

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра программирования и информационных технологий

Техническое задание  
на разработку мобильной игры  
«Приключенческая игра с коллекционированием «Fishventure»»

Исполнители

\_\_\_\_\_ Богачев-Воевудский А.

\_\_\_\_\_ Веремеев В.

\_\_\_\_\_ Габелко М.

\_\_\_\_\_ Штукатуров Д.

\_\_\_\_\_ Елфимов А.

\_\_\_\_\_ Дубровин Д.

Заказчик

\_\_\_\_\_ Тарасов В.С.

Воронеж 2025

# СОДЕРЖАНИЕ

Термины и определения .....	5
1 Предпроектное исследование .....	9
1.1 Анализ рынка мобильных игр .....	9
1.1.1 Глобальный рынок игр .....	9
1.2 Анализ конкурентов.....	9
1.2.1 Stardew Valley .....	9
1.2.2 Fishing Planet.....	10
1.2.3 Fishdom.....	11
1.3 Целевая аудитория и рынок .....	11
1.3.1 Портрет потребителя .....	11
1.3.2 Сегментация по возрасту.....	12
1.3.3 Анализ региона распространения .....	14
1.3.4 Оценка SAM (Serviceable Addressable Market) .....	15
1.3.5 Оценка SOM (Serviceable Obtainable Market).....	15
1.4 Дорожная карта .....	16
2 Техническое задание.....	18
2.1 Цели и задачи проекта .....	18
2.1.1 Цели .....	18
2.1.2 Задачи .....	18
2.1.3 Критерии успешности .....	18
2.2 Функциональные и нефункциональные требования.....	19
2.2.1 Функциональные требования .....	19
2.2.2 Нефункциональные требования .....	21
2.3 Потенциальные риски.....	22
2.4 Пользовательские сценарии.....	22
2.4.1 Начало новой игры.....	22
2.4.2 Ловля рыбы.....	23
2.4.3 Улучшение удочки.....	23
2.4.4 Поимка легендарной рыбы .....	24
2.4.5 Исследование мира .....	24

2.4.6 Продажа рыбы .....	25
2.4.7 Выбор наживки.....	25
2.5 Перечень основных функциональных блоков системы.....	26
2.5.1 Система управления игроком .....	26
2.5.2 Система рыбалки.....	26
2.5.3 Система улучшений .....	26
2.5.4 Система магазина .....	27
2.5.5 Система поимки легендарной рыбы .....	27
2.5.6 Система сохранения.....	27
2.5.7 Система коллекционирования .....	28
3 Начальная архитектура .....	29
3.1 UML диаграммы.....	29
3.2 ER диаграмма.....	29
3.3 Схема API.....	30
3.3.1 Регистрация пользователя.....	30
3.3.2 Авторизация пользователя.....	30
3.3.3 Обновление токена .....	31
3.3.4 Отправка логов .....	31
3.3.5 Отправка метрик игрока.....	32
3.3.6 Генерация записок .....	33
3.4 Структура игры .....	33
3.4.1 Основные компоненты .....	33
3.4.2 Взаимодействие компонентов .....	34
3.5 Стек технологий .....	34
3.5.1 Backend.....	34
3.5.2 Игровой движок: .....	34
3.5.3 Хранение данных: .....	35
4 Дизайн макеты .....	36
4.1 UI Kit.....	36
4.1.1 Цветовая палитра: .....	36
4.1.2 Цветовые акценты:.....	36
4.1.3 Шрифты: .....	36

4.1.4 Компоненты:.....	37
4.2 Брендбук.....	37
4.2.1 Логотип: .....	37
4.2.2 Фирменный стиль: .....	37
4.2.3 Анимации:.....	38
4.2.4 Звуки:.....	38
4.2.5 Дополнительные элементы: .....	39
4.2.6 Элементы декора:.....	39
5 Организация проекта .....	40
5.1 Настройка гита .....	40
5.1.1 Гит репозиторий.....	40

## Термины и определения

**Десктоп** — компьютер.

**Пиксель-арт** — графика, стилизованная под старые игры, где графика состояла из крупных пикселей.

**RPG (Role Play Game)** — тип игр, в котором игрок отыгрывает определенного персонажа и ему предоставлен выбор действий.

**FPS (Frame Per Second)** — количество кадров в секунду, наглядно отображающее производительность на устройстве.

**Юзабилити** — удобство / пригодность использования приложения.

**Логирование** — это фиксация событий в работе приложения, помогающая его разработчикам выявлять баги системы.

**User stories** — пользовательские сценарии, использующиеся для планирования сессии пользователя приложения.

**Acceptance Criteria** — критерии приемки, важная практика для улучшения коммуникации между разработчиками и заказчиками, а также неотъемлемая часть создания качественных пользовательских сценариев.

**UI Kit** — набор готовых решений пользовательского интерфейса.

**Иконографика** – это графический способ подачи информации, данных и знаний, целью которого является быстро и чётко преподнести сложную информацию.

**Лор мира** — совокупность историй, персонажей, особенностей мира и мифологии, создающих глубокий и убедительный контекст для игрового процесса.

**Геймплей** — совокупность игровых механик, правил и взаимодействий, составляющих основу игрового процесса.

**База данных (БД)** — организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

**API** — программный интерфейс, т.е. описание способов взаимодействия одной компьютерной программы с другими.

**Large Language Models, LLM** — языковая модель, состоящая из нейронной сети со множеством параметров (обычно миллиарды весовых коэффициентов и более), обученной на большом количестве неразмеченного текста с использованием обучения без учителя.

**Локация** — участок игрового мира, доступный для исследования игроком.

**Мини-игра** — интерактивная механика, направленная на выполнение конкретной задачи.

**Механика** — это базовый элемент или система правил, определяющая, как игрок взаимодействует с игровым миром.

**Тайлы** — это способ организации игрового мира с помощью сетки из квадратов или шестиугольников, каждый из которых представляет собой отдельный элемент игрового пространства.

**Достижения** — это специальные награды или знаки отличия, которые игрок получает за выполнение определённых задач, достижение целей или прохождение определённых этапов в игре.

**Инвентарь** — это система управления предметами, которые игрок собирает, использует или хранит в процессе игры.

**Паттерн** — это заранее определённая последовательность действий или поведение, которое следует игровой механике или логике.

**Мини-игра** — это небольшая, самостоятельная игра внутри основной игры или отдельный игровой режим, который предлагает уникальный или сосредоточенный игровой опыт.

**Промт** — это входной текст или инструкция, которую пользователь предоставляет языковой модели для генерации ответа или выполнения задачи.

**Эндпоинт** — это конкретный адрес или URL, через который взаимодействуют клиент и сервер. Эндпоинты определяют, какие ресурсы или функции доступны через API, и как с ними можно взаимодействовать.

**Rest API** — это архитектурный стиль для создания веб-сервисов, который использует стандартные HTTP-методы для взаимодействия между клиентом и сервером.

**Микросервис** — это архитектурный стиль, при котором приложение разбивается на небольшие, независимые сервисы, каждый из которых выполняет одну конкретную задачу.

**Unity** — это популярный игровой движок, используемый для создания видеоигр и интерактивных приложений.

**Слайдеры** — это элементы интерфейса, позволяющие пользователям выбирать значения в пределах заданного диапазона, перемещая ползунок.

**Спрайт** — это двухмерное изображение или анимация, используемая в видеоиграх для отображения персонажей или объектов.

**Брендбук** — это документ или часть документации, содержащая руководства по использованию фирменного стиля компании, включая логотипы, цвета, шрифты и другие элементы дизайна.

**Текстура** — это изображение, применяемое к поверхности спрайтов или других игровых объектов для придания им визуального стиля и деталей. Текстуры помогают создать уникальный визуальный стиль игры, добавляя цвета, узоры и другие визуальные элементы.

