

Учреждение образования
«Гродненский государственный политехнический колледж»

Специальность: 2-40 01 01 «Программное обеспечение информационных
технологий»

Специализация: 2-40 01 01.35 ««Программное обеспечение обработки
экономической и деловой информации»

Предмет «Конструирование программ и языки программирования»

Группа: ПЗТ-40

КУРСОВОЙ ПРОЕКТ

Тема: Разработка игрового приложения «Путь авантюриста»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Разработала
Руководитель проекта

К.В.Савило
А.Г.Бабуль

2025

Содержание

Введение	4
1 Анализ предметной области и формулировка требований к программе	5
1.1 Исследование предметной области.....	5
1.2 Инструменты разработки	5
2 Проектирование.....	5
2.1 Диаграмма вариантов использования	7
2.2 Диаграмма деятельности.....	8
2.3 Описание тестов.....	10
3 Построение программы	11
4 Тестирование	16
5 Применение	17
5.1 Назначение и условия применения программы.....	17
5.2 Инсталляция	17
5.3 Выполнение программы.....	19
Заключение.....	24
Список использованных источников.....	25
Приложение А Листинг программы	
Приложение Б Описание тестов	
Приложение В Тестовый сценарий	

					КП 2-40 01 01.35.40.12.25 ПЗ		
Изм.	Лист	№докум.	Подпись	Дата			
Разраб.	Савило				Разработка игрового приложения «Путь авантюриста»	Лит.	Лист
Пров.	Бабуль						3
							25
Н. контр.							
Утв.							

Введение

На данном курсовом проекте была поставлена задача разработать игровое приложение «Путь авантюриста». Данный программный продукт позволит пользователю получить эстетическое удовольствие, создаст неповторимый игровой опыт, сочетающий приключения, исследование, торговлю и сражения. Игроки получают удовольствие от самого процесса путешествия, открытия новых локаций и достижения успеха.

Создаваемое приложение ориентировано на пользователей старше 12 лет.

Далее приведено краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ предметной области и формулировка требований к программе». В нем можно ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, функциональные и нефункциональные требования к программному продукту. В подразделе «Инструменты разработки» рассмотрена среда, в которой создается данный программный продукт.

В разделе «Проектирование задачи» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе составлены диаграмма вариантов использования и диаграмма деятельности.

«Построение программы» – третий раздел отчета, в котором описываются все элементы и объекты, которые использованы при реализации данного приложения. В этом разделе описана диаграмма классов.

Четвертый раздел – «Тестирование». В нем представлен отчет о проведении тестирования.

В разделе «Применение» описано назначение программы, область применения и среда функционирования программного обеспечения.

«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 приведен листинг программы.

В приложении Б представлено описание тестов.

В приложении В представлен тестовый сценарий.

В графической части представлена диаграмма классов

1 Анализ предметной области и формулировка требований к программе

1.1 Исследование предметной области

Наименование задачи – игровое приложение «Путь авантюриста».

Цель разработки данного программного продукта заключается в создании игрового приложения, которое позволит пользователю окунуться в фэнтезийный мир увлекательных приключений, где он в роли мага-авантюриста, сможет выполнять интересные задания и исследовать игровой мир.

Игровое приложение выполнено в жанре RPG в стиле фэнтези-средневековья.

Назначение: данный программный продукт ориентирован на несколько категорий игроков: любителей аниме и фэнтези, поклонников ролевых игр (RPG), фанатов крафта и торговли, любителей экшена и приключений, креативных игроков. Таким образом, игра будет привлекательна для широкой аудитории, включая как казуальных игроков, так и более опытных любителей RPG, что делает её универсальной и многогранной.

В центре сюжета стоит авантюрист, чей жизненный путь состоит в поиске приключений. Игрок приходит в город, где ему предстоит вступить в гильдию авантюристов «Бесконечный путь», здесь его будут ждать разнообразные задания, от охоты на монстров до расшифровки древних текстов. Игра предлагает игрокам возможность развивать своего персонажа, повышая его ранг в гильдии и открывая новые уровни сложности и уникальные награды. Игрок может взаимодействовать с NPC, покупать и продавать различные предметы, а также украшать свой собственный дом.

Задача игрока - достичь наивысшего ранга, в гильдии, выполнив все задания и получить звание лучшего авантюриста, а также же наслаждаться увлекательными приключениями в мире фэнтези.

Периодичность использования данного программного продукта неограниченна. Игрок может в любой момент сбросить игровой прогресс, а так же имеет возможность даже после выполнения всех заданий продолжать игру.

Существует множество игр об авантюристах, однако абсолютных аналогов данного программного продукта не было найдено.

1.2 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.

