

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий
Кафедра «Информационной безопасности»

Направление подготовки: 10.03.01 Информационная безопасность

ОТЧЕТ

по проектной практике

Студент: Шараев Иван Евгеньевич Группа: 241-334

Место прохождения практики: Московский Политех, кафедра «Информатика и информационные технологии»

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Рябчикова Анна Валерьевна

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. Общая информация о проекте	4
2. Общая характеристика деятельности организации (заказчика проекта)	5
3. Описание задания по проектной практике	7
3.1. Базовая часть	8
3.2. Вариативная часть	15
ЗАКЛЮЧЕНИЕ	25
СПИСОК ЛИТЕРАТУРЫ	26
Часы работы	28

Введение

Настоящий отчет представляет результаты проектной практики, выполненной в рамках образовательной программы по направлению «Автоматизированные системы обработки информации и управления». Практика проводилась на базе кафедры «Информатика и информационные технологии» Московского Политехнического университета и была направлена на разработку веб-сайта в соответствии с дисциплиной «Проектная деятельность», а также на исследование библиотеки C# RogueSharp и создание игры-RogueLike в рамках вариативной части задания.

Общая информация о проекте

Наименование проекта: Группа проектов игровой индустрии. Проект «Stasis» (I курс).

Актуальность проекта:

На современном игровом рынке наблюдается растущий интерес к нестандартным игровым решениям, способным предложить игрокам уникальные повествовательные и интерактивные опыты. Этот тренд отражает изменение восприятия видеоигр как не просто развлечения, а полноценной формы искусства и культурного явления, в которой значительное внимание уделяется не только геймплею, но и нарративу, взаимодействию с игроком и передаче сложных идей через виртуальный мир.

Интерактивные повествования стали важной тенденцией в последние годы. Они позволяют игрокам участвовать в создании сюжета, активно влияя на развитие истории. Такие игры отказываются от линейности классических повествований, предоставляя широкий спектр решений и моральных выборов, которые могут кардинально изменить исход событий. Это не только расширяет горизонты для разработчиков, но и предлагает игрокам возможность погружения в глубокие, иногда даже философские размышления о природе свободы воли, ответственности и последствиях действий.

Актуальность создания таких проектов обусловлена не только спросом на игры с нелинейным сюжетом, но и потребностью в играх, которые выходят за рамки традиционного понимания видеоигр. Современные геймеры хотят не просто развлекаться, но и испытывать эмоциональные и интеллектуальные вызовы. Это становится особенно важно в эпоху, когда медиа все больше влияют на восприятие реальности, предлагая нам не только испытывать эмоции, но и задумываться над глубокими вопросами.

Линейность событий, которая в традиционных видеоиграх жестко предопределяет развитие сюжета, уступает место более гибким моделям. Игроки, влияющие на сюжетные арки, получают возможность не только следовать по заранее прописанным путям, но и испытывать разнообразие сценариев в зависимости от своих решений. Это создает более глубокий уровень вовлеченности, где игроки могут изучать различные варианты развития событий и анализировать свои действия, подобно тому, как они поступали бы в реальной жизни.

Таким образом, игры с интерактивными повествованиями, предлагающие игрокам стать частью философской дискуссии, имеют огромный потенциал. Они не только соответствуют запросам современного игрового рынка, но и вносят значительный вклад в развитие видеоигр как культурного явления. В перспективе такие проекты могут стать важной вехой в эволюции игр, раскрывая их как инструмент не только для развлечения, но и для глубоких размышлений о жизни и человеческой природе.

Проблематика:

Проблематика проекта включает множество аспектов, которые требуют глубокого осмысления и решения для успешной реализации, таких как:

1. Нелинейность и сложность повествования:

Одной из главных проблем при создании игры с нелинейным сюжетом является необходимость поддерживать баланс между свободой игрока и целостностью нарратива. Непродуманная структура может привести к путанице или потере смысла. Разработчикам необходимо тщательно продумать сценарий, чтобы все возможные варианты действий игрока гармонично вписывались в общую картину, не вызывая ощущения хаоса и несоответствий в сюжете. Это особенно важно, когда сюжет включает элементы, требующие от игрока глубокого осмысления событий. Если игрок не будет понимать или оценивать развитие событий, основная идея игры может потеряться.