**Core-механика** — это основные правила и принципы, определяющие игровой процесс и взаимодействие с игрой.

**Баланс игры** — это настройка игровых элементов таким образом, чтобы игровой процесс был справедливым и увлекательным для всех игроков.

**Qwen** — это большая языковая модель, разработанная Alibaba Cloud. Она поддерживает различные языки и может выполнять различные задачи, связанные с обработкой естественного языка.

**Confluence** — это инструмент для совместной работы и управления знаниями, разработанный компанией Atlassian, который позволяет командам создавать, организовывать и обсуждать работу в едином пространстве.

**MVP (Minimum Viable Product)** — это концепция в разработке продуктов, особенно в стартапах и инновациях, которая подразумевает создание продукта с минимальным набором функций, достаточным для удовлетворения первичных потребностей пользователей и проверки основных бизнес-гипотез.

**Реиграбельность** — это способность игры поддерживать интерес и мотивацию игрока к повторному прохождению или возвращению в игру.

**Игрок** — это пользователь приложения.



## **1 Предпроектное исследование**

### **1.1 Анализ рынка мобильных игр**

#### **1.1.1 Глобальный рынок игр**

Sensor Tower опубликовала новый отчет "State of Mobile 2025", охватывающий 10 отраслей и 23 рынка, в котором делаются выводы о состоянии рынка мобильных приложений. Отмечается тренд развития инди-сегмента и проектов, связанных с ИИ. В 2024 году в их отчете утверждается, что мировая IAP-выручка рынка мобильных игр составила 80,9 млрд долларов, что на 4% выше прошлого года.

Факторы рынка:

- В 2024 мировая IAP-выручка рынка мобильных игр составила 80,9 млрд долларов.
- Снижение количества новых игр в топе, но стабильность по загрузкам. С 2020 года число новых игр в топ-1000 сократилось с 200 до 100, однако среднее число загрузок на игру остается стабильным (1–2 млн). Это указывает на смещение фокуса с количества на качество, что выгодно инди-разработчикам.
- Наблюдается глобализация рынка: 80% загрузок и расходов приходится на приложения, выпущенные вне страны разработчика. Это позволяет инди-командам масштабироваться на международные рынки.

### **1.2 Анализ конкурентов**

#### **1.2.1 Stardew Valley**

Описание и специфика:

- Фермерский симулятор с рыбалкой, пиксель-арт, RPG-элементы.
- Нет внутриигровых покупок, только цена за саму игру.
- Рыбалка является одной из механик игры, но не основной.
- С 2019 года имеет мобильную версию.

Платформа: ПК, мобильные устройства IOS/Android

Вывод: фанаты этой игры могут стать потенциальной аудиторией нашей игры.

### **1.2.2 Fishing Planet**

Описание и специфика:

- Реалистичный многопользовательский онлайн-симулятор рыбалки от первого лица.
- Динамическая погода.
- Динамическая вода, меняющаяся в зависимости от ветра, течения и глубины.

Платформа: ПК, мобильные устройства IOS/Android

Вывод: очень реалистичный симулятор рыбалки может быть слишком сложен для большинства игроков

### **1.2.3 Fishdom**

Описание и специфика:

- Гибридный жанр – сочетание «три в ряд» (match-3) и симулятора аквариума.
- Fishdom — бесплатная игра, но с внутриигровыми покупками и монетизацией через рекламу.
- Развитие аквариума – игроки зарабатывают монеты, чтобы покупать рыб, декор и улучшать подводный мир.
- Разнообразие уровней – сотни головоломок с разными целями.

Платформа: ПК, мобильные устройства IOS/Android.

Вывод: Простая игра с головоломками и уходом за рыбками, но может надоесть из-за повторяющихся уровней.

## **1.3 Целевая аудитория и рынок**

### **1.3.1 Портрет потребителя**

Исследования аудитории мобильных игр показывают, что основная аудитория – 18–34 года, но растет доля игроков старше 35. Дети (до 13 лет) тоже играют, но их доля зависит от региона (в некоторых странах ограничены платежи).

В зависимости от пола распределение игроков в мобильные игры между мужчинами и женщинами примерно равное, но с небольшими различиями в зависимости от региона и жанров игр:

- Женщины чаще играют в казуальные и социальные игры
- Мужчины – в экшн и стратегии

### 1.3.2 Сегментация по возрасту

Таблица 1 - Возрастные группы

Портрет сегмента	Описание, специфика	Боли, потребности
Дети и подростки (12–17 лет)	<p>Зачастую владеют только мобильным устройством.</p> <p>Любят простые, но увлекательные механики (коллекционирование, прокачка).</p> <p>Низкий доход или вообще никакой.</p> <p>Высокая вовлеченность, ценят достижения, соревновательные элементы.</p>	<p>Нет денег на платные игры, следовательно необходима низкая цена или бесплатная модель распространения.</p> <p>Необходимость проявить себя в сравнении с другими, следовательно нужны соревновательные элементы.</p>

Молодые взрослые (18–30 лет)	<p>Студенты и офисные работники, играющие в свободное время.</p> <p>Ценят глубину геймплея (RPG-элементы, исследование мира).</p> <p>Готовы платить за контент, если игра увлекает.</p>	<p>Мало времени, следовательно важно автосохранение и возможность играть урывками.</p> <p>Хотят чувствовать прогресс, следовательно нужна система улучшений, открытие новых локаций.</p>
Взрослые (30+ лет)	<p>Люди, которые играют для релакса после работы.</p> <p>Чаще выбирают игры без жестких дедлайнов и стресса.</p> <p>Менее активны в соревновательных аспектах, но ценят атмосферу.</p>	<p>Не любят агрессивную монетизацию, следовательно лучше подписка или разовые покупки.</p> <p>Хотят погружаться в мир, следовательно важны лор, генерация историй, атмосфера.</p> <p>Удобство, следовательно поддержка планшетов, крупный интерфейс.</p>

Исходя из анализа рынка, было принято решение выбрать возраст целевой аудитории 12+, потому что эта возрастная группа обладает высокой вовлеченностью, ценит и предпочитает бесплатные игры или игры с низкой ценой, что соответствует нашей модели монетизации. Кроме того, возрастная категория 12+ охватывает все необходимые возрастные сегменты, включая подростков, молодых взрослых и взрослых, что позволяет максимально увеличить охват аудитории и привлечь как можно больше игроков.

### 1.3.3 Анализ региона распространения

В качестве региона распространения игры было принято решение сосредоточиться на российском рынке. Это позволит более эффективно использовать маркетинговые ресурсы и лучше понять потребности игроков.

Объем российского рынка мобильных игр в 2024 году достиг \$247,2 млн, увеличившись на 12%. Это позволило России занять пятое место в мире по количеству загрузок игр и 36-е по доходам от внутриигровых покупок. Такие данные в конце января 2025 года представила исследовательская компания Sensor Tower.

Особенности российского рынка:

— Рост рынка

- а) Российский рынок мобильных игр демонстрирует стабильный рост, что делает его привлекательным для разработчиков.
- б) Увеличение доходов на 12% говорит о потенциале для новых игр.

— Аудитория

- а) Основная аудитория мобильных игр в России — это люди в возрасте 18–34 лет.
- б) Растет доля игроков старше 35 лет.
- в) Дети (до 13 лет) также играют, но их доля зависит от региона и ограничений на платежи.

— Монетизация

- а) Российские игроки предпочитают бесплатные игры с внутриигровыми покупками или рекламой.
- б) Платные игры могут быть менее популярны, но при правильном маркетинге могут найти свою аудиторию.

#### **1.3.4 Оценка SAM (Serviceable Addressable Market)**

Факторы влияющие на оценку:

- Мобильные геймеры в России: примерно 20 млн человек.
- Доля интересующихся жанром: 0.5–1.5% (меньше, чем у бесплатных игр, так как платный барьер снижает интерес).