					УП КПиАП 2-40 01 01.35.40.12.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

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 2013 – для написания документации к программному продукту;
- Smart Install Maker – утилита для создания инсталляторов;
- Dr.Explain – инструмент разработки пользовательской документации;
- GitHub – веб-сервис для хостинга IT-проектов;
- Google Drive – сервис хранения, редактирования и синхронизации файлов, разработанный компанией Google.

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

- процессор Intel Core i5-12500H (3.30 / 4.50 GHz);
- ОЗУ: 8Gb;
- память: HDD 512Gb;
- ОС: Windows 10.

					УП КПиЯП 2-40 01 01.35.40.12.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		6

2 Проектирование

2.1 Диаграмма вариантов использования

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

Суть такой диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования.

В данной диаграмме вариантов использования актером является игрок.

Игроку доступны следующие варианты использования: играть, читать справку, изменить музыку, выйти из игры.

У варианта использования «Играть» наблюдаются следующие «extend»-связи: посмотреть инвентарь, перейти на локации, взаимодействовать с персонажем, купить игровые предметы, продать добытые ресурсы, пройти тест на вступление в гильдию, украсит дом, выбрать задание соответствующее рангу.

У варианта использования «Изменить настройки» наблюдаются следующие «extend»-связи: выбрать мелодию, остановить воспроизведение музыки, продолжить воспроизведение музыки.

У варианта использования «Посмотреть инвентарь» наблюдаются следующие «extend»-связи: перейти к использованию некоторых предметов, посмотреть информацию о предметах.

У варианта использования «Взаимодействовать с персонажем» наблюдаются следующие «extend»-связи: начать диалог.

У варианта использования «Украсить дом» наблюдаются следующие «include»-связи: разместить предмет интерьера.

У варианта использования «Выбрать задание, соответствующее рангу» наблюдаются следующие «extend»-связи: собрать растения, расшифровать сообщение, собрать кристаллы, создать зелье, изловить белку.

У варианта использования «Перейти к использованию некоторых предметов» наблюдаются следующие «extend»-связи: создать зелье, посмотреть информацию в руководстве начинающего травника, посмотреть информацию в руководстве по зельеварению.

Диаграмма вариантов использования представлена на рисунке 1.