2. Понимание и восприятие мета-игр:

Мета-игры, а также проекты, активно разрушающие "четвертую стену", все еще остаются нишевым жанром. Несмотря на их популярность среди узкого круга игроков, их восприятие в массовом сегменте остается под вопросом. Не каждый игрок готов к анализу происходящего или к играм, которые, на первый взгляд, ломают привычные механики и ожидания. Проблема заключается в том, что такие игры могут быть неправильно поняты или восприняты как несерьезные или неоправданно сложные. Поэтому важно найти способ привлечь более широкую аудиторию, не упрощая при этом концептуальную глубину игры.

3. Целостность и однородность игрового процесса:

Из-за мультижанровой природы игры, следует соблюдать осторожность в сочетании различных жанров и уровней. Многочисленные переходы между жанрами могут сбить игрока с толку, создавая ощущение, что игра разрозненна или не структурирована. Смешение жанров, если оно выполнено плохо, может создать барьер для погружения в игровой мир. Важно, чтобы игровые механики и стилистические переходы были органичны и поддерживали цельное восприятие игры, не нарушая плавности процесса.

4. Свобода выбора и логические тупики:

Нелинейные сюжеты часто подразумевают множество выборов, которые влияют на развитие истории. Однако свобода действий может привести к тому, что игроки окажутся в логических тупиках — ситуациях, когда дальнейшее развитие событий становится непонятным или противоречивым. Это может происходить из-за недостаточной проработки возможных исходов решений или их взаимодействия. Чтобы избежать этого, разработчикам необходимо создавать механизмы, которые будут предугадывать такие ситуации и предоставлять игроку выходы, не нарушая при этом целостности сюжета.

5. Реплейабилити (повторное прохождение) и усталость от вариантов:

Нелинейные игры зачастую предполагают многократное прохождение для изучения всех возможных исходов и развития сюжета. Однако это может привести к усталости игроков от необходимости повторять уже знакомые участки игры. Чем больше различных ветвей развития событий, тем сложнее избежать повторений. Проблема здесь состоит в том, чтобы предложить уникальный и интересный опыт при каждом повторном прохождении, минимизируя скуку от уже виденных сцен и выборов.

6. Управление эмоциональной вовлеченностью игрока:

Интерактивные повествования могут создавать сильное эмоциональное вовлечение, особенно если игрок ощущает реальное влияние своих действий на мир игры и ее персонажей. Однако с этим возникает проблема: как сохранить эмоциональный отклик игрока, когда сюжет дробится на множество ветвей? Слишком много вариантов исхода может размыть эмоциональную значимость каждого из них. Необходимо найти способы сделать каждое решение важным, но не перегружать игрока количеством возможных вариантов, что может снизить эмоциональную привязанность к сюжету и персонажам.

Цель проекта: Создание циклической мультижанровой игры с опорой на оригинальный сюжет.

Задачи проекта:

1. Разработка игровой логики
2. Написание сценария
3. Создание визуальной части игры
4. Создание аудиальной части игры

Общая характеристика деятельности организации

Наименование заказчика: Московский политехнический университет

Структура организации

Институт имеет 6 филиалов: Рязанский институт (филиал) Московского политехнического университета, Чебоксарский институт (филиал) Московского политехнического университета, Коломенский институт (филиал) Московского политехнического университета, Электростальский институт (филиал) Московского политехнического университета, Тучковский филиал Московского политехнического университета и Ивантеевский филиал Московского политехнического университета.

При институте осуществляет свою работу 81 кафедра по тринадцати факультетам и институтам.

Факультеты/Институты:

1. Высшая школа печати и медиаиндустрии
2. Институт графики и искусства книги имени В.А. Фаворского
3. Институт издательского дела и журналистики
4. Полиграфический институт
5. Инженерная школа (факультет)
6. Транспортный факультет
7. Факультет базовых компетенций
8. Факультет довузовской подготовки
9. Факультет информационных технологий
10. Факультет машиностроения
11. Факультет урбанистики и городского хозяйства
12. Факультет химической технологии и биотехнологии
13. Факультет экономики и управления

Помимо этого, Московский Политех включает в себя 205 иных структурных подразделений.

Описание деятельности Московского политехнического университета

Институт занимается.

Научной деятельностью, в рамках которой сотрудничает с другими вузами России и других стран;

Проектной деятельностью, привлекая индустриальных партнёров для решения нестандартных задач;

Внеучебной деятельностью (профорганизации, ассоциации студентов).