Итог:  $SAM = 20 \text{ млн} * 1\% = 200 \text{ тыс. потенциальных игроков.}$

#### **1.3.5 Оценка SOM (Serviceable Obtainable Market)**

Факторы влияющие на оценку:

- Маленький маркетинговый бюджет.
- Конкуренция со стороны популярных игр.
- Платность игры снижает вероятность установки.
- Цена – маленькая цена может подтолкнуть к покупке.

Итог:  $SOM = 200 \text{ тыс.} * 0.25\text{--}2.5\% = \text{от } 500 \text{ до } 5000 \text{ потенциальных установок за первый год.}$

### 1.4 Дорожная карта

Дорожная карта составлена на период 2025 года, в течении которого, основная часть проекта будет завершена к июню, а дальше будет поддержка проекта и выпуск дополнительно контента для удержания аудитории.

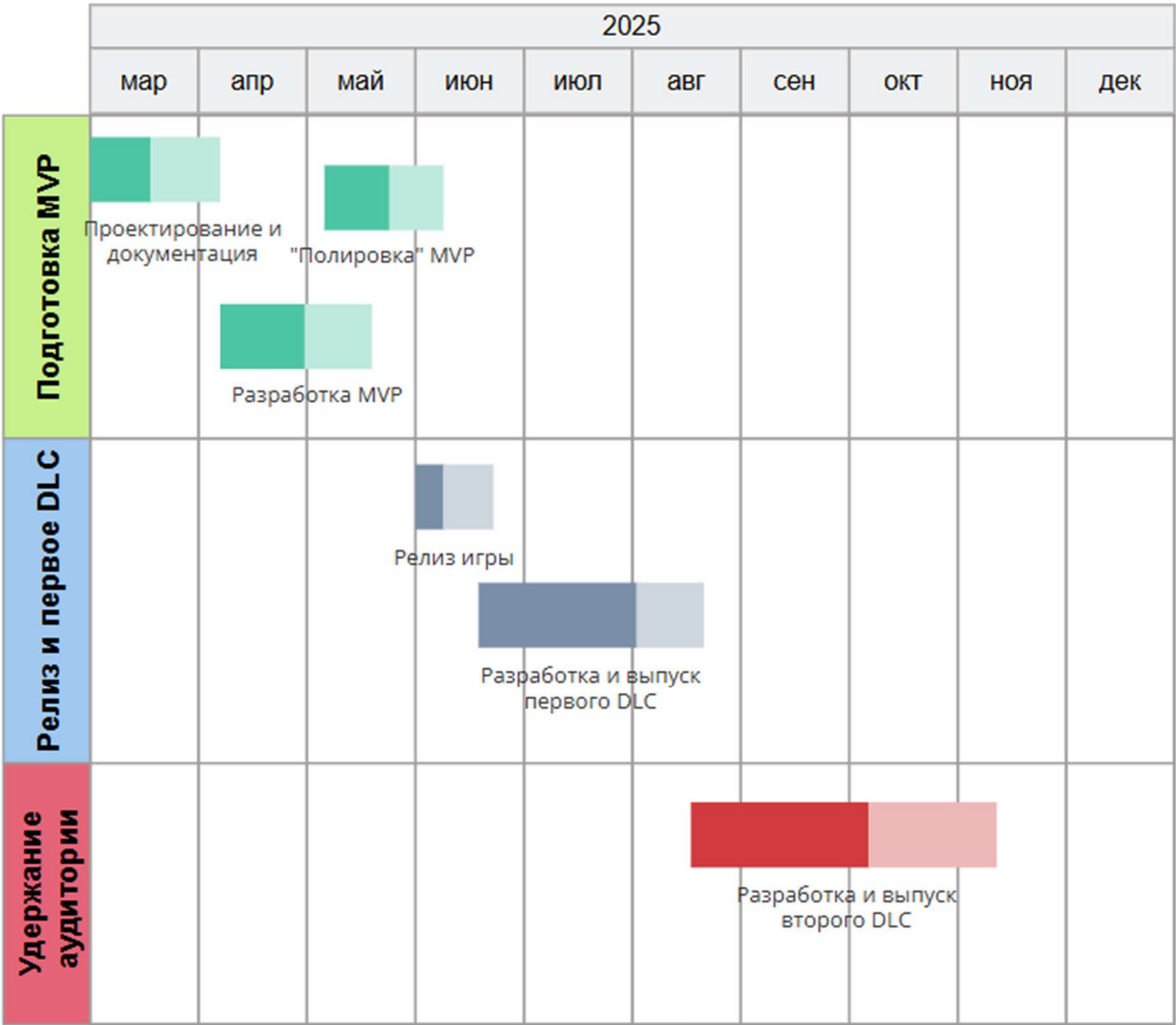


Рисунок 1 — Roadmap проекта





## **2 Техническое задание**

### **2.1 Цели и задачи проекта**

#### **2.1.1 Цели**

Поставленные цели:

- Создать игру в жанре «рыбалка», сочетающую коллекционирование, исследование мира и улучшение инструментов.
- Использовать LLM для генерации уникального лора игрового мира для мотивирования игрока исследовать его и повышения реиграбельности.
- Монетизировать игру путем единоразовой покупки.

#### **2.1.2 Задачи**

Задачи для достижения целей:

- Разработать игру на движке Unity, сочетающую рыбалку, коллекционирование, исследование мира и улучшение инструментов.
- Реализовать механизм генерации лора мира игры на основе LLM Qwen 2.5.
- Выпустить игру в качестве платного продукта в магазин мобильных приложений RuStore.

#### **2.1.3 Критерии успешности**

Критерии оценки успешности игры:

- Удержание игроков на 30 минут – 1 час в день.
- Позитивные отзывы о графике и игровом процессе.
- Сумма полученного дохода от продажи игры.

## 2.2 Функциональные и нефункциональные требования

### 2.2.1 Функциональные требования

В системе должны быть реализованы:

- **Система сохранения прогресса:** позволяет игроку продолжить игру с момента последнего сохранения. Прогресс сохраняется автоматически при выходе из игры или в процессе игрового сеанса и хранится локально на устройстве игрока.
- **Мини-игра рыбалки:** позволяет игроку ловить рыбу через выполнение интерактивной механики рыбалки, включающей: шкалу прогресса, последовательность стрелочек и таймер. Шкала прогресса отображает текущий прогресс ловли рыбы. Игрок должен повторять последовательность стрелочек в заданном порядке, выполнение каждой последовательности ограничено таймером (один цикл). Процент правильного выполнения последовательности влияет на заполнение или опустошение шкалы прогресса. Если шкала полностью заполняется, рыба считается пойманной; если она становится пустой, рыба срывается.

- **Параметры сложности рыбалки для каждой рыбы:** для каждого вида рыбы задаются параметры, определяющие сложность её ловли. Эти параметры включают штраф за провал цикла (на сколько уменьшается шкала прогресса при ошибке), очки прогресса за успешный цикл (на сколько заполняется шкала прогресса при правильном выполнении), время на выполнение цикла (длительность таймера) и длину последовательности стрелочек (количество действий, которые игрок должен выполнить).
- **Возможность начать ловлю рыбы с любой части водоёма:** возможность начать рыбалку с любого тайла суши, прилегающего к водоёму.
- **Выбор наживки:** возможность покупать разные виды наживки в магазине и выбирать их перед началом рыбалки. Выбор наживки влияет на вероятность ловли определённых видов рыб.
- **Система улучшения удочки:** в приложении будет возможность, в процессе исследования находить улучшения для удочки в сундуках или покупать их в магазине. Эти улучшения влияют на параметры мини-игры рыбалки, такие как штраф за ошибку, очки прогресса или время на выполнение цикла.
- **Продажа рыбы:** продавать пойманную рыбу в магазине за внутриигровую валюту. Цена рыбы будет зависеть от её вида.
- **Исследование мира:** для мотивации игрока исследовать локации в них размещаются сундуки, содержащие уникальные улучшения и записки с историей мира (лор). Сундуки расставлены по локации в заранее определённых точках. Некоторые сундуки доступны сразу, а другие требуют решения головоломок или выполнения дополнительных условий. Чтобы открыть сундук, необходимо подойти на соседний от него тайл.