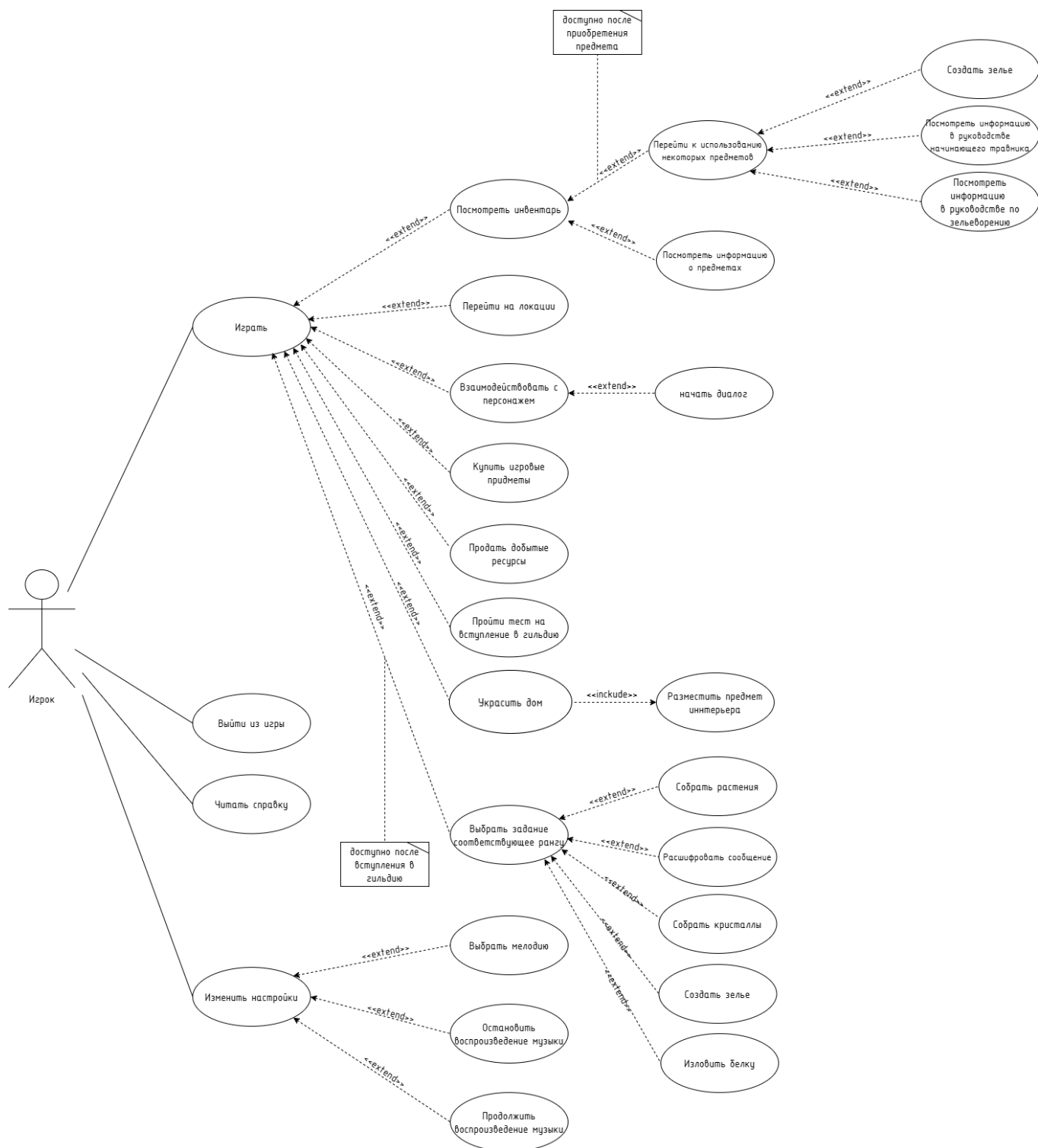


Рисунок 1 – Диаграмма вариантов использования

2.2 Диаграмма деятельности

Диаграмма деятельности — это поведенческая диаграмма, которая показывает поток работы или действий в рамках системы или процесса. Она отображает последовательность шагов и возможные варианты выполнения работы,

включая параллельные процессы и ветвления. Диаграмма деятельности включает в себя такие элементы, как начальные и конечные узлы, узлы действий, узлы решений, вилки и слияния, а также потоки управления, которые связывают эти узлы.

На диаграмме отображен процесс покупки предмета.
Диаграмма деятельности представлена на рисунке 2.