Творческой деятельностью, организовывая выступления и различные танцевальные, песенные, театральные, тематические мероприятия со свободных входов;

Спортивной деятельностью, с завидной периодичностью завоёвывая золотые медали на Московских и Всероссийских студенческих соревнованиях по мини-футболу пауэрлифтингу, волейболу, различным боевым единоборствам, парусному спорту

Описание задания по проектной практике

Задание на проектную (учебную) практику разработано для студентов первого курса, обучающихся по направлениям подготовки, связанным с информационными технологиями и информационной безопасностью. Трудоёмкость практики составляет 72 академических часа. Задание может выполняться индивидуально или в составе группы до 3 человек. Для управления версиями будет использоваться Git, для написания документации — Markdown, а для создания статического веб-сайта — языки разметки HTML и CSS, но опционально допускается использовать генераторы статических сайтов, такие, как Hugo. В качестве платформы для размещения репозитория допустимо использовать как GitHub, так и GitVerse, что обеспечивает гибкость в выборе инструментов. Также предусмотрено взаимодействие с организациями-партнёрами, включая стажировки, которые будут приниматься к зачёту при оценке.

Задание состоит из двух частей. Первая часть(базовая) является общей и обязательной для всех студентов. Вторая часть вариативная.

Базовая часть задания включает в себя:

- 1. Настройка Git и репозитория:**
- 2. Написание документов в Markdown:**
- 3. Создание статического веб-сайта:**
- 4. Взаимодействие с организацией-партнёром:**
- 5. Отчёт по практике**

Описание задания по проектной практике

Базовая часть

Настройка Git и репозитория.

Рабочей операционной системой для выполнения базовой части задания была Windows 11. Для работы с системой управления версиями Git потребовался установщик Git for Windows. В ходе первичной настройки (параметры оставлены по умолчанию) были установлены три приложения:

- Git CMD (командная строка Git),
- Git Bash (терминал с эмуляцией Bash),
- Git GUI (графический интерфейс, удобный для новичков).

Хотя Git предоставляет собственные инструменты, все основные команды выполнялись в стандартном интерпретаторе командной строки Windows (cmd.exe).

Создание и структура репозитория.

На платформе GitHub был создан публичный репозиторий для удобного управления версиями проекта. Его структура включает:

- README.md — основной файл с описанием практики;
- reports — отчёт по практике;
- site — исходный код сайта.

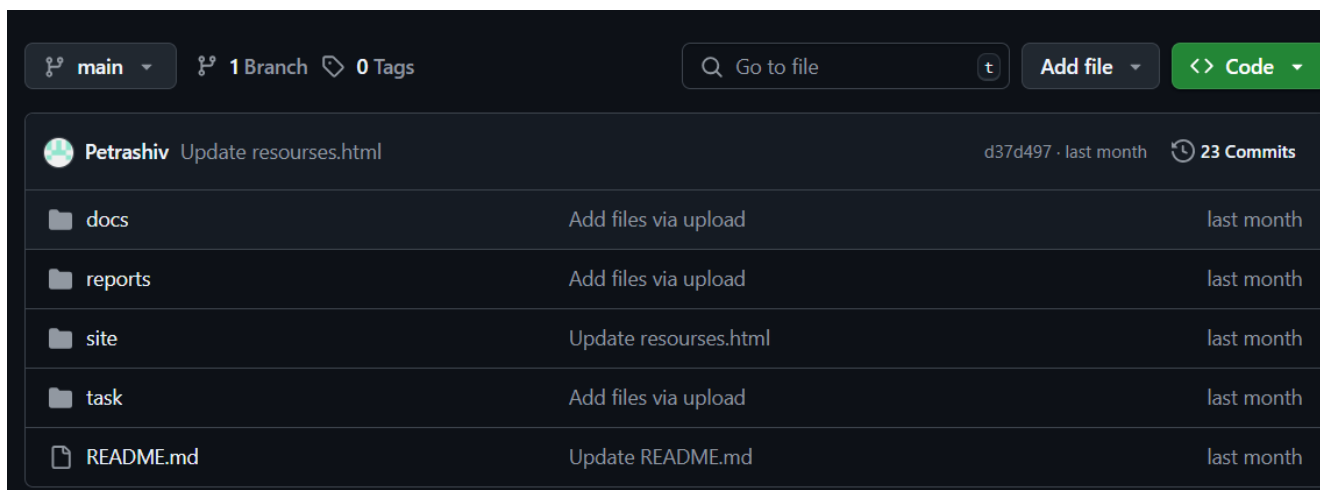


Рис 2. Структура репозитория

Работа с Git: фиксация изменений.

