

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка demo-версии 2D игры с элементами хоррора»
по дисциплине «Проектный практикум»

Заказчик: Насыров И.А.

Куратор: Насыров И.А.

Генеральный директор ООО "ЛамАрт"

Студенты команды Half-Sharpshooter

Петров К.А.

Александров Г.Д.

Фамилия И.О.

Екатеринбург, 2025

СОДЕРЖАНИЕ

Введение	3
1 Цель проекта	5
2 Критерии оценки к программному продукту	6
2.1 Уточнение требований	6
3 Обзор игр-аналогов	7
4 Составление бэклога	10
4.1 Примеры задач для каждого члена команды	11
5 Описание методологии разработки и процесса работы	12
6 Архитектура программного продукта.....	13
7 Стек разработки.....	14
8 Индивидуальные отчеты участников команды	15
9 Сценарии использования	18
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	22

ВВЕДЕНИЕ

Цель данного проектного практикума заключается в создании демо-версии видеоигры в жанре «Survival horror», сочетающей элементы выживания, исследования и сюжетного погружения для коммерческого распространения.

Жанр survival horror является одним из востребованных на рынке видеоигр, благодаря своему умению погружать игроков в атмосферу напряжения и страха. Наш проект актуален для аудитории, ищущей сложный и вызывающий геймплей в сочетании с пугающей и напряженной атмосферой, а также для фанатов S.T.A.L.K.E.R. и Darkwood.

Разработка игры также предоставляет команде уникальную возможность применять теоретические знания в области программирования, дизайна и управления проектами на практике. Это позволяет участникам углубить свои профессиональные навыки и получить ценный опыт работы в команде.

Кроме того, проект ориентирован на создание продукта, который может быть успешно представлен на рынке инди-игр, привлекая как отечественную, так и международную аудиторию. В данный момент, на фоне вновь возросшего интереса к серии игр S.T.A.L.K.E.R., возрастающего интереса к инди-играм, наша игра имеет потенциал выделиться благодаря своей атмосфере и сложному геймплею.

Игра предназначена для персональных компьютеров, основные платформы распространения включают Steam, Epic Games Store и VKPlay, что обеспечивает доступность как для российской, так и для международной аудитории.

В этом семестре проект позиционируется как демо-версия игры для привлечения инвестиций от потенциальных издателей.

Готовый продукт включает:

- 1) Готовый 1 акт для демонстрации геймплея и атмосферы.
- 2) Проработанные игровые механики для демо-уровня.
- 3) Готовая в рамках демо версии локация.
- 4) Презентация и материалы для представления издателям

1 Цель проекта

Целью проекта, поставленная заказчиком, является разработка видеоигры, готовой для коммерческого распространения. Игра должна быть достаточно качественной, чтобы привлечь внимание как игроков, так и инвесторов, при этом демонстрируя все основные аспекты геймплея, дизайна и технической реализации.

2 Критерии оценки к программному продукту

Основные критерии оценки включают:

Играбельная demo-версия или готовая игра, для дальнейшей презентации игрокам и инвесторам;

Механики отточены до необходимого минимума.

Дизайн приближен к финальной версии (Дизайн интерфейсов, окружения и т.д.)

Наличие понятной документации технической части для дальнейшей разработки;

2.1 Уточнение требований

Основное и единственное требование заказчика – разработать Demo-версию игры.

Хотя на начальном этапе конкретных ограничений и технических требований не было, наша команда самостоятельно определила список внутренних критериев для достижения поставленной цели. Эти критерии включали:

- Создание проработанного дизайн-документа, описывающего игровой процесс, сюжет и визуальные элементы.
- Разработка базовых механик: система инвентаря, механика стрельбы и ближнего боя, взаимодействие с окружением (NPC и различные хранилища предметов).
- Использование инструментов управления проектами для распределения задач и отслеживания прогресса, а также для составления дизайн документа. Данными инструментами по просьбе заказчика должны были стать корпоративная Jira и Confluence.