- **Система инвентаря:** в инвентаре персонажа хранятся все пойманные рыбы. Возможность продавать рыбу в магазине за внутриигровую валюту.
- **Система достижений:** игра выдает игроку достижение, при выполнении игроком особых условий, возможно скрытых от игрока
- **Механика тайловой карты:** карта реализуется в виде 2D-сетки тайлов с координатами (x, y), где каждый тайл содержит информацию о типе тайла: вода, суша и т.п. А также данные об интерактивных объектах, таких как сундуки или декорации. Тайлы используются для определения взаимодействий игрока с окружающей средой и обеспечивают базовую структуру для размещения объектов и управления игровым процессом.
- **Механика взаимодействия:** взаимодействие с объектами происходит через проверку соседних тайлов в направлении взгляда игрока. Радиус взаимодействия составляет один тайл от текущей позиции.
- **Система лора мира:** в сундуках будут лежать записки, текст которых генерируется при старте новой игры с применением LLM Qwen 2.5 через бесплатный API.

### 2.2.2 Нефункциональные требования

В приложении должны быть реализованы:

- **Масштабируемость:** начальная архитектура должна закладывать простоту масштабируемости. Добавление новых видов рыб и локаций

должно быть возможно с незначительными изменениями кодовой базы.

- **Производительность:** игра должна обеспечивать работоспособность и минимальную частоту кадров в 30 FPS на устройствах с ОС Android версии 10+.
- **Безопасность:** реализация системы аутентификации с использованием токенов.

## 2.3 Потенциальные риски

В процессе разработки возможны следующие риски:

- Изменение и добавление эндпоинтов API.
- Изменение и добавление таблиц БД.
- Замена спрайтов, текстур на другие.
- Не пройти контроль качества в RuStore.
- По мере разработки архитектура игры, наименования классов и способы их взаимодействия могут изменяться.

## 2.4 Пользовательские сценарии

В игре рассматриваются следующие пользовательские сценарии.

### 2.4.1 Начало новой игры

- **Как:** новый игрок
- **Что хочу:** начать играть
- **Почему:** чтобы погрузиться в мир рыбалки и начать собирать коллекцию рыб
- **Acceptance Criteria:**
  - а) Игрок попадает в главное меню, где есть кнопка «Новая игра»

#### 2.4.2 Ловля рыбы

- **Как:** игрок
- **Что хочу:** ловить рыбу с использованием удочки
- **Почему:** чтобы пополнить свою коллекцию и заработать деньги на улучшения
- **Acceptance Criteria:**
  - а) Игрок может выбрать любую точку водоёма для рыбалки
  - б) Процесс ловли включает мини-игру с шкалой прогресса, последовательностью стрелочек и таймером
  - в) Игрок получает рыбу в инвентарь после успешной ловли

#### 2.4.3 Улучшение удочки

- **Как:** игрок
- **Что хочу:** найти улучшение для удочки

— **Почему:** чтобы облегчить процесс ловли рыб

— **Acceptance Criteria:**

а) Игрок может исследовать локацию и находить улучшения в сундуках

б) Игрок может купить некоторые улучшения

#### **2.4.4 Поимка легендарной рыбы**

— **Как:** игрок

— **Что хочу:** поймать легендарную рыбу

— **Почему:** чтобы получить доступ к новым местам локации

— **Acceptance Criteria:**

а) В особых местах с использованием особой наживки может клюнуть легендарная рыба

б) Мини-игра поимки легендарной рыбы может содержать особые настройки

в) Параметры мини-игры рыбалки выбираются самыми сложными, в сравнении с не легендарными рыбами

#### **2.4.5 Исследование мира**

— **Как:** игрок

— **Что хочу:** исследовать локации

— **Почему:** чтобы найти скрытые сундуки с улучшениями и узнать больше о лоре мира, отыскать новые места для рыбалки



— **Acceptance Criteria:**

- а) Игрок может перемещаться между локациями.
- б) На карте есть скрытые точки интереса
- в) Наличие мини-головоломок в точках интереса

## 2.4.6 Продажа рыбы

— **Как:** игрок

— **Что хочу:** продать пойманную рыбу

— **Почему:** чтобы получить деньги для покупки наживки или улучшений

— **Acceptance Criteria:**

- а) Каждый вид рыбы имеет свою цену
- б) После продажи рыба пропадает из инвентаря
- в) Продавать можно только в магазине

## 2.4.7 Выбор наживки

— **Как:** игрок

— **Что хочу:** выбрать подходящую наживку

— **Почему:** чтобы увеличить шанс поймать рыбу

— **Acceptance Criteria:**

- а) Игрок может купить наживку в магазине

- б) Каждая наживка влияет на вероятность ловли определённых видов рыб
- в) Игрок может выбрать наживку перед началом рыбалки

## **2.5 Перечень основных функциональных блоков системы**

### **2.5.1 Система управления игроком**

Система отвечает за:

- Перемещение по локации независимо от сетки мира
- Взаимодействие с интерактивными объектами (водоем, магазин, сундуки)
- Перемещение между локациями

### **2.5.2 Система рыбалки**

Система отвечает за:

- Мини-игра при взаимодействии с водоемом
- Генерации рыбы и ее параметров для мини-игры

### **2.5.3 Система улучшений**

Система отвечает за:

- Применение найденных игроком улучшений для удочки

#### **2.5.4 Система магазина**

Система отвечает за:

- Покупка наживки
- Покупка улучшений
- Продажа рыбы
- Выдача заданий игроку

#### **2.5.5 Система поимки легендарной рыбы**

Система отвечает за:

- Механика поимки с повышенной сложностью
- Открытие новых областей на карте

#### **2.5.6 Система сохранения**

Система отвечает за:

- Сохранение прогресса при выходе из игры
- Сохранение прогресса при определенных игровых событиях

### **2.5.7 Система коллекционирования**

Система отвечает за:

- Отображение в журнале собранной коллекции рыб

## **3 Начальная архитектура**

### **3.1 UML диаграммы**

На UML диаграммах (См. приложение А) представлены и изображены процессы аутентификации пользователя, игровые механики, запуск новой игры и сохранение прогресса.

### **3.2 ER диаграмма**

На ER-диаграмме (См. приложение Б) показаны таблицы базы данных.

— Таблица Players хранит данные об аккаунте игрока:

- а) Логин
- б) Хэш пароля
- в) Токен авторизации
- г) Дата регистрации
- д) Дата рождения пользователя

— Таблица PlayerLogs хранит данные о периодах активности игрока:

- е) Игрок, к которому относится запись
- ж) Дата и время входа в игру
- з) Дата и время выхода из игры

— Таблица Logs хранит общие логи. Будет использоваться для хранения ошибок при вылетах игры:

- и) Хранит текст лога

к) Дату и время создания записи

### 3.3 Схема API

#### 3.3.1 Регистрация пользователя

Создание нового аккаунта пользователя с указанием логина, пароля и даты рождения:

- Эндпоинт: POST /api/v1/auth/register
- Заголовки: Content-Type: application/json
- Тело запроса: {"**login**": <string>, "**password**": <string>, "**birth\_date**": <string>}
- Ответы:
  - а) Если успешно: HTTP Status 201. JSON ответ: {"**auth\_token**": <string>}, если регистрация прошла успешно
  - б) Если с ошибками: HTTP Status 400. JSON ответ: {"**error**": <message>}, если не получилось зарегистрировать

#### 3.3.2 Авторизация пользователя

Для авторизации необходимо передать логин и пароль:

- Эндпоинт: POST /api/v1/auth/login
- Заголовки: Content-Type: application/json
- Тело запроса: {"**login**": <string>, "**password**": <string>}
- Ответы:

- а) Если успешно: HTTP Status 200.  
JSON ответ: {"auth\_token": <string>}, если авторизация прошла успешно
- б) Если с ошибками: HTTP Status 403. JSON ответ: {"error": <string>}, если не получилось авторизироваться

### 3.3.3 Обновление токена

Получение нового токена авторизации, необходимо передать действующий токен авторизации и логин пользователя.

