данными. Модель данных важна для обеспечения целостности и консистентности данных в приложении. Она помогает разработчикам понять, как данные должны быть структурированы, и обеспечивает основу для взаимодействия с базами данных. Модель данных также играет ключевую роль в оптимизации производительности и масштабируемости системы.

Модель данных рассмотрим в приложении Ж.

### 2.2.5 Диаграмма классов

Диаграмма классов — это диаграмма, используемая в объектно-ориентированном программировании для визуализации структуры системы. Она показывает классы, их атрибуты, методы и взаимоотношения между ними. Диаграммы классов позволяют разработчикам увидеть, как компоненты системы взаимосвязаны, что помогает в планировании и реализации программного обеспечения. Это важный инструмент для понимания и документирования архитектуры приложения, а также для координации командной работы.

Диаграмму классов рассмотрим в приложении И.

### 2.2.6 Диаграмма объектов

Диаграмма объектов – это диаграмма, предназначенная для моделирования статических связей между объектами системы в конкретный момент времени. В отличие от диаграммы классов, которая отображает классы и их связи в общем виде, диаграмма объектов фокусируется на конкретных экземплярах классов и их связях, показывая состояние системы в определенной ситуации. Связи между объектами показывают, как взаимодействуют конкретные экземпляры классов в ходе выполнения сценария. Диаграмму объектов рассмотрим в приложении К.

### 2.3 Разработка пользовательского интерфейса

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемой игры.

Для разработки визуального дизайна использовались сдержанные, мягкие цвета для удобства использования программного продукта. В ходе разработки были спроектированы дизайны всех страниц игры, как видимых пользователю, так и невидимых. Ранее разработанная структура главного меню расположена на рисунке 1.

Для организации эффективной работы пользователя нужно создать целостный программный продукт данной предметной области, в котором все компоненты будут сгруппированы по функциональному назначению. Все исходные данные будут разделены на несколько групп.

Прототип – это наглядная модель пользовательского интерфейса. В сущности, это «черновик», созданный на основе представления разработчика о потребностях пользователя. Итоговое отображение программы может отличаться от прототипа.

Прототип UX представлен в приложении Б, ссылка на UX: <https://www.figma.com/design/p0K1T5Xd0auwcm93NSB5YT/ЛБ17?node-id=0-1&p=f&t=FlxsIgD9LAg6ShDr-0>

Прототип UI представлен в приложении Б.

### 3 Реализация

#### 3.1 Руководство программиста

##### 3.1.1 Организация данных

Основным средством хранения информации в приложении являются: png-изображения для хранения фоновых изображений, изображений игрового персонажа, кнопок. В проекте также присутствуют текстовые файлы для хранения прогресса прохождения.

Основными функциями приложения являются:

- Запуск игрового приложения;
- начало игры с помощью кнопки «Играть»;
- настройка параметров игры через кнопку «Настройки»;
- просмотр актуальных новостей и обновлений посредством кнопки «Новости»;
- выход из приложения нажатием кнопки «Выход»;
- передвижение внутри игры с использованием клавиш «A», «D», «W», «S»;
- приостановка игры через кнопку «Пауза»;
- сохранение текущего прогресса и выход из игры через опцию «Сохранить и выйти»;
- сохранение без выхода из игры с помощью кнопки «Сохранить»;
- продолжение игры после паузы через кнопку «Продолжить».

##### 3.1.2 Структура программы

Данный проект содержит 2 сцены, рассмотрим их назначение:

сцена «\_MainMenu» – главное меню приложения;

сцена «Scene1» – игровая сцена игры;

Также проект содержит скрипты, которые отвечают за определённые функции, рассмотрим их назначение:

скрипт «\_MainMenu» закреплён за пустым объектом «MenuManager», расположенным на сцене «\_MainMenu» отвечает за кнопки в главном меню;

скрипт «GameUIManager» закреплён за кнопкой «Save and Exit Btn» на сцене «Scene1» и выполняет функцию перехода на главное меню, сцену «\_MainMenu», ;

скрипт «Movement» закреплён за объектом «Player» расположенной на сцене «Scene1» и выполняет функцию передвижения игрока;

скрипт «Timer» закреплён за объектом «Timer», расположенной на сцене «\_MainMenu» и «Scene1», который выполняет подсчёт проведенного в игре времени;

скрипт «PixelCamer» закреплён за камерами и выполняет функцию пикселизации камер.