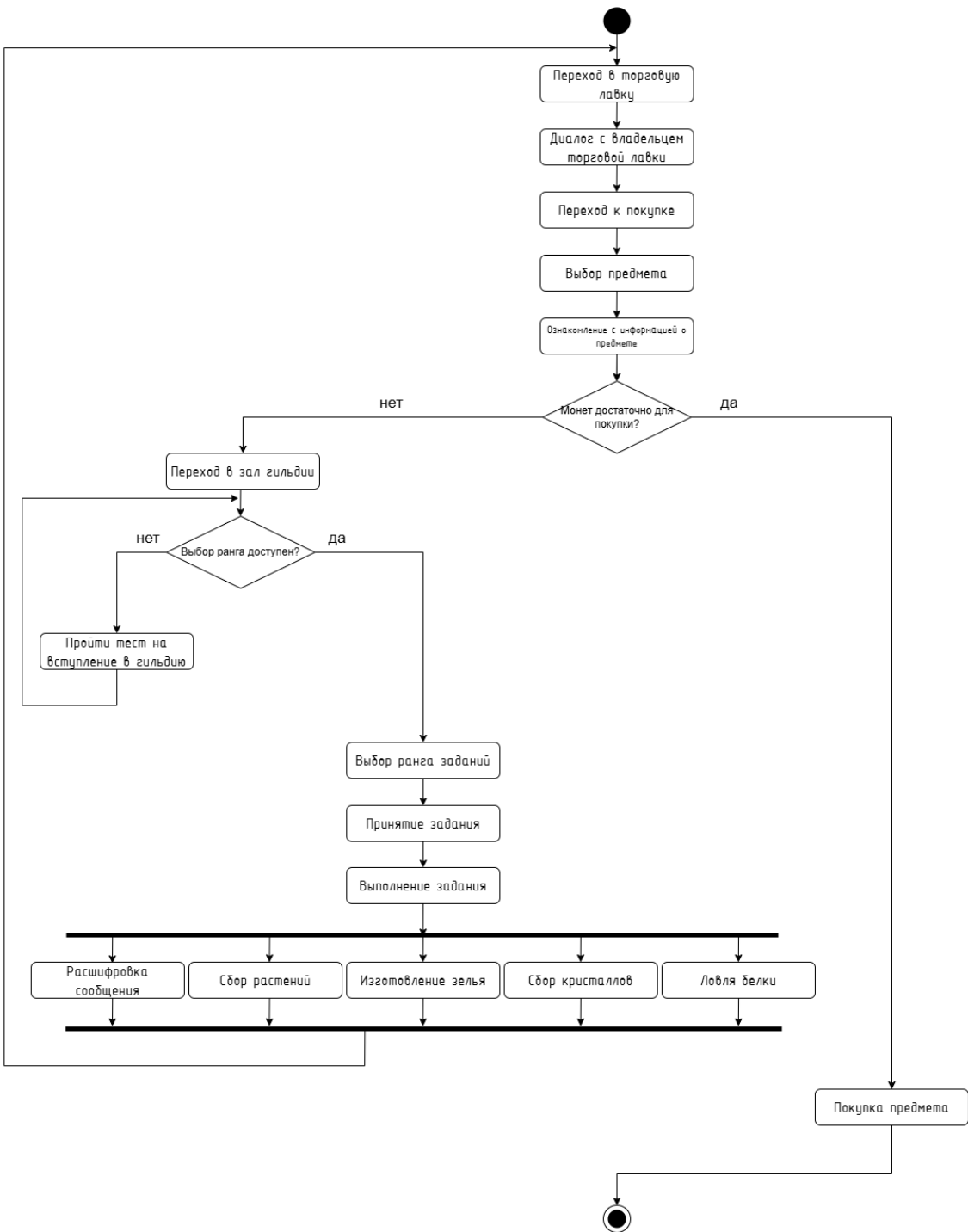


Рисунок 2 – Диаграмма деятельности

2.3 Описание тестов

При разработке приложения необходимо будет провести тест по игровому процессу.

Тесты по игровому процессу проводятся для обеспечения высокого качества игры, выявления ошибок, улучшения геймплея и повышения удовлетворенности игроков. Они помогают разработчикам понять, насколько увлекательна и удобна игра, выявить баги и несоответствия требованиям, а также оценить общую игровую механику.

Описание тестового сценария представлено в приложении Б.

					УП КПиЯП 2-40 01 01.35.40.12.25 ПЗ	Лист
						10
Изм.	Лист	№докум.	Подпись	Дата		

3 Построение программы

Диаграмма классов – это визуальное представление структуры системы, отображающее классы, их атрибуты, методы и взаимосвязи между ними. Она является одним из ключевых элементов объектно-ориентированного проектирования и используется для моделирования статической структуры приложения.

Основная цель диаграммы классов – показать, как элементы системы связаны друг с другом, какие свойства и функции они имеют, а также какие отношения существуют между ними.

Диаграмма помогает разработчикам и архитекторам понять архитектуру системы, определить связи между объектами и грамотно спроектировать базу данных или архитектуру приложения.

В данном разделе будут описаны основные 11 классов из 64.

Класс «playerControler» содержит:

Свойства:

- public float speed: скорость движения игрока;
- public float jumpForce: сила прыжка;
- public float moveInput: входные данные для движения;
- private Rigidbody2D rb: компонент Rigidbody2D для физики;
- private bool facingRight: направление, в котором смотрит игрок;
- private bool isGrounded: состояние, находится ли игрок на земле;
- public Transform freePos: позиция для проверки касания земли;
- public float checkRadius: радиус проверки касания;
- public LayerMask whatIsGround: слой, определяющий, что считается землей;
- private Animator anim: компонент Animator для анимации.

Методы:

- private void Start(): инициализация компонентов;
- private void FixedUpdate(): обновление физики, движение игрока;
- private void Update(): обновление состояния игрока;
- void Flip(): переворот игрока в другую сторону.

Класс «Bullet_enemy» содержит:

Свойства:

- public float offset: угол поворота пули;
- public GameObject bullet: префаб пули;
- public Transform shotPoint: точка, откуда будет вылетать пуля;
- private float timeBtwShots: время между выстрелами;
- public float startTimeBtwShots: начальное время между выстрелами;

– public Transform parentObject: объект-родитель для пули.

Методы:

– private void Start(): инициализация корутины для стрельбы;

– private IEnumerator ShootBullets(): корутина для создания пули с задержкой;

– private void Update(): обновление положения пули.

Класс «Bullet» содержит:

Свойства:

– public float speed: скорость движения пули;

– public float lifetime: продолжительность жизни пули;

– public float lifetime1: время, через которое пуля будет уничтожена;

– public float distance: дистанция, на которую будет двигаться пуля;

– public int damage: урон, который наносит пуля;

– public LayerMask whatIsSolid: слой для коллизий с твердыми объектами;

– public LayerMask whatIsWrag: слой для коллизий с врагами;

– private Vector2 direction: направление движения пули;

– public GameObject bulletEffect: эффект, который появляется при столкновении.

Методы:

– private void Start(): уничтожение пули через заданное время;

– public void SetDirection(Vector2 dir): установка направления движения;

– private void Update(): обновление движения пули и проверка на столкновения.

Класс «hp» содержит:

Свойства:

– public int hp: здоровье игрока;

– public Image a: изображение первой полосы здоровья;

– public Image b: изображение второй полосы здоровья.

Методы:

– public void boy(): уменьшает здоровье игрока.

Класс «Rang» содержит:

Свойства:

– public GameObject pauseMenuUI: меню паузы;

– public Text text1: текст для отображения результата;

– public int heal: количество здоровья;

– public bool b: флаг состояния.

Методы:

– public void Start(): инициализация значений;

– void Update(): проверка состояния игры;

– private IEnumerator HandleGameOver(): обработка окончания игры.