3 Обзор игр-аналогов

В процессе разработки нашей игры мы проанализировали несколько проектов, обладающих схожей атмосферой, жанром или механиками. Особое внимание было уделено следующим тайтлам:

Darkwood — игра, вдохновившая нас уникальным подходом к жанру survival horror. Её вид сверху, нелинейный сюжет и упор на исследование и выживание стали для нас ключевыми геймплейными ориентирами. Darkwood выделяется мощной атмосферой, уязвимостью игрока и необычной системой освещения для 2D-проекта. Она стала основным референсом в плане ощущений и подачи напряжённости.

S.T.A.L.K.E.R. — серия игр, послужившая вдохновением для построения сеттинга и нарратива. Постапокалиптический мир, наполненный аномалиями, опасностями, свободой исследования и взаимодействием с NPC, заложил основу атмосферы и мира нашей игры.

Дополнительно, мы провели сравнительный анализ с рядом менее масштабных, но концептуально близких проектов — **Noct**, **Subterrain** и **Motte Island**, чтобы чётко определить сильные и слабые стороны каждого, и учесть это при создании нашей игры.

Сходства и отличия по основным аспектам:

Механики выживания: Как и в Darkwood, важную роль играет управление ресурсами — еда, медикаменты, боеприпасы. Однако у нас оно будет более сбалансированным, без излишнего усложнения, чтобы сохранить динамику и доступность для широкой аудитории.

Атмосфера и мир: Как и в S.T.A.L.K.E.R., наш мир насыщен лором и скрытыми историями. Но мы отказались от пустых пространств и длинных

пробежек — каждая локация в игре насыщена, а атмосфера опасности будет сопровождать игрока на каждом шагу.

Геймплейный баланс: В отличие от Darkwood, где акцент сделан на атмосферу, у нас больше внимания уделяется геймплейному разнообразию, менеджменту и активному взаимодействию с окружением, чтобы избежать однообразия.

Интерфейс и подача: В отличие от Subterrain, мы стремимся к интуитивному и современному интерфейсу без перегрузки, сохранив при этом глубину управления.

Анимация и визуал: Мы учли визуальное однообразие Noct и Motte Island, найдя ассеты подходящие под стилистику, сохранив при этом стилистическую строгость, подходящую для survival horror.

Основные особенности, выделяющие наш проект:

Психологическое давление и напряжение — через окружение, звук, освещение и редкие внезапные угрозы создаётся устойчивое чувство тревоги. Схватки с сильными врагами чередуются со спокойными, но пугающими тихими моментами.

Плотный, насыщенный мир — игрока постоянно окружают элементы, с которыми можно взаимодействовать: тайники, документы, радио, разрушаемые объекты и неожиданные случайные события.

Гибридная атмосфера — мы соединили элементы Darkwood (освещение, тревожная музыка) и S.T.A.L.K.E.R. (зоны, мутанты, сюжеты, квесты) с динамичным геймплеем и индивидуальной системой выживания.

Развитие и прокачка — в отличие от Motte Island, у нас есть система развития персонажа и тактических решений, а каждый бой может быть решён разными способами.

Таким образом, наш проект сочетает в себе лучшее из жанра survival horror, избегая ошибок предшественников и предлагая уникальный игровой опыт, ориентированный на атмосферу, геймплей и выживание в враждебном, но проработанном мире.

4 Составление бэклога

Для структурирования процесса разработки был составлен подробный бэклог, который включал ключевые этапы работы и задачи для каждого члена команды. С первого дня работы над проектом, мы вели канбан доску на Yougile, на которой ставили задачи исходя из технического задания, которые нужно было выполнить в течение определенного времени – спрингта. Все задачи распределялись по спрингтам длительностью 1 недели с учётом их приоритетности.

Основные этапы разработки в этом семестре включали:

- Этап перепроектирования: переработка концепции, реструктуризация дизайн-документа, детализация геймплейных механик. На этом этапе разработчики исправляют ошибки с прошлого семестра, дизайнеры продолжают подбирать референсы и подготавливать новые элементы дизайна.
- Этап разработки: реализация системы инвентаря, механики боя, взаимодействия с NPC, прорисовывание NPC, локаций, интерфейса.
- Этап тестирования: тестирование игрового процесса, поиск и исправление багов, начало продвижения в социальных сетях

Изначально разработку планировалось вести по данному плану:

1. Этап перепроектирования (2 недели):

- Реформирование документов игры, изменение сюжета, подбор новых референсов, исправление ошибок в коде с прошлого семестра

На этом этапе разработчики исправляли ошибки кода с прошлого семестра, а геймдизайнер-аналитик занимался реструктуризацией проекта.