### 3.1.3 Спецификация программы

Точное название проекта и его состав приводится в Таблице 6.

Таблица 6 — Спецификация программы

Имя файла	Назначение
1	2
ELoT.exe	Исполняемый файл проекта
*.png	Формат изображения, который используется в Unity для текстур и спрайтов.
*.unity	Файл, который может содержать данные о сценах в Unity.
*.unity.meta	Метаданные для файла .unity, которые содержат информацию о его идентификаторе и других свойствах.
*.asset	Файл, содержащий данные активов (например, текстуры, модели и т.д.) в Unity.
*.asset.meta	Метаданные для файла .asset, содержащие информацию о его идентификаторе и других свойствах.
*.meta	meta - Общий файл метаданных для различных активов в Unity, содержащий информацию об идентификаторе и других свойствах.
*.anim	anim - Файл анимации, который содержит данные о движении объектов в Unity.
*.DLL	Библиотека динамической компоновки, которая содержит код на C# и используется в Unity.
*.cs	Файл исходного кода на языке C#.
*.cs.meta	файл метаданных для файла .cs.
*.prefab	файл префаба, который представляет собой шаблон объекта в Unity.
*.prefab.meta	файл метаданных для файла .prefab.



## 4 Тестирование

### 4.1 Тесты на использование

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Разработанные тест-кейсы и статус их выполнения представлены в приложение К.

Расписание работ над проектом представлено в таблице 7.

Таблица 7 – Расписание работ над проектом

Имя	Дата	Деятельность	Продолжительность, ч
Уланович Владислав	01.12.2024	Разработка тест-кейсов	4
Шульга Сергей	02.12.2024	Проведение тестирования	4
Уланович Владислав	03.12.2024	Написание отчётов о дефектах	4
Шульга Сергей	04.12.2024	Проведение регрессионного тестирования	2
Уланович Владислав	04.12.2024	Написание отчёта о результатах тестирования	1
Шульга Сергей	04.12.2024	Разработка тест-кейсов	4

### 4.2 Отчёт о результатах тестирования

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

Статистика по всем дефектам представлена в таблице 8.

Таблица 8 – Статистика по всем дефектам

		Важность			
Статус	Количество	Низкая	Средняя	Высокая	Критическая
Найдено	0	0	0	0	0
Исправлено	0	0	0	0	0
Проверено	0	0	0	0	0
Открыто заново	0	0	0	0	0
Отклонено	0	0	0	0	0

## 5 Руководство пользователя

Игровое приложение «Elven Illusion of Time» позволит пользователю провести время с пользой, получить эстетическое удовольствие, способствует улучшению таких навыков, как логическое мышление.

Создаваемое приложение ориентировано на пользователей старше 16 лет.

Качество и скорость работы приложения всегда зависит от самих характеристик персонального компьютера. Поэтому приложение должно было быть протестировано на разных машинах.

Тестирование проводилось на разных персональных компьютерах и результаты были удовлетворительные.

Данный программный продукт можно использовать в любое время, так как он не ограничен в использовании.

Загрузка игрового приложения предполагает наличие свободной памяти в размере 1 Гб.

### 5.1 Руководство системного программиста

Для того, чтобы установить программу необходимо разархивировать файл EIoT.zip.

### 5.2 Запуск программы

После этого, как игра была разархивирована, по двойному нажатию на «Elven Illusion of time.exe» игра будет запускаться.

### 5.3 Инструкция по работе с программой

При запуске игры пользователя встречает главное меню, в котором игрок может начать игру, почитать новости, настроить игру и выйти из игры. Главное меню представлено на рисунке 1.