Класс «Enemy» содержит:

Свойства:

– public int health: здоровье врага;

– public GameObject deathEffect: эффект смерти врага.

Методы:

– private void Update(): проверка состояния здоровья и уничтожение врага;

– public void TakeDamage(int damage): уменьшает здоровье врага.

Класс «Gun» содержит:

Свойства:

– public float offset: угол поворота;

– public GameObject bullet: префаб пули;

– public Transform shotPoint: точка выстрела;

– private float timeBtwShots: время между выстрелами;

– public float startTimeBtwShots: начальное время между выстрелами.

– public Transform parentObject: объект-родитель для пуль.

Методы:

– private void Update(): обновление состояния и стрельба.

Класс «hil» содержит:

Свойства:

– public Image a: изображение для первой полосы здоровья;

– public Image b: изображение для второй полосы здоровья;

– public Text text: текст для отображения количества зелий.

Методы:

– void Update(): проверка нажатия клавиш для использования зелий;

– private void hiler(): восстанавливает здоровье игрока.

Класс «casanije_wrag» содержит:

Свойства:

– public int x = 0: используется для проверки нужно ли отталкивание игрока.

Методы:

– private void OnCollisionEnter2D(Collision2D collision): нанесении урона при касании игрока и его отталкивание при необходимости.

Класс «Draggable» содержит:

Свойства:

– private Canvas canvas: ссылка на родительский Canvas;

– private RectTransform rectTransform: прямоугольная трансформация объекта;

– private Vector3 originalPosition: оригинальная позиция объекта.

Методы:

- public void OnBeginDrag(PointerEventData eventData): сохраняет оригинальную позицию при начале перетаскивания;
- public void OnDrag(PointerEventData eventData): обновляет позицию объекта при перетаскивании;
- public Vector3 GetCurrentPosition(): возвращает текущую позицию объекта.

Класс «JewelryManager» содержит:

Свойства:

- public GameObject[] jewelryPrefabs: массив префабов украшений;
- public Button[] buttons: кнопки для создания украшений;
- private Dictionary<int, int> quantities: словарь для хранения количества украшений;
- public Transform parentTransform: объект-родитель для созданных украшений;
- public Vector2 spawnPosition: позиция, куда будут создаваться украшения;
- private List<Draggable> draggableJewelry: список перетаскиваемых украшений.

Методы:

- void Start(): инициализация кнопок и загрузка данных;
- void Update(): проверка нажатий для управления украшениями;
- void LoadQuantities(): загрузка количества украшений из файла;
- void LoadJewelryPositions(): загрузка позиций украшений из файла;
- public void OnJewelryButtonClick(int index): создание экземпляра украшения;
- void SavePositions(): сохранение позиций перетаскиваемых украшений в файл.

Далее приведено описание связей в диаграмме классов.

Зависимость обозначает такое отношение между классами, что изменение спецификации класса-поставщика может повлиять на работу зависимого класса, но не наоборот.

Класс Bullet вызывает методы классов hp и Enemy, тем самым уменьшая переменные жизнью игрока и врага при попадании.

Класс hill изменяет(увеличивает) переменную класса hp для увеличения единиц жизни игрока.

Класс casanije_wrag вызывает метод класса hp, для нанесения урона.

Класс Rang использует информацию из hp (значение переменной hp), чтобы обновлять свои поля.

Класс Enemy изменяет переменную класса Rang, определяя условие определения ранга.

Отношение ассоциации показывает, что объекты одной сущности (класса) связаны с объектами другой сущности таким образом, что можно перемещаться от объектов одного класса к другому.

Классы `Gan` и `Bullet_enemy` создают пули используют метод `Bullet`, чтобы задавать ей направление.

Класс `JewelryManager` добавляет компонент `Draggable` для перетаскивания объектов, то есть управления ими.

Диаграмма классов представлена в графической части проекта.

					УП КПиЯП 2-40 01 01.35.40.12.25 ПЗ	Лист
						15
Изм.	Лист	№докум.	Подпись	Дата		

4 Тестирование

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.

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

Разработанный тестовый сценарий и статус его выполнения представлен в приложении Б.

Расписание проведения и время, затраченное на тестирование, описано в таблице 2.

Таблица 2 – Расписание работ над проектом

Имя	Дата	Деятельность	Продолжительность, ч
Савило Ксения	30.05.2025	Разработка тестов	2
Савило Ксения	05.06.2025	Тестирование игрового приложения	2
Савило Ксения	10.06.2025	Исправление найденных ошибок	2
Савило Ксения	15.06.2025	Проведение регрессионного тестирования	1
Савило Ксения	18.06.2025	Составление отчета о результатах тестирования	2

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 3.