- Эндпоинт: POST /api/v1/auth/refresh
- Заголовки: Content-Type: application/json
- Тело запроса: {"auth\_token": <string>, "login": <string>}
- Ответы:

- в) Если успешно: HTTP Status 200.  
JSON ответ: {"auth\_token": <string>}, если успешный
- г) Если с ошибками: HTTP Status 403.  
JSON ответ: {"error": <string>}, если токен или логин не действителен

### 3.3.4 Отправка логов

Для отправки записи лога на сервер

- Эндпоинт: POST /api/v1/logs

- Заголовки: Content-Type: application/json
- Тело запроса: {"auth\_token": <string>, "login": <string>, "log\_text": <string>}
- Ответы:

а) Если успешно: HTTP Status 200.

б) Если с ошибками авторизации: HTTP Status 403.  
JSON ответ: {"error": <string>}, если токен или логин не действителен.

### 3.3.5 Отправка метрик игрока

Для отправки времени входа в игру и выхода из игры в БД на сервере.

- Эндпоинт: POST /api/v1/player\_logs
- Заголовки: Content-Type: application/json
- Тело запроса: {"auth\_token": <string>, "login": <string>, "entered\_at": <string>, "exit\_at": <string>}
- Ответы:

а) Если успешно: HTTP Status 200.

б) Если с ошибками авторизации: HTTP Status 403.  
JSON ответ: {"error": <string>}, если токен или логин не действителен.

в) Если с ошибками даты и времени: HTTP Status 400.  
JSON ответ: {"error": <string>}, если даты и время переданы не корректно.



### 3.3.6 Генерация записок

Генерирует текст записки по переданному промту.

— Эндпоинт: POST /api/v1/generate\_note

— Заголовки: Content-Type: application/json

— Тело запроса: {“auth\_token”: <string>, “login”: <string>, “prompt\_text”: <string>}

— Ответы:

а) Если успешно: HTTP Status 200.

JSON ответ: {“note\_text”: <string>}

б) Если с ошибками авторизации: HTTP Status 403.

JSON ответ: {“error”: <string>}, если токен или логин не действителен

## 3.4 Структура игры

### 3.4.1 Основные компоненты

Архитектура приложения (См. приложение В) построена по паттерну RCV (Raw View (Raw Scene), Components, View (Scene)). Явного разделения на слои нет. Основные компоненты:

— Raw View – сцена, на которой расставляются объекты.

— Components – то, что вешается на объекты. Компоненты могут получать ссылки на объекты и менять их.

— View – сцена, в которой на объекты навешаны компоненты.

### **3.4.2 Взаимодействие компонентов**

Способы взаимодействия компонентов:

- На Raw View расставляются объекты, на которые потом вешаются компоненты, и сцена переходит в состояние View.
- Начинается цикл, в котором сцена передает данные компонентам, а они изменяют сцену.

## **3.5 Стек технологий**

### **3.5.1 Backend**

Серверная часть состоит из:

- Язык: Python (FastAPI)
- База данных: MySQL.
- Аутентификация: по логину и паролю или токену
- LLM-интеграция: Микросервис на Python (использует Qwen через API)

### **3.5.2 Игровой движок:**

Игра будет написана на основе:

— Движок: Unity (C#).

### **3.5.3 Хранение данных:**

Хранение данных приложения:

— Локальные: память устройства

— Серверные: база данных на MySQL

## 4 Дизайн макеты

В спрайтах (См. приложение Г) показаны главный герой, пример здания, текстуры тайлов, деревья и главное меню.

### 4.1 UI Kit

#### 4.1.1 Цветовая палитра:

— Основные цвета: Синий(#6d9df3), зеленый(#4c9718), коричневый(#8f563b), оранжевый(#e85e00), темно-серый(#4b4b4b), белый(#fffbfb).

— Дополнительные цвета: Фиолетовый(#76428a), темно-синий(#1d38dc), ярко-желтый(#fbf236).

#### 4.1.2 Цветовые акценты:

В приложении будут применяться следующие цветовые акценты:

— Использование градиентов

— Контрастные цвета.

#### 4.1.3 Шрифты:

В игре будут использоваться Pixelated (пиксельные) шрифты

#### **4.1.4 Компоненты:**

— Кнопки:

- а) Основные кнопки: прямоугольная форма с закругленными углами, фон — градиент между красным и розовым, текст белый.
- б) Второстепенные кнопки: светлые тона (бежевый или светло-зеленый), текст черный.
- в) Активные кнопки: добавление эффекта свечения (светло-бежевый, красный).

— Меню:

- а) Главное меню: фоновое изображение озера, кнопки расположены вертикально по центру.
- б) Настройки: слайдер для регулировки громкости, кнопка включения и выключения звука.

## **4.2 Брендбук**

### **4.2.1 Логотип:**

Описание логотипа:

— Изображение кота в пиксель-арт стиле, держащего рыбу в лапах.

— Кот выполнен в розовых, рыба — в синих.

### **4.2.2 Фирменный стиль:**

Фирменный стиль будет строится на:

— Общий стиль:

- а) Фэнтези-ретро в пиксельном стиле с акцентом на природу и рыбалку.
- б) Элементы дизайна вдохновлены старыми 8-битными играми, но с современной адаптацией для удобства.

— Иконографика:

- а) Все иконки выполнены в пиксель-арт стиле: удочка, рыбы, монеты, наживка, записки.
- б) Иконки имеют четкие границы и минимальное количество деталей для сохранения читаемости.

#### **4.2.3 Анимации:**

В игре будут реализованы следующие анимации:

— Нажатия для кнопок.

— Передвижения персонажа.

— Рыбалки

#### **4.2.4 Звуки:**

В приложении будут реализованы

- 8-ми битные мелодии для фоновой музыки.
- Звуковые эффекты: плеск воды, кот мяукает при взаимодействии, звуки открывания дверей.

#### **4.2.5 Дополнительные элементы:**

Для создания фирменного стиля будут использованы пиксельные текстуры для создания объема и наложения теней.

#### **4.2.6 Элементы декора:**

Элементы декора в приложении:

- Пиксельные деревья, горы, камни, кусты, лед, трава, дома, предметы интерьера для создания атмосферы.
- Декоративные рамки для окон и меню, выполненные в стиле деревянных конструкций.