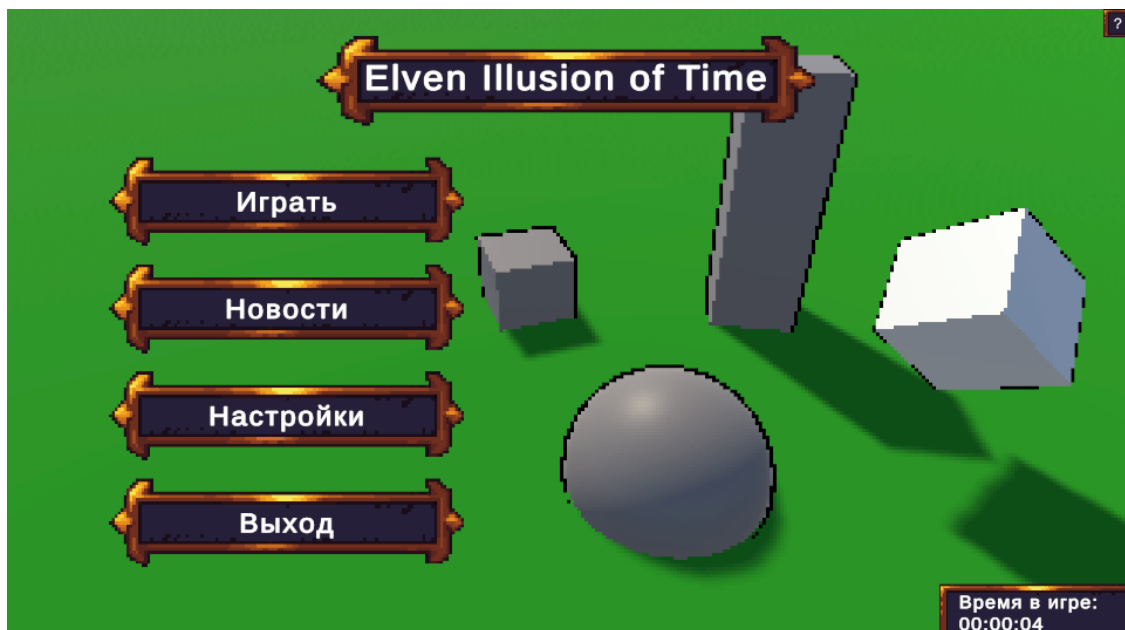


Рисунок 1 – Главное меню.

У пользователя есть возможность кликнуть по кнопке Играть, Новости, настройки, выход, справка (знак вопроса в правом верхнем углу).

При нажатии на кнопку «Справка» ничего не произойдёт, она сделана для дальнейшего поддержания приложения.

При нажатии на кнопку «Новости» будет совершен переход на вкладку новости. Меню новостей представлено на рисунке 2.



Рисунок 2 – Меню новостей.

При нажатии на кнопку «Назад» будет переход на главное меню, аналогично и в других меню.

Изм.	Лист	Поддокум.	Подпись	Дата

ТЗ 2-40 01 01.35.40.12.24

Лист

21

При нажатии на кнопку «Выход» приложение будет закрыто.

При нажатии на кнопку «Настройки» будет выполнен переход в меню настроек, есть возможность смены кнопок управления и графики, также настройка звука. Меню настроек представлено на рисунке 3.