Таблица 3 – Статистика по всем дефектам

Статус	Количество	Важность			
		Низкая	Средняя	Высокая	Критическая
Найдено	1	1	0	0	0
Исправлено	1	1	0	0	0
Проверено	1	1	0	0	0
Открыто заново	0	0	0	0	0
Отклонено	0	0	0	0	0

Исходя из данных по приведенным ранее таблицам, можно сделать вывод, что все возможные ошибки и недочеты были найдены и исправлены.

5 Применение

5.1 Назначение и условия применения программы

Цель данного программного продукта заключается в разработке программного продукта, который позволит пользователю получить эстетическое удовольствие, создаст неповторимый игровой опыт, сочетающий приключения, исследование, торговлю и сражения.

Создаваемое приложение будет рассчитано на любого пользователя.

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.

Сама программа была разработана на программном устройстве со следующими характеристиками:

- процессор Intel Core i5-12500H (3.30 / 4.50 GHz);
- ОЗУ: 8Gb;
- память: HDD 512Gb;
- ОС: Windows 10.

5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл Setup.exe. Появится мастер установки игрового приложения «Путь авантюриста», представленный на рисунке 4.

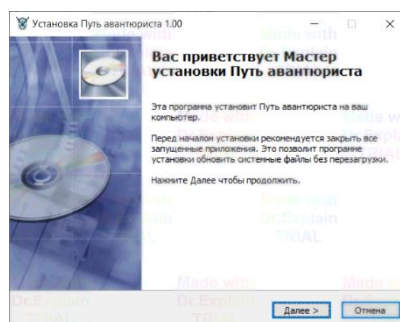


Рисунок 4 – Мастер установки игрового приложения «Путь авантюриста»

После нажатия кнопки «Далее» появляется возможность выбора места для установки программного продукта, представленное на рисунке 5.

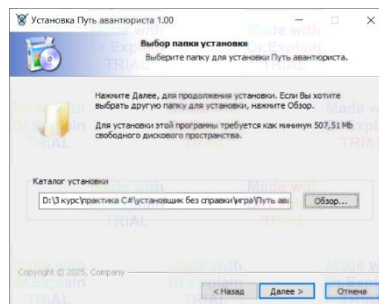


Рисунок 5 – Мастер установки игрового приложения «Путь авантюриста»

После нажатия на кнопку «Далее» появляется возможность выбрать создавать ли ярлыка игрового приложения, представленная на рисунке 6.

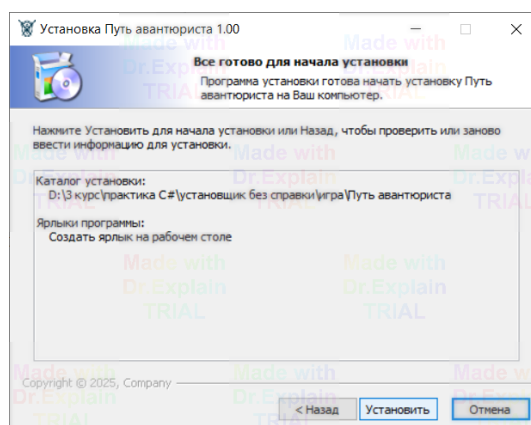


Рисунок 6 – Создание ярлыка

После нажатия на кнопку «Далее» открывается форма завершения установки, представленная на рисунке 7.

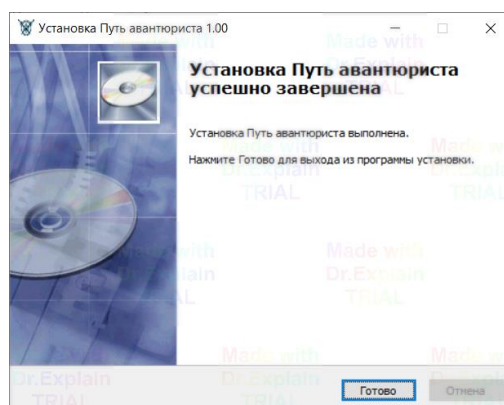


Рисунок 7 – Завершение установки

После установки на «Рабочем столе» появится ярлык для запуска приложения, который представлен на рисунке 7.

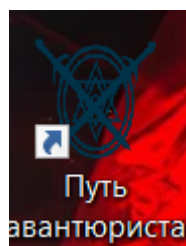


Рисунок 8 – Ярлык приложения

5.3 Выполнение программы

При запуске игры пользователя встречает экран загрузки. Экран загрузки представлен на рисунке 9.



Рисунок 9 – Экран загрузки

После игрок попадает на Главное меню, как представлено на рисунке 10.



Рисунок 10– Главное меню

У пользователя есть возможность кликнуть по кнопке «Выйти из игры», «Музыка», «Играть», «Справка».

При нажатии на кнопку в виде изображения листка с вопросительным знаком будет открыта справка.

При нажатии на кнопку «Выход из игры» приложение будет закрыто.

При нажатии на кнопку играть попадаете на сцену «Входа в город», где вас встречает предисловие к игре. Сцена «Входа в город» представлена на рисунке 11.