Каждое значимое изменение фиксируется (commit) в три этапа:

1. Добавление в индекс (git add).

- Чаще всего используется `git add .` (добавляет все изменённые файлы).

2. Создание коммита (git commit -m "описание").

- Комментарий помогает быстро находить нужные версии через `git log`.

3. Отправка на удалённый репозиторий (git push).

- Вместо `git push --all origin` использовалась более точная команда `git push origin branch:branch` для избежания конфликтов.

Документация в Markdown.

Все текстовые файлы проекта (.md) написаны в Markdown — удобном формате для статей, отчётов и документации. README.md — ключевая информация о практике,

Дополнительные Markdown-файлы могут добавляться для пояснения структуры папок или описания компонентов проекта.

Создание статического веб-сайта.

Мы разделили эту задачи на 4 подзадачи:

1. Сбор информации
2. Разработка стилей
3. Разработка сайта

Разработка стилей

Стили сайта были написаны вручную с использованием CSS, чтобы обеспечить полный контроль над внешним видом и анимациями. Основные этапы разработки:

1. Базовая настройка — сброс стандартных отступов (``margin``, ``padding``) и применение ``box-sizing: border-box`` для корректного расчета размеров.
2. Глобальные стили — задан плавный переход (``transition``) для всех интерактивных элементов и шрифт по умолчанию (``Segoe UI``).
3. Цветовая схема — использованы градиенты (сине-голубые для хеддера и подвала) и нейтральные оттенки для фона.
4. Типографика — выравнивание текста, межстрочные интервалы (``line-height``) и акцентные заголовки с бордерами.
5. Адаптивная верстка — гибкая сетка (``grid``) и медиазапросы для корректного отображения на мобильных устройствах.
6. Эффекты взаимодействия — плавное поднятие карточек (``transform: translateY``) и изменение цвета ссылок при наведении.

7. Тени и скругления — добавление ``box-shadow`` и ``border-radius`` для современного объемного дизайна.

8. Липкое меню — фиксация навигации при скролле (``position: sticky``).

Взаимодействие с организацией-партнёром

23 апреля 2025 года состоялась онлайн-конференция, организованная компанией «Эдит Про» — ведущим разработчиком решений в области цифровизации и ИТ. Мероприятие собрало представителей различных организаций и стало площадкой для обсуждения ключевых аспектов современной ИТ-индустрии.

Наиболее запоминающимся мероприятием стала деловая игра "Диалог с Заказчиком", наглядно продемонстрировавшая реалии клиентского взаимодействия. Участники разделились на команды по 7-10 человек, и каждая получила таблицу для работы.

Сначала команды выступали в роли поставщиков услуг. Нам нужно было сформулировать один ключевой вопрос к гипотетическому заказчику, который помог бы понять его потребности перед началом сотрудничества. Вопросы записывались в первый столбец таблицы – например, о целях проекта, бюджете или сроках. После этого листы передавались другим командам, которые теперь становились "заказчиками".

Здесь началась самая интересная часть. Каждая команда-заказчик получала случайно назначенный стиль поведения. Кто-то должен был отвечать предельно кратко, буквально одним словом. Другие, наоборот, писали пространные, витиеватые ответы, полные общих фраз, но без конкретики. Некоторым достались роли скептических или непоследовательных заказчиков, чьи ответы могли быть полны недоверия или внутренних противоречий. Эти ответы вносились во второй столбец таблицы напротив исходного вопроса.

Затем листы вернулись к первоначальным командам. Наша задача резко усложнилась: нужно было разработать детальный план поездки для этого

заказчика – выбрать место назначения, рассчитать бюджет, спланировать посещаемые места и питание – основываясь только на том ответе, который мы получили. Столкнуться с этим вживую было невероятно поучительно. Если заказчик отвечал односложно, приходилось строить догадки и делать смелые допущения. Если ответ был длинным, но бессодержательным – вычленять крупными буквами полезной информации из потока слов. Противоречивые ответы заставляли искать компромиссы и расставлять приоритеты.