## **5 Организация проекта**

### **5.1 Настройка гита**

#### **5.1.1 Гит репозиторий**

Ссылка: [https://github.com/Aleygv/Fishing\\_game](https://github.com/Aleygv/Fishing_game)



## ПРИЛОЖЕНИЕ А

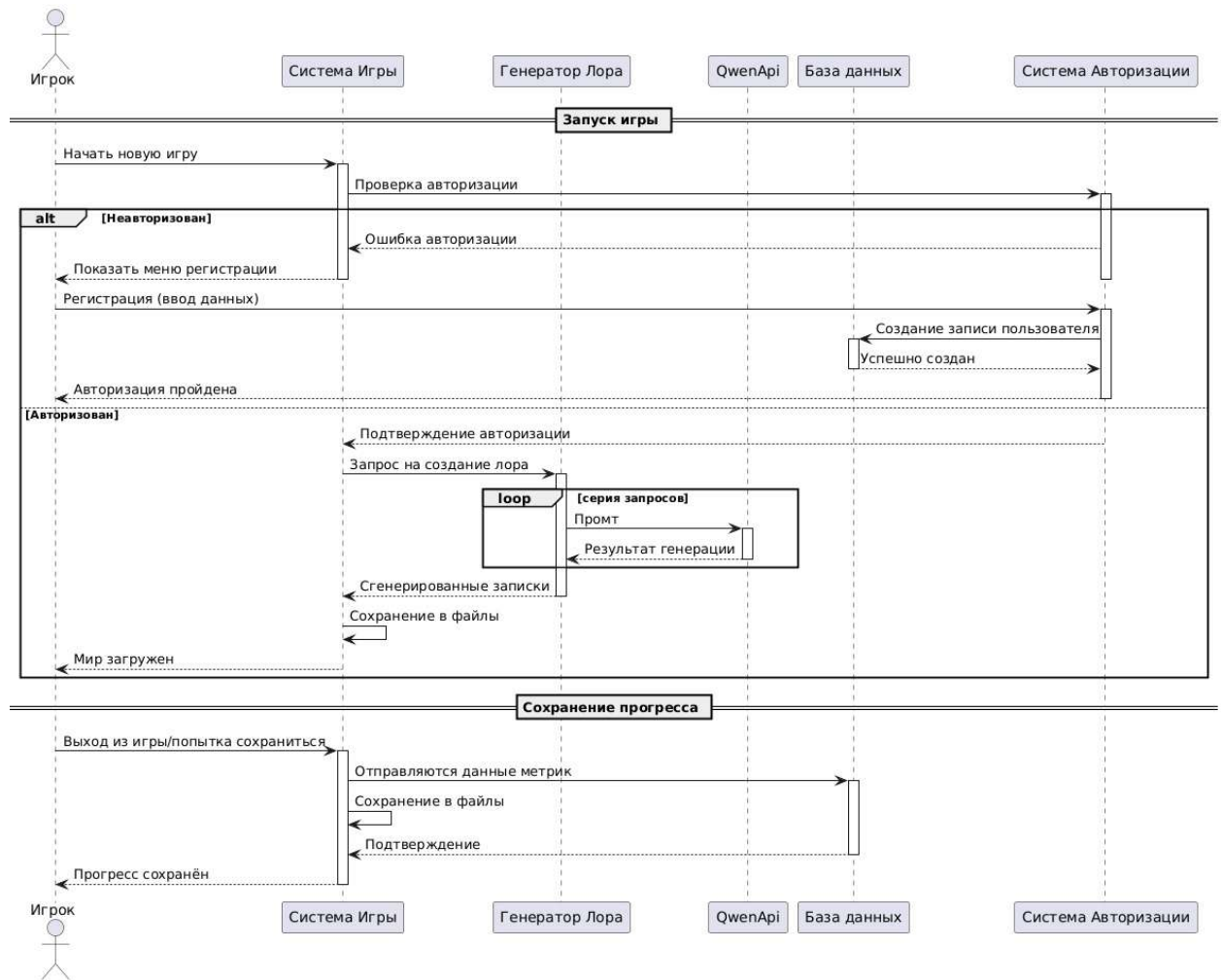


Рисунок 2 — Диаграмма начала игры и сохранения прогресса