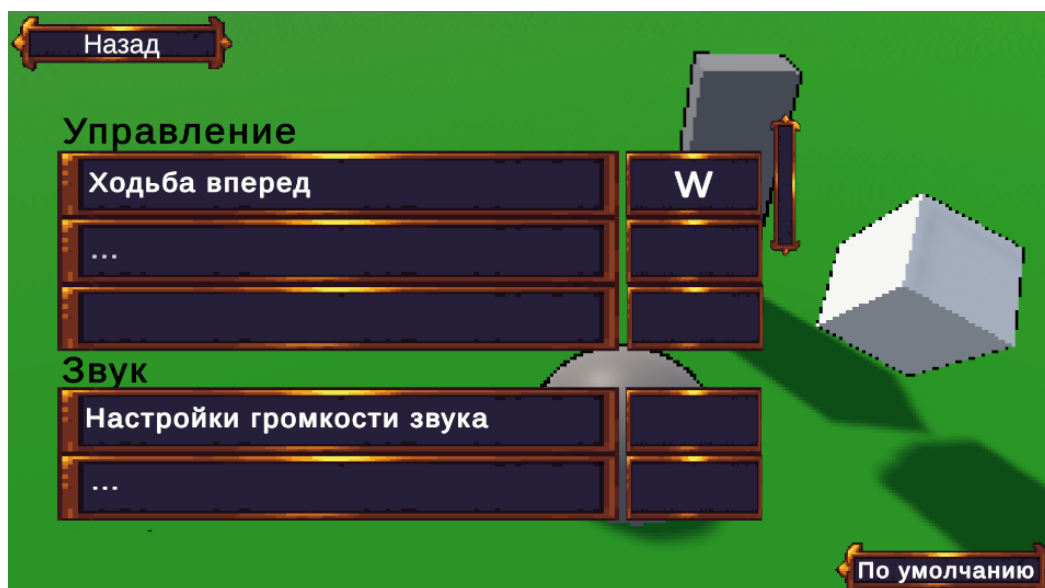


Рисунок 3 – Меню настроек.

При нажатии на кнопку «По умолчанию» настройки будут сброшены». При нажатии на кнопку «Играть» будет переход в меню с выбором сохранений. Меню с сохранениями представлено на рисунке 4.



Рисунок 4 – Меню сохранений

При нажатии на кнопку «Сохранение 1» будет переход в игру. Начало игры представлено на рисунке 5.

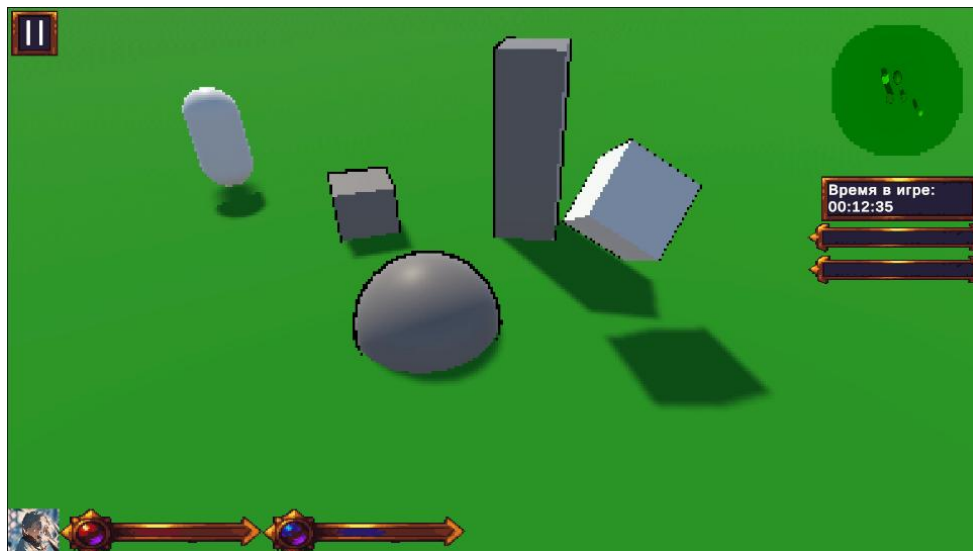


Рисунок 5 – Начало игры.

Управление персонажем при помощи кнопок W, A, S, D, а также пробела: пробел отвечает за прыжок, W отвечает за ходьбу вперед, A – ходьба влево, S – ходьба назад, D – ходьба вправо.

При нажатии на кнопку «||» (пауза) будет открыто меню паузы. Меню паузы представлено на рисунке 6.