Финальный этап – презентация планов и обсуждение – ярко показал, насколько по-разному можно интерпретировать одни и те же слова заказчика. Планы поездок у команд, получивших схожие по стилю ответы, могли кардинально отличаться. Это наглядно доказало, как легко ошибиться в понимании истинных потребностей клиента, особенно когда его манера общения неидеальна.

Польза от этой игры оказалась огромной. Она не просто рассказала, а буквально заставила прочувствовать, с какими типами заказчиков можно столкнуться в реальности и какие трудности возникают при сборе требований. Мы осознали, насколько критично уметь задавать точные вопросы, "читать между строк", делать обоснованные предположения в условиях неопределенности и искать рабочие компромиссы. Игра отлично развила гибкость мышления и показала, что успех проекта часто зависит не только от технических навыков, но и от умения расшифровать и удовлетворить запросы клиента, какими бы сложными они ни были.

По завершении конференции приятным бонусом стало объявление от компании «Эдит Про». Они пригласили всех заинтересованных участников на стажировку, предоставив контакты HR-отдела. Это предложение вызвало живой отклик среди присутствующих.



Рис 3. Участники конференции вместе с представителями компании

Вариативная часть

Разработка RogueLike игры на C#

1. Введение

Данный отчет описывает процесс разработки прототипа Rogue-Like игры в фэнтезийном сеттинге, выполненной на движке Unity с использованием библиотеки RogueSharp. Основной целью проекта было создание динамичной игры с процедурной генерацией подземелий, сбором ресурсов, боем с монстрами и исследованием многоэтажных локаций. Ключевой особенностью реализации стал отход от классической пошаговости в пользу режима реального времени для передвижения игрока и противников, что потребовало адаптации подходов, типичных для жанра. Библиотека RogueSharp использовалась как фундамент для генерации и логического представления игрового мира.

2. Архитектура и генерация игрового мира

Основу генерации игрового пространства составлял скрипт MapGenerator. Его ключевой метод GenerateMap() отвечал за создание экземпляра класса DungeonMap из библиотеки RogueSharp. Этот объект DungeonMap содержал всю логическую информацию о структуре уровня: расположение стен, пола, проходимых клеток и комнат. Для визуализации сгенерированной карты в Unity использовался циклический обход каждой клетки логической карты DungeonMap. На основе типа клетки (стена, пол, проходимая зона) в соответствующих координатах игрового поля Unity происходила расстановка префабов тайлов – блоков стен и пола. Такой подход обеспечил четкое соответствие между логическим представлением уровня в RogueSharp и его визуальным воплощением на сцене Unity.

3. Процедурное размещение объектов и механики взаимодействия

Генерация интерактивных объектов на уровне реализовывалась после создания базовой карты. Количество сундуков с золотом определялось случайным образом в диапазоне от 0 до 3 на этаж. Для каждого сундука случайным образом выбиралась одна из сгенерированных комнат (представленных объектами типа Rectangle). Внутри выбранной комнаты с помощью метода GetRandomWalkableLocationInRoom() находилась случайная проходимая клетка. Важным условием было предотвращение наложения объектов: позиция сундука проверялась на то, чтобы она не совпадала с позицией люка.



Рис 4. Генерация сундуков на этаже

Сам люк генерировался один на этаж, аналогичным образом размещаясь в случайной проходимой клетке случайной комнаты. Взаимодействие игрока с сундуком при нахождении рядом и нажатии клавиши приводило к его «открытию». Это действие сопровождалось появлением всплывающего анимированного текста (реализованного через Unity Animator), отображавшего количество полученных золотых монет. Взаимодействие с люком служило переходом на следующий этаж подземелья. Технически это реализовывалось через перезагрузку текущей игровой сцены. Для сохранения прогресса игрока (такого как здоровье, уровень, собранное золото) между перезагрузками сцен использовались механизмы сохранения состояния (например, статичные классы или DontDestroyOnLoad). Чтобы скрасить неизбежную задержку при регенерации нового этажа, была реализована экранная загрузка (loading screen), отображавшаяся в момент перехода.