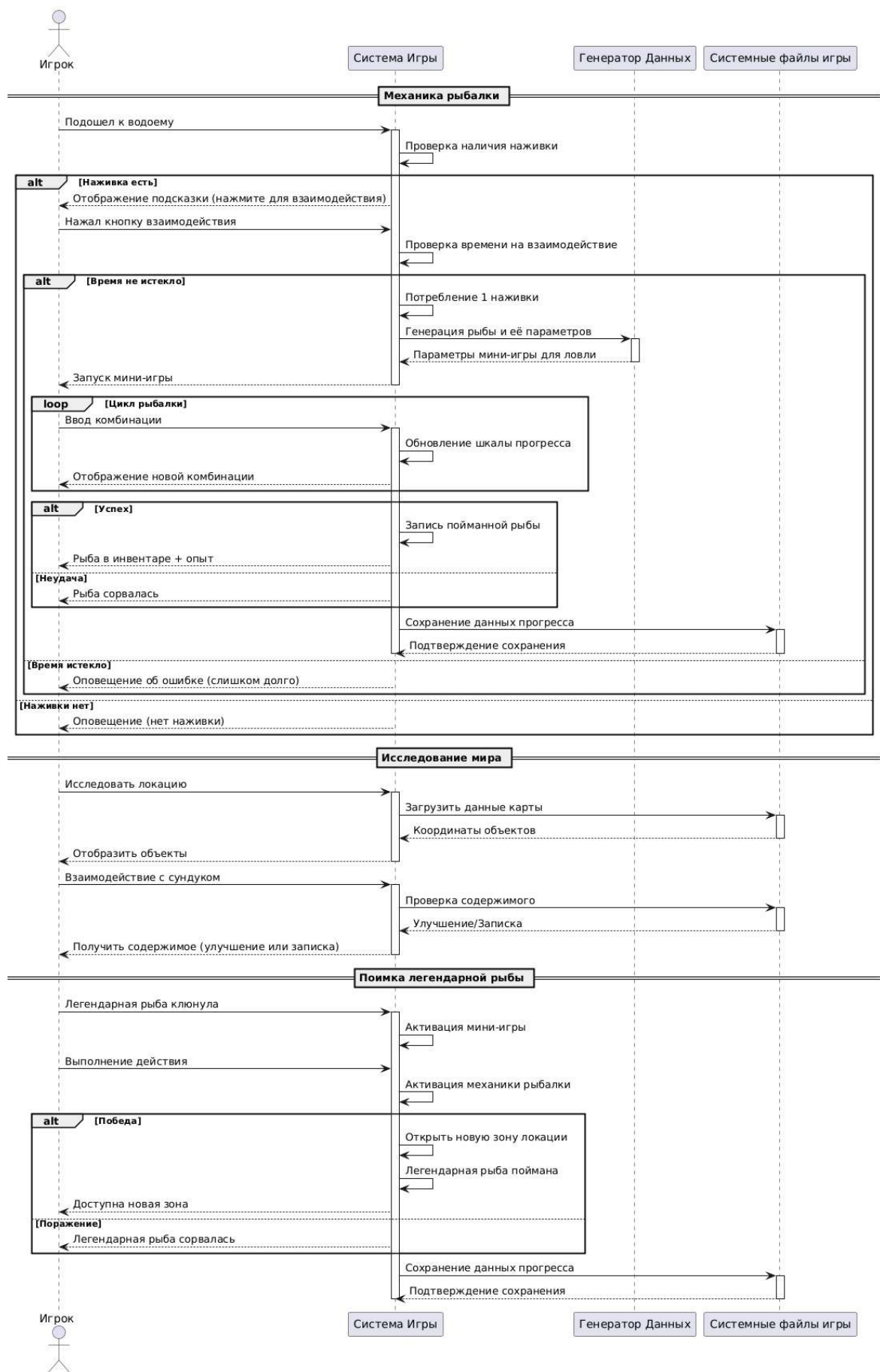


Рисунок 3 — Диаграмма рыбалки, исследования мира



## ПРИЛОЖЕНИЕ Б

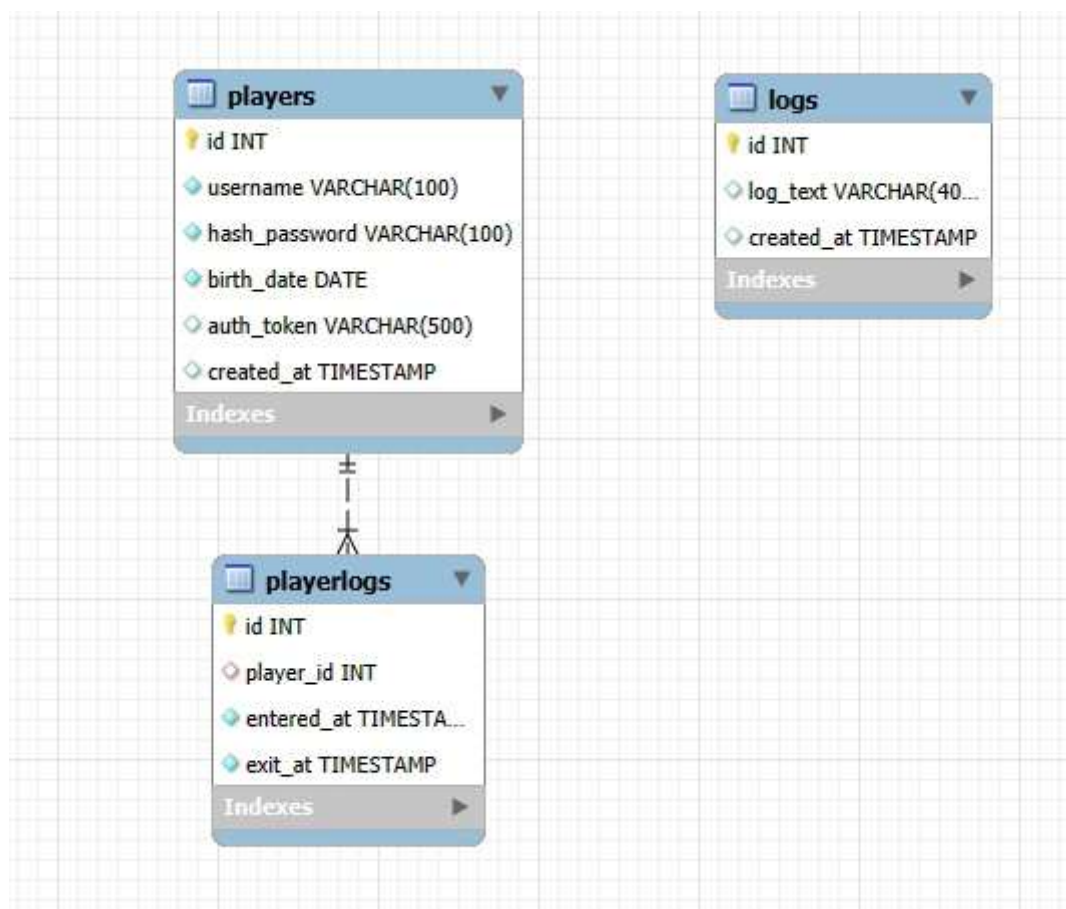


Рисунок 5 — ER диаграмма

## ПРИЛОЖЕНИЕ В

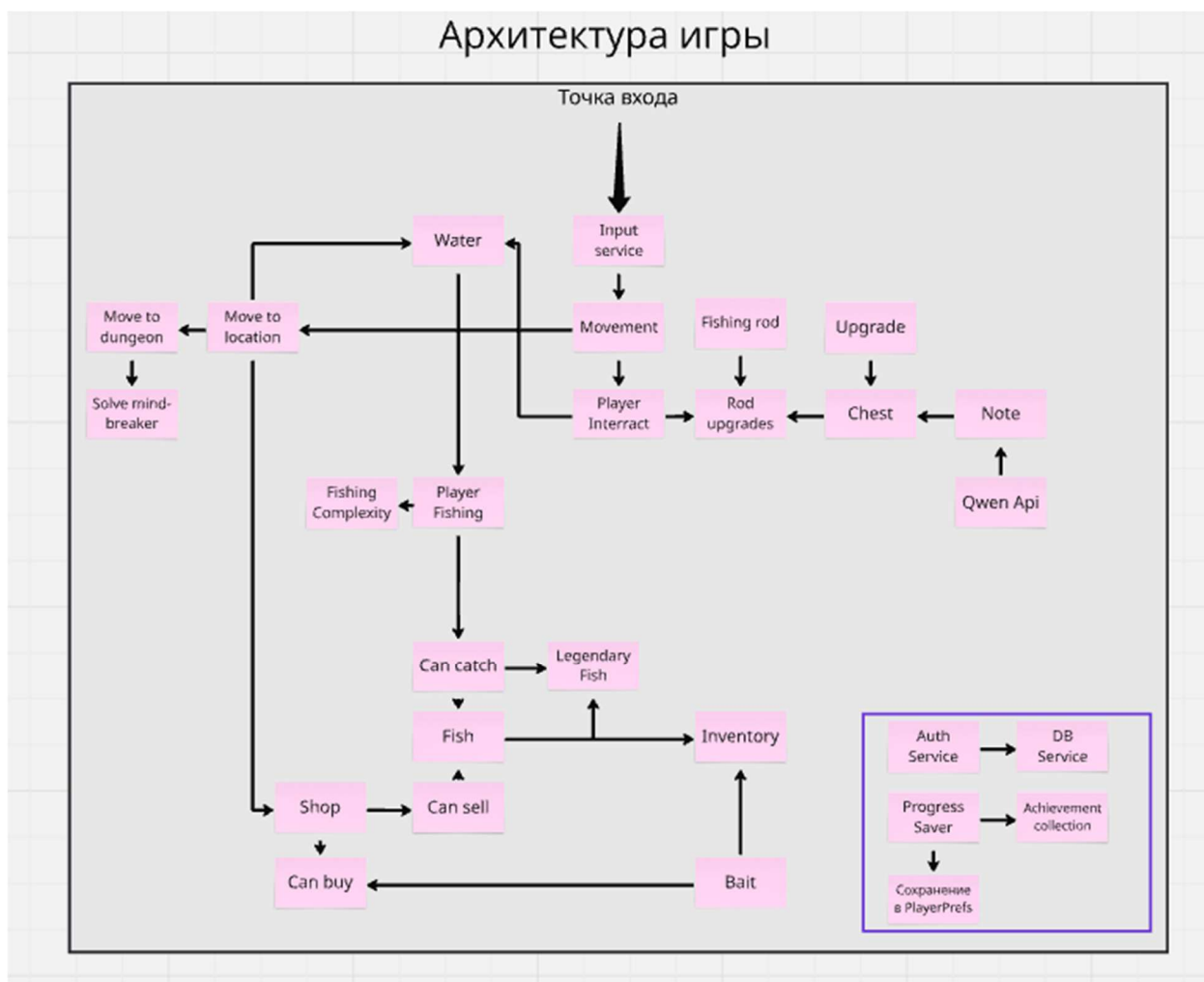


Рисунок 6 — Архитектура игры

## ПРИЛОЖЕНИЕ Г



Рисунок 7 — Объекты мира

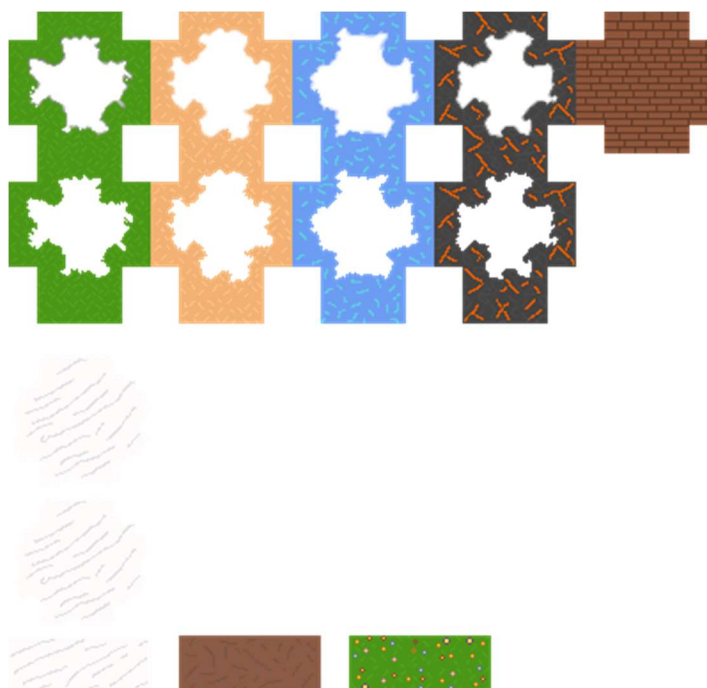