2. Этап разработки (7-8 недель):

- Создание оставшихся с прошлого семестра базовых механик: инвентаря, квестов, системы NPC и т.д.
- Уточнение всех механик, проработка баланса и поиск ассетов для игры

На данном этапе велась активная разработка игры с ежедневными отчётами всех участников команды.

3. Этап тестирования (2-3 недели):

- Поиск багов и их исправление.
- Полировка визуальных элементов и интерфейсов, настройка игровых сценариев.

Из-за возникших сложностей и изменений в команде, настоящие сроки выглядели иначе:

- Этап проектирования занял 2 недели согласно плану
- Этап разработки занял также 8 недель, однако из-за замедлений и остановок не получилось реализовать всё то, что планировалось
- Этап тестирования отсутствовал, так как не успели закончить этап разработки.

4.1 Примеры задач для каждого члена команды

- Георгий Александров(геймдизайнер/аналитик): разработка сюжетной линии, ведение дизайн-документа, описание механик (инвентарь, взаимодействие с NPC и т.д.), поиск ассетов деревьев, кустов, земли и т.д. для игры.
- Кирилл Петров(разработчик): реализация механик стрельбы NPC, перенос анимаций в игру.
- Евгений Гайдукевич (разработчик): обновление системы инвентаря, внедрение всплывающих подсказок, взаимодействие предметов инвентаря с объектами.
- Кирилл Мезенцев (разработчик): создание механик мыслей, AI NPC, системы CodeLock

5 Описание методологии разработки и процесса работы

Для управления процессом разработки нашей игры мы выбрали методологию Scrum, которая позволяет эффективно организовать командную работу и адаптироваться к изменениям требований. Каждый спринт длился одну неделю без выходных, и в течение спрингта мы проводили следующие встречи:

- **Четверг — загрузка задач:** распределение задач между участниками команды и уточнение целей на неделю.
- **Пятница-Воскресенье - стендап:** короткие собрания для обсуждения текущего прогресса и выявления препятствий.
- **Понедельник — демо:** демонстрация результатов работы за спринт и обсуждение достижений.

6 Архитектура программного продукта

Архитектура нашего программного продукта построена на основе модели MVC (Model-View-Controller). Этот подход позволяет разделить данные, логику приложения и интерфейс пользователя, что способствует упрощению разработки и последующего сопровождения проекта.

- **Model (Модель):** отвечает за управление данными игры, включая состояние инвентаря, характеристики персонажа и параметры окружения.
- **View (Представление):** реализует визуальное отображение данных и взаимодействие с пользователем. Все элементы интерфейса, такие как инвентарь и диалоговые окна, построены в соответствии с этой частью.
- **Controller (Контроллер):** обеспечивает связь между моделью и представлением. Например, обработка действий игрока, таких как перемещение, использование предметов или взаимодействие с NPC, проходит через контроллеры.

Благодаря архитектуре MVC команда смогла работать над различными аспектами игры параллельно. Однако из-за недостатка опыта у разработчиков и тимлида, применение архитектуры не всегда было последовательным.

7 Стек разработки

Хоть в команде нет дизайнера, приходилось использовать для разработки дизайна Photoshop, потому что он позволяет удобно и легко создавать необходимые для работы изображения, а также некоторые участники команды ранее имели опыт его использования.

Для разработки самой игры использовался Godot, поскольку он легок для освоения.

Для организации совместной разработки выбран GitHub. Он использует систему контроля версий Git, которая позволяет отслеживать и управлять изменениями в коде. GitHub позволяет нескольким разработчикам работать над одним проектом, за счет создания веток для работы над отдельными функциями, а затем их объединения. GitHub предлагает функции безопасности, такие как защита веток и автоматическое обнаружение уязвимостей в зависимостях.