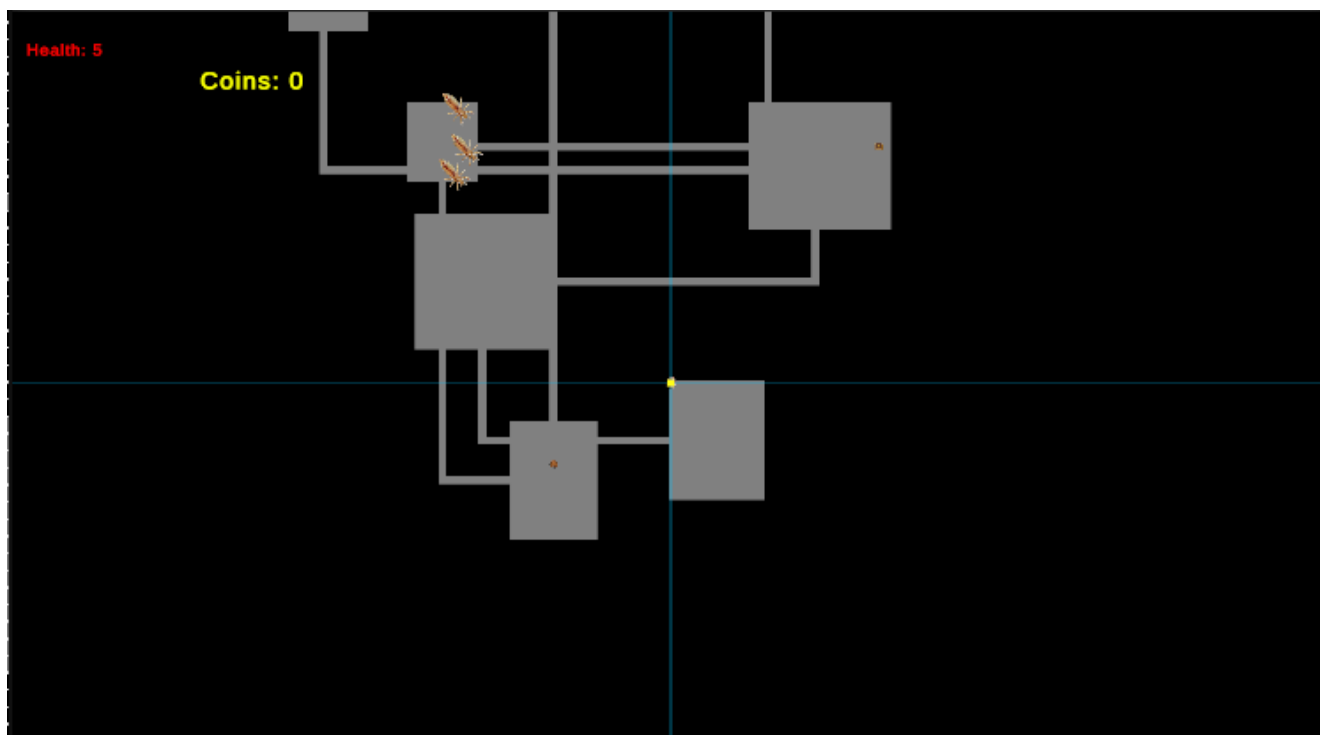


Рис 5. Процедурная генерация карты

4. Особенности геймплея и технические аспекты

Отличительной чертой проекта стало использование режима реального времени для управления игроком и поведения противников, что контрастирует с традиционными пошаговыми Rogue-Like, для которых часто выбирают RogueSharp. Это потребовало адаптации систем управления и ИИ, изначально заточенных под пошаговость. Механика сбора золота из сундуков с визуальной обратной связью в виде анимированного текста стала важным элементом положительного подкрепления игрока. Система переходов между этажами через люки, несмотря на техническую перезагрузку сцены, обеспечивала ощущение непрерывного исследования многоуровневого подземелья благодаря сохранению ключевых параметров игрока и наличию экрана загрузки. Работа с DungeonMap из RogueSharp в качестве источника истины для позиционирования объектов (сундуков, люков) и проверки проходимости клеток доказала свою эффективность.



Рис 6. Экран загрузки уровня

Реализована система монет, позволяющая игроку зарабатывать внутриигровую валюту в процессе сражений. Валюта используется для улучшения характеристик персонажа через продуманную и визуально оформленную систему прокачки.

Был создан загрузочный экран, обеспечивающий плавные переходы между сценами, что способствует более целостному восприятию игрового процесса. Интерфейс игрока получил особое внимание — он информативен, эстетичен и адаптирован под различные этапы игры. Также реализовано постоянное отображение текущих характеристик персонажа, что помогает игроку отслеживать свой прогресс.

Особое внимание уделялось плавности взаимодействия и визуальной обратной связи при совершении действий, что повышает общее качество восприятия игры.