Рисунок 8 — Текстуры поверхностей



Рисунок 9 — Персонаж

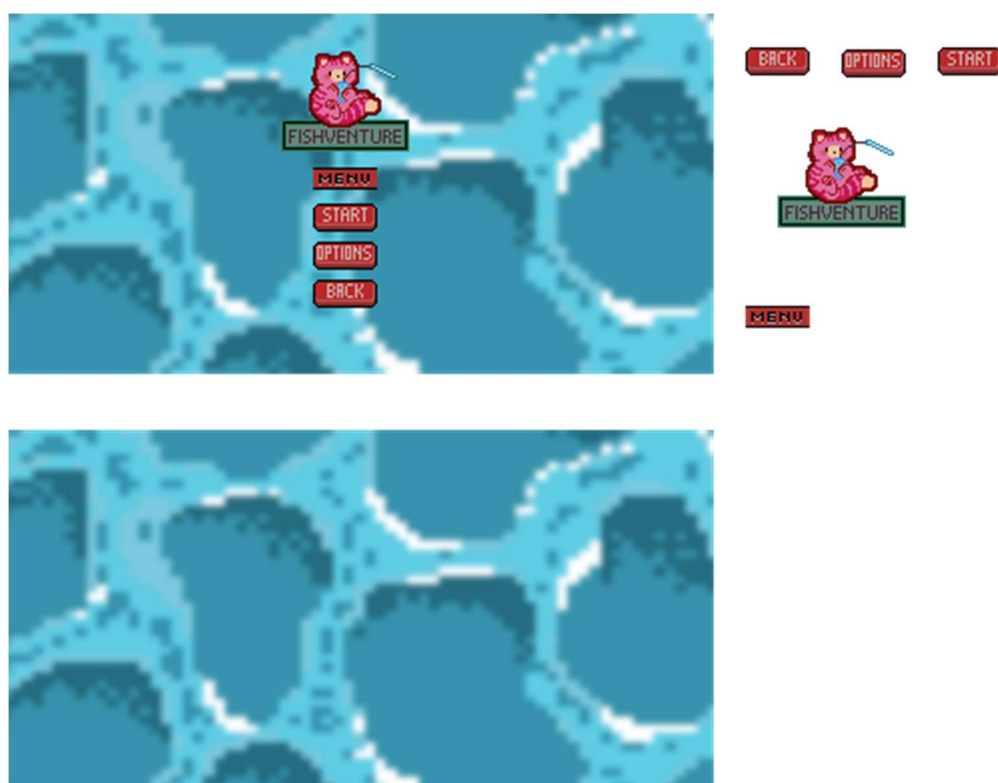


Рисунок 10 — Меню игры

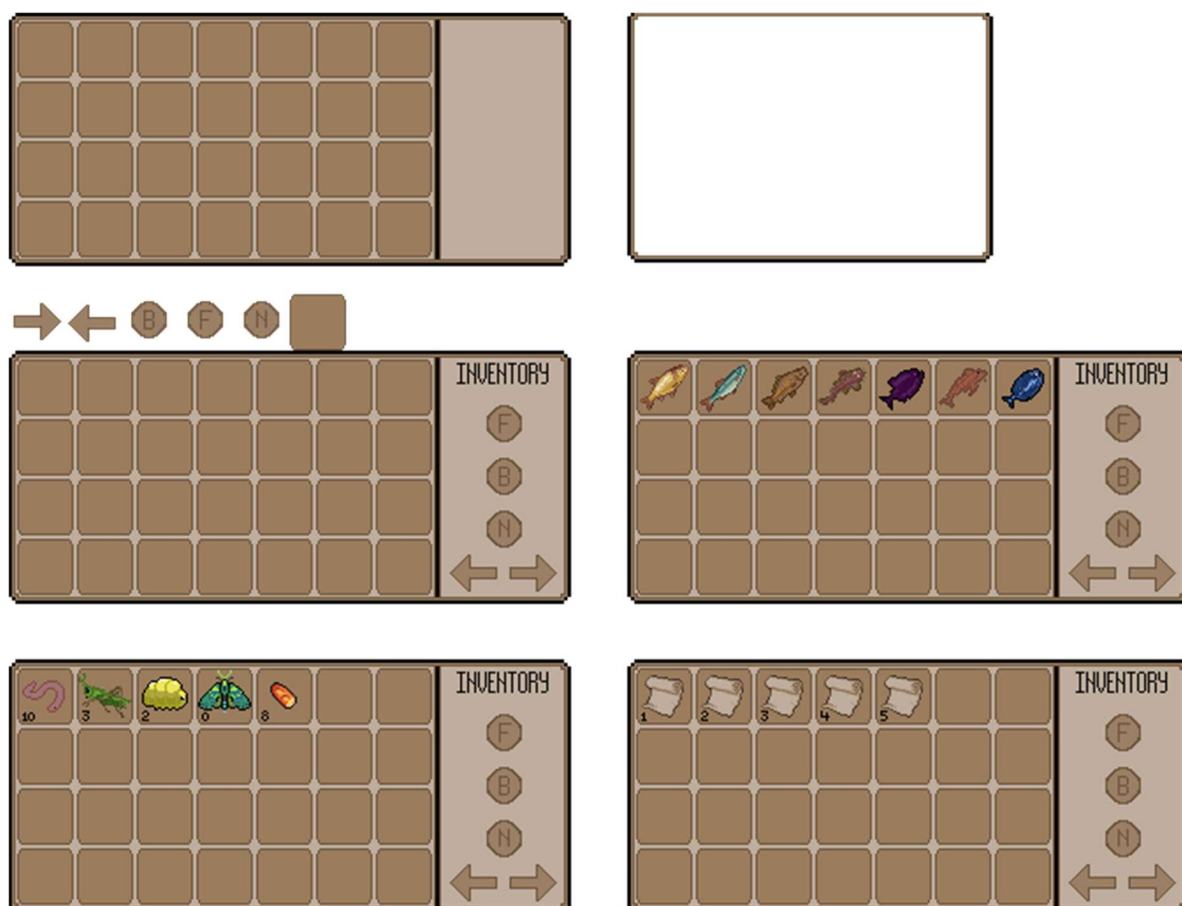


Рисунок 11 — Меню инвентаря



Рисунок 12 — Меню магазина



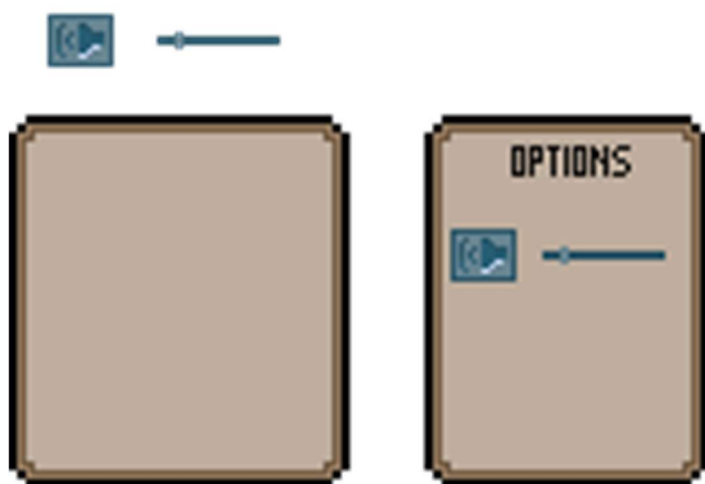


Рисунок 13 — Окно настроек