Для хранения всей документации и геймдизайна используется Google Drive, поскольку он предоставляет доступ к файлам с любого устройства, подключенного к интернету, несколько пользователей могут одновременно работать над одним документом, а также имеет автоматическое сохранение версий файлов, что позволяет отслеживать изменения и возвращаться к предыдущим версиям при необходимости.

8 Индивидуальные отчеты участников команды

Петров Кирилл: В начале семестра составил график встреч (загрузка, стендапы, демо), далее в течение семестра занимался управлением командой, расставлением задач и их проверкой. А также вёл активную работу над программной разработкой проекта.

Реализовал более 10 задач, в каждой из которых было более одной механики. Основной вектор разработки был направлен на развитие технической работы игры, а также различные механики, связанные со стрельбой и статусами, а также исправление архитектуры проекта для упрощения дальнейшего расширения проекта.

Во время работы встретился с проблемами оптимизации игрового движка при работе с большими данными и большими проектами (для понимания Godot создан в основном для разработки небольших 2D платформеров), так же было проблемой, что из-за плохой коммуникации команды в прошлом семестре, проект имеет очень плохую архитектуру, из-за этого были большие проблемы при разработке, исправление архитектуры не представлялось возможным в этом семестре, соответственно, проблема осталась открытой.

Александров Георгий: В этом проекте проанализировал уже созданные документы и сеттинг с прошлого семестра, и переработал их, чтобы создать интересную игру, опираясь на техническое задание от заказчиков. Реструктурировал все документы проекта и подвел под общий стиль, полностью расписал все механики и создал таблицу с быстрым поиском механики.

Также полностью расписал сюжет первого акта игры, персонажей и их мотивы. А работа разработчиков над квестами и локациями проверялась на неточности и ошибки сюжета.

Вспоминая о проблемах, с которыми пришлось столкнуться, во время работы приходилось менять концепции механик или названия объектов либо из-за сложности реализации, либо из-за других обстоятельств, поэтому было

сложно исправлять текст во множестве документах сразу. Также из-за отсутствия дизайнера мне проходилось самому находить все ассеты для игры, просить людей на бесплатной основе создать дизайн персонажей.

Мезенцев Кирилл: В этом проекте занимался созданием таких механик, как кодовый замок, который может закрывать двери, система выносливости персонажа, подсказками для игроков, которые показываются на определённых объектах, чтобы игроки могли понять, как с ними взаимодействовать, мысли персонажей, а точнее их реплики, которые они произносят фоново, созданием логики неигровых персонажей.

Было тяжело адаптироваться с игрового движка Unity на Godot. Не всегда получалось реализовать задачу так, что бы она не занимала много времени в спринте и была работоспособной

Гайдукевич Евгений: За время работы в этом проекте разработал систему квестов, добавление, удаление задания, их вывод на экран и возможность работать с системой из другой части программы не связанной с системой квестов.

Данный участник команды разработал систему инвентаря и её графическое отображение, таким образом, что предмет мог размещаться в нескольких ячейках инвентаря, и активные слоты с категориями, куда можно положить предметы определенного класса: оружие, броня, предметы для использования.

Также сделал систему предметов таким образом, что предметы можно добавить на уровень, а персонаж подходя к ним может их подобрать и положить в инвентарь. Реализовал логику активных предметов, которые можно использовать для восстановления статусов игрока, реализовал взаимодействие предметов инвентаря с окружающим игрока миром. Реализовал Роjo класс для получения информации о предмете, который находится в слоте инвентаря. Сделал систему обмена предметами с торговцем. Создал классы оружий применяя ООП. Сделал записки, которые игрок может прочитать. Во время чтения записи игра ставится на паузу.

9 Сценарии использования

Сценарий использования описан на видео хранящемся на платформе VK
Видео.

Ссылка:https://vkvideo.ru/video-230615202_456239017?list=ln-yBImrQXVOzq3Jcc6Pk

ЗАКЛЮЧЕНИЕ

Разработка игры столкнулась с рядом трудностей, которые существенно повлияли на её завершение. Основное требование заказчика — создание играбельной демо-версии — выполнено лишь частично. Несмотря на наличие отдельных механик (инвентарь, освещение, нейтральные NPC и диалоги с ними, стрельба, квесты), их интеграция в единый игровой процесс не была завершена, что сделало невозможным представление полноценного игрового опыта. Это связано с рядом факторов, включая изменение требований на поздних этапах разработки и недостаток времени для адаптации.