Заключение

В ходе проектной практики были успешно выполнены все поставленные задачи, включая базовую и вариативную части. Освоены ключевые технологии веб-разработки: работа с Git для контроля версий, создание структурированного репозитория, оформление документации в Markdown, а также разработка адаптивного веб-сайта с использованием HTML и CSS.

Основные задачи проекта успешно выполнены: освоена и применена библиотека RogueSharp для процедурной генерации логической структуры уровней (комнат, стен, проходимых зон); реализована система размещения интерактивных объектов (сундуков, люков) с учетом условий непересечения и доступности; разработаны механики взаимодействия с объектами (анимация открытия сундука, сбор золота с визуальной обратной связью, переход между этажами); внедрена система сохранения состояния игрока при смене уровней и экран загрузки.

Проект позволил получить практический опыт работы с процедурной генерацией, интеграцией логической библиотеки (RogueSharp) с движком реального времени (Unity), разработкой интерактивных механик и управлением состоянием игры при смене сцен. Созданный прототип демонстрирует жизнеспособность подхода и служит основой для дальнейшего развития: добавления разнообразных врагов и их ИИ, внедрения системы инвентаря и предметов, расширения вариативности процедурной генерации, а также улучшения визуальной составляющей и эффектов.

Участие в конференции от компании «Эдит Про» дало ценные инсайты о современных трендах в области информационных технологий. Знания, полученные в ходе мероприятия, были применены при работе над проектом.

Практика также способствовала развитию навыков командной работы, планирования и распределения задач. Опыт, полученный при работе с Unity, C# и библиотекой RogueSharp, является ценным для дальнейшей деятельности в области игровой разработки.

Список литературы

1. Unity Technologies. Документация Unity: Руководство пользователя [Электронный ресурс]. – Режим доступа: <https://docs.unity.com/> (дата обращения: 20.05.2025).
2. Faron Bracy. RogueSharp Documentation [Электронный ресурс]. – Режим доступа: <https://roguesharp.wordpress.com/> (дата обращения: 20.05.2025).
3. Herbert Wolverson. Hands-on Rust: Effective Learning through 2D Game Development and Play. – Raleigh, NC: Pragmatic Bookshelf, 2021. – 284 с.
4. Tanya X. Short, Tarn Adams. Procedural Generation in Game Design. – Boca Raton, FL: CRC Press, 2017. – 320 с.
5. Jesse Schell. Искусство геймдизайна. Книга линз / Пер. с англ. – СПб.: Питер, 2021. – 672 с.
6. Jake Ströh. Unity in Action: Multiplatform game development in C#. – *3rd ed. – Shelter Island, NY: Manning Publications, 2022*. – 480 с.
7. Adam Boduch. Game Development Patterns with Unity 2021: Explore practical game development using software design patterns and best practices in Unity and C#. – *2nd ed. – Birmingham, UK: Packt Publishing, 2021*. – 420 с.
8. Unity Technologies. Saving and Loading Data in Unity [Электронный ресурс]. – Режим доступа: <https://learn.unity.com/tutorial/saving-and-loading-data> (дата обращения: 20.05.2025). – (Статья на Unity Learn).
9. Adam Millard. The Game Maker's Toolkit [YouTube Channel] [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/@GMTK> (дата обращения: 20.05.2025). – (Канал YouTube).
10. Марк Грегори. Разработка игр на Unity. Справочник / Пер. с англ. – М.: ДМК Пресс, 2022. – 500 с.

Часы работы

1. Подготовительный этап (17 часов):

- Изучение задания и требований;
- Ознакомление с Git, Markdown, HTML/CSS;

2. Работа с Git и документацией (3 часа):

- Настройка Git, создание репозитория;
- Написание README.md и структуры проекта.

3. Верстка сайта (10 часов):

- Изучение адаптивной вёрстки (HTML/CSS);
- Создание стилей.

4. Взаимодействие с организацией-партнёром (6 часов)

5. Вариативная часть: Разработка RogueLike игры на C# (40 часов):

- Изучение теории;
- Изучение библиотеки RogueSharp.

6. Написание отчёта по практике (6 часов).

Приложения

1. GitHub репозиторий проектной практики <https://github.com/Mr-Paket/mospol-practise>
2. GitHub репозиторий проекта из вариативной части задания <https://github.com/Petrashiv/VeinOfTheForgotten>