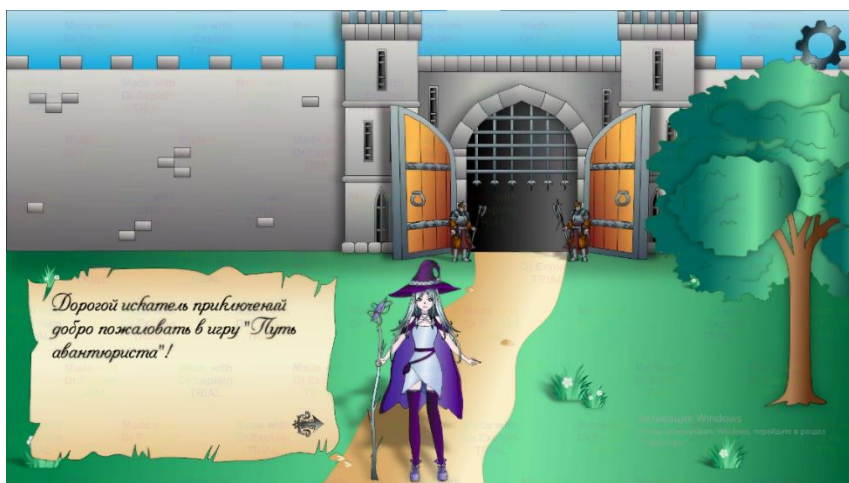


Рисунок 11– Сцена «Входа в город»

Для перелистывания предисловия необходимо нажать на кнопки стрелки, расположенную в нижнем правом углу листа предисловия.

После прочтения предисловия, можно продолжить игру нажав на кнопку «В путь», или сбросить игровой прогресс к изначальному (если таковой был), нажав на кнопку «Начать с нуля».

Нажав на одну из данных кнопок вы попадаете на улицу города где расположены здания, в которые можно войти нажав на дверь.

Управление персонажем осуществляется при помощи стрелок клавиатуры, или же клавиш WASD: левая стрелка или клавиша A и правая стрелка или клавиша D это движения в лево и право соответственно, верхняя стрелка или W - прыжок, нижняя стрелка или клавиша S - приседание. Для атаки используется левая клавиша мыши, атака доступна только на сценах сражения.

В левом верхнем углу расположена кнопка перехода в инвентарь (изображена в виде сумки), а в правом верхнем углу кнопка «Меню».

На сцене инвентарь можно просмотреть количество добытых во время игры ресурсов, а также информацию о них, кликнув по значку предмета из левой части инвентаря. Для некоторых предметов, предусмотрена кнопка «Использовать», которая будет располагаться в нижнем правом углу, и нажав на которую будет произведён переход на сцену использования предмета. Покинуть инвентарь можно, а также другие аналогичные сцены можно нажав кнопку «Выход». Инвентарь представлен на рисунке 12.

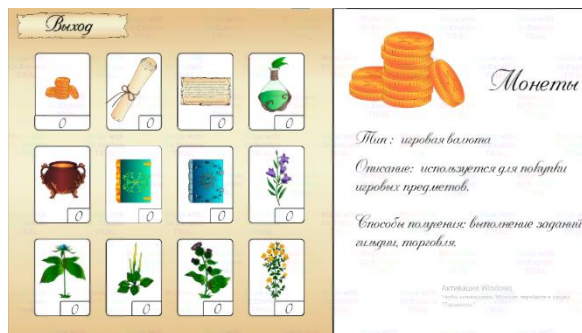


Рисунок 12 – Инвентарь

Нажав на кнопку «Меню» будет открыто меню, на котором расположены кнопки аналогичные кнопкам главного меню. Нажав на крестик меню будет закрыто. При покупке предмета «Карта» в меню будет добавлена кнопка «Карта», которая ведёт на сцену карты, где отмечаются посещённые локации, кликнув по названиям которых можно их посетить.

Нажав на дверь здания гильдии, располагающегося на улице города, будет произведён переход на сцену «Зал гильдии», представленный на рисунке 13.



Рисунок 13 – Сцена «Зал Гильдии»

На данной сцене располагается доска заданий, с доступными заданиями и администратор гильдии. Чтобы получить доступ к заданиям необходимо зарегистрироваться в гильдии, подойдя к администратору, и нажав на кнопку «Начать диалог» (изображена в виде трёх вопросительных знаков), вы получите необходимую информацию о дальнейших действиях. После будет отображена кнопка «Тест», нажав на которую вы попадёте на сцену для прохождения теста.

После прохождения теста вы будете зарегистрированы в гильдии, а ваш текущий ранг будет отображён в верхнем левом углу около кнопки «Инвентарь».