На ранних этапах проекта команда демонстрировала медленную работу, однако ближе к середине разработки получилось дойти до нужного темпа. Были созданы ключевые элементы механик и определён стиль игры. Однако неопытность участников, особенно в работе с Godot, сказалась на темпах разработки. Более того, задержки с предоставлением доступов к Jira и Confluence усложнили планирование задач и первоначальную организацию работы.

Тестирование промежуточных версий игры выявило несколько ключевых проблем:

- Отсутствие связности систем:** инвентарь и боевые механики не интегрированы с персонажем, что делает игровой процесс фрагментированным.
- Недоработанные элементы геймплея:** механики взаимодействия с противниками и полноценная стрельба отсутствуют, что ограничивает возможности демонстрации проекта.
- Ограниченный визуальный контент:** хотя базовые текстуры и интерфейсы созданы, их проработка требует улучшений для соответствия финальному виду.

Эти недостатки привели к невозможности создать играбельную демо-версию, которая могла бы быть представлена потенциальным издателям.

Несмотря на трудности, у проекта есть значительный потенциал. Для улучшения качества продукта и достижения поставленных целей рекомендуется:

1. Завершение текущих механик:

- Исправление некоторых ошибок в боевых механиках, инвентаре и взаимодействии с персонажем.
- Доработка системы NPC, включая поведение и диалоги.

2. Улучшение процесса планирования:

- Использование бесплатного сервиса Kaiten для планирования задач
- Более чёткая постановка целей на каждый спринт, чтобы избежать перегрузок или недооценки задач, а также уменьшить количество встреч

3. Расширение контента:

- Поиск дизайнера и последующее увеличение разнообразия визуальных и игровых элементов.
- Добавление новых механик, которые могут усилить атмосферу игры и разнообразить геймплей.

4. Оптимизация проектирования демо:

- Сфокусироваться на создании короткой, но содержательной демо-версии, демонстрирующей ключевые аспекты игры.
- Упрощение механик для быстрой реализации, сохраняя при этом уникальность проекта.

При завершении текущих механик и устранении выявленных проблем проект может стать основой для полноценной игры. Потенциальное направление развития включает:

- **Углубление сюжета:** создание дополнительных локаций и миссий для расширения игрового мира.
- **Оптимизация кода:** улучшение производительности игры и её адаптация под разные платформы.
- **Маркетинг и продвижение:** акцент на создании качественного контента для привлечения инвесторов и формирования заинтересованной аудитории.

Проект, несмотря на текущие недостатки, обладает потенциалом благодаря запланированным механикам и атмосферой, которые выделяют его среди аналогов. Основные задачи на будущее включают завершение демо-версии и оптимизацию процесса разработки. Эти шаги позволят представить проект в выгодном свете как игрокам, так и потенциальным издателям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 2.1 Основы Git - Создание Git-репозитория - URL: <https://git-scm.com/book/ru/v2/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8Git%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%BE%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D1%8F> (дата обращения: 10.02.2025).
2. Godot Docs – 4.4 branch – URL: <https://docs.godotengine.org/en/stable/> (дата обращения: 20.02.2025).
3. Бесплатные скрипты – URL: <https://godotengine.org/asset-library/asset> (дата обращения: 25.04.2025).
4. Этапы создания игры – URL: <https://learngames-rtf.ru/> (дата обращения: 15.04.2025).
5. Решение технических проблем Godot-a – URL: <https://stackoverflow.com/questions/tagged/godot> (дата обращения: 15.04.2025).
6. Матерал для создания экономики - URL: <https://apptractor.ru/develop/kak-postroit-umnoyu-igrovuyu-ekonomiku.html> (дата обращения: 8.05.2026)
7. Дополнительный материал для создания экономики - URL: https://gdcuffs.com/monetization_in_games1/ (дата обращения: 01.05.2025)