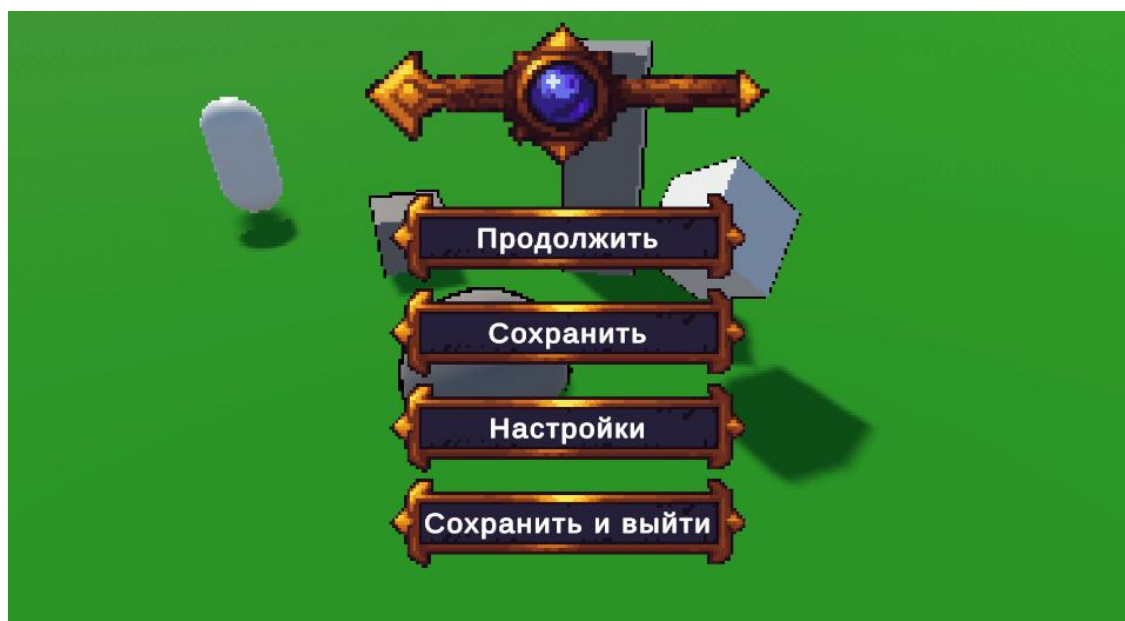


Рисунок 6 – Меню паузы.

При нажатии на кнопку «Продолжить» игра возобновится.

При нажатии на кнопку «Сохранить» ничего не произойдёт, она сделана для дальнейшего поддержания приложения.

При нажатии на кнопку «Настройки» будет выполнен переход в меню настроек, есть возможность смены кнопок управления и графики, также настройка звука. Меню настроек представлено на рисунке 3.

При нажатии на кнопку «Сохранить и выйти» будет выполнен переход в главное меню. Главное меню представлено на рисунке 1.

#### **5.4 Завершение работы с программой**

Завершить работу игрового приложения можно следующим способом: в главном меню присутствует кнопка «Выйти», после нажатия на которую игра закрывается.

					ТЗ 2-40 01 01.35.40.12.24	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

## Заключение

Целью данного техно-рабочего проекта по предмету ТРПО является разработка и проектирование программного продукта – игрового приложения «Elven Illusion of Time». Данный программный продукт позволит игрокам осознать значение времени и научиться эффективнее его управлять. Игра покажет, как важно бережно относиться ко времени, использовать его с умом и планировать свою жизнь грамотно. Этот игровой опыт поможет развить навыки тайм-менеджмента, сделает пользователей более дисциплинированными и организованными.

В ходе реализации поставленной задачи были получены знания по использованию платформы Unity, а также закреплены знания курса предмета «Технология разработки программного обеспечения».

Поставленная задача реализована частично. В соответствии с задуманными требованиями создано и протестировано 15 функций.

В ходе тестирования все исключительные ситуации были обработаны. Проект работает без сбоев и ошибок. В поставленной задаче был реализован простой и понятный пользовательский интерфейс.

Исходя из этого, можно сделать вывод, что частичная реализация программы прошла успешно.

### Список использованных источников

1. Как создать игру на Unity [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/kak-sozdat-igru-na-unity/> – Дата доступа: 07.11.2024.

2. ПО для разработки игр [Электронный ресурс]. – Режим доступа: <https://unity.com/ru/games> – Дата доступа: 08.11.2024.

3. Как сделать 3D игру на Unity [Электронный ресурс]. – Режим доступа: <https://www.progkids.com/blog/kak-sdelat-3d-igru-na-unity> – Дата доступа: 07.11.2024.

					ТЗ 2-40 01 01.35.40.12.24	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26