Также будет открыт доступ к заданиям вашего ранга, чтобы перейти к выполнению задания, нужно выбрать ранг на доске заданий, кликнув по нему, выбранный ранг обводится рамкой. Далее нужно кликнуть по листку задания, после чего вы можете ознакомиться с условиями выполнения и наградой за выполнение,

и нажать кнопку «Принять», после чего вы будете направлены на сцену соответствующего задания.

Вернуться не выполнив задание, можно нажав на кнопку «Гильдия». При успешном выполнении задания вы получите сообщение о зачислении награды, провалив задание вы также получите соответствующее сообщение с возможностью начать заново нажав на соответствующую кнопку или вернуться в гильдию.

В игре присутствуют функции торговли, перейдя на сцены магазинов, подойдя к торговцам и начав диалог описанным выше способом будут отображены кнопки «К покупке», «К продаже». Нажав на каждую из которых вы перейдёте на соответствующую сцену.

Для покупки нужно нажать на предмет и в правом углу его описания нажать кнопку «Купить». Слева отображено количество ваших монет, справа цена предмета за единицу.

Аналогично устроена сцена продажи. Для продажи нужно нажать на предмет ввести количество в поле «Шт.» и нажать кнопку «Продать».

Также в игре присутствует возможность украшать дом персонажа. Нажав на кнопку в виде плюса, расположенного около инвентаря, будет открыт список украшений, нажав на плюс повторно, список будет скрыт, нажав на кнопку из списка на сцену будет добавлено соответствующее украшение. Украшения можно перемещать с помощью курсора, удалять нажав клавишу `backspace`, изменять размер нажимая на `+` и `-`, поворачивать в лево и право нажав клавишу `Tab`, а также поворачивать по окружности, нажав клавишу `Q`, все эти действия можно применять к украшению на которое наведён курсор.

В игре есть функция Крафта зелий. Приобретя котёл и нажав кнопку «Использовать» в инвентаре, будет произведён переход на сцену котла, где нажав на кнопку ингредиента, и расположив его с помощью мыши в ячейки в соответствии с рецептом будет создано зелье, которое необходимо забрать, кликнув по нему. Сцена котелка отображена на рисунке 14.

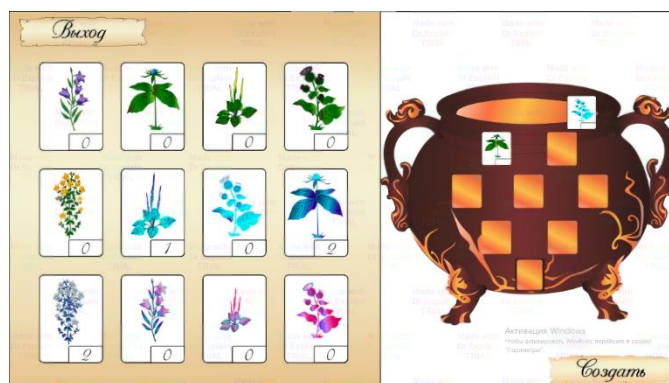


Рисунок 14 – Сцена «Котелок»

На сценах с врагами в левом верхнем углу будут отображаться единицы здоровья, при наличии зелья здоровья их можно будет восстанавливать, нажав на клавишу F. потеряв все очки здоровья игрок провалит задание.

					УП КПиЯП 2-40 01 01.35.40.12.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		23

Заключение

Целью данного курсового проекта была разработка программного продукта «Путь авантюриста». В ходе тестирования не было выявлено исключительных ситуаций. Проект работает без сбоев и ошибок. В процессе разработки поставленной задачи были получены знания по использованию платформы Unity и закрепились знания по языку программирования C# и закрепились умения создания собственных функций.

Поставленная задача выполнена в соответствии со всеми ранее задуманными требованиями, созданы и протестированы все необходимые компоненты проекта.

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 программа реализована успешно.

					УП КПиЯП 2-40 01 01.35.40.12.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		24

Список использованных источников

1. Для программистов, не знакомых с Unity [Электронный ресурс]. – Режим доступа: <https://unity.com/ru/how-to/programming-unity/> – Дата доступа: 10.04.2025.
2. Unity с нуля: Занятие 2. C# и ООП [Электронный ресурс]. – Режим доступа: <https://aelit.by/unity-s-nulya-zanyatie-2-s-i-oop/D.html> – Дата доступа: 20.04.2025.
3. Полное руководство по языку программирования C# 13 и платформе .NET 9 [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/tutorial/> – Дата доступа: 25.04.2025.
4. Создаем 2D-игру на Unity: инструкция для новичка [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/sozdaem-2d-igru-na-unity-instrukciya-dlya-novichka-2020-09-01> – Дата доступа: 30.04.2025.
5. Руководство по C# [Электронный ресурс]. – Режим доступа: <https://vscode.ru/prog-lessons/sozдание-redaktirovanie-i-udalenie-dannyih-xml-fayla-s-sharp.html> – Дата доступа: 01.05.2025.