УДК 004.89

РАЗРАБОТКА АДАПТИВНОЙ СИСТЕМЫ ГЕНЕРАЦИИ ИГРОВЫХ КВЕСТОВ И ДИАЛОГОВ НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

В. Т. Трофимчук[1] [0009-0001-9106-9614], **В. В. Кугуракова**[2] [0000-0002-1552-4910]

[1, 2] Казанский федеральный университет, г. Казань, Россия

[1] vselord.beta@gmail.com, [2] vlada.kugurakova@gmail.com

Аннотация

Рассмотрена проблема создания динамических нарративных систем для видеоигр с интерактивностью в реальном времени. Представлены разработка и тестирование компонента интеграции GPT для генерации диалогов, выявившие критическое ограничение облачных решений – задержку в 30 с., неприемлемую для игрового процесса. Предложена гибридная архитектура адаптивной системы, сочетающая LLM с механизмами обучения с подкреплением. Особое внимание уделяется решению проблем консистентности игрового мира и управлению долгосрочным контекстом взаимодействий с NPC через RAG-подход. Обоснован переход к парадигме Edge AI с применением методов квантования для достижения целевой задержки 200–500 мс. Разработаны метрики оценки персонализации и динамической адаптации контента.

Ключевые слова: *видеоигры, большие языковые модели, генерация диалогов, генерация квестов, адаптивные квесты, процедурная генерация контента, агентное поведение, игровой искусственный интеллект, машинное обучение в играх.*

ВВЕДЕНИЕ

Современная игровая индустрия переживает значительную трансформацию, связанную с интеграцией технологий искусственного интеллекта (ИИ). Согласно данным аналитического ресурса Statista[1], в начале 2025 г. более трети разработчиков игр по всему миру заявили, что их студии уже используют ИИ в различных аспектах разработки. Это свидетельствует о стремительном проникновении нейросетей в геймдев, где они применяются для генерации текстур, написания диалогов, создания анимации и даже замены актеров озвучки.

Согласно обзору [1], в 2023–2025 г. наблюдается экспоненциальный рост исследований по интеграции больших языковых моделей (LLM)[2] в игровую индустрию, при этом 67% работ посвящены решению проблем адаптивности и персонализации игрового контента (см. табл. 1).

Табл. 1. Статистика применения LLM в играх

Направление применения	Доля исследований, %	Основные решаемые проблемы
Генерация диалогов NPC	42	Консистентность, персонализация
Динамические квесты	28	Баланс сложности, связность сюжета
Поведение NPC	18	Предсказуемость, адаптивность
Процедурный контент	12	Качество, разнообразие

[1] Statista – немецкая компания, специализирующаяся на рыночных и потребительских данных. URL: https://www.statista.com/

[2] LLM (Large Language Model, «большая языковая модель») — тип программы искусственного интеллекта, которая обучена работать с текстом: понимать, генерировать, пересказывать, дополнять, переводить и анализировать его.

Ключевой проблемой современных игр остается создание динамичных и реалистичных игровых миров, где неигровые персонажи (NPC[3]) демонстрируют адаптивное поведение и способны к осмысленному взаимодействию с игроком. Исследование [2] показало, что 99% геймеров считают, что продвинутый ИИ улучшит геймплей, при этом основные проблемы современных игр включают повторяющиеся диалоги (60%) и неспособность NPC адаптироваться к изменениям в игре.

Внедрение ИИ в игровую индустрию сопровождается рядом этических и юридических вызовов. Согласно отчету GDC[4] 2024, 84% разработчиков выражают обеспокоенность этикой ИИ, включая копирование стилей без согласия авторов и снижение ценности человеческого труда.

Проблема авторских прав также остается острой. В 2023 г. компания Valve[5] отклоняла публикацию игр в Steam[6] из-за опасений относительно юридического статуса художественных элементов, созданных ИИ. В 2024 г. эта компания сформулировала политику относительно таких игр, предусматривающую проверку контента на предмет нарушений авторских прав.

Профессиональные союзы также активно реагируют на распространение ИИ. Американская гильдия артистов телевидения и радио (SAG-AFTRA[7])

[3] Non-Playable Character (NPC) — неигровой персонаж (неиграбельный персонаж) в играх, который не находится под контролем игрока.

[4] GDC (Game Developers Conference) — ежегодная конференция для профессиональных разработчиков компьютерных игр.

[5] Valve (Valve Corporation) (также известна как Valve Software) — американская частная компания, занимающаяся разработкой, изданием и цифровой дистрибуцией компьютерных игр.

[6] Steam — онлайн-сервис цифрового распространения компьютерных игр и программ, разработанный и поддерживаемый компанией Valve.

[7] SAG-AFTRA (сокр. от Screen Actors Guild-American Federation of Television and Radio Artists) – американский профсоюз, представляющий актеров, журналистов, радиоведущих, записывающих артистов, дублеров, публичных ораторов и других специалистов в области медиа и развлечений.

подала жалобу на создателей Fortnite[8] за использование ИИ для озвучки персонажа без надлежащего уведомления. Данные примеры демонстрируют необходимость разработки четких правовых рамок для использования ИИ в игровой индустрии.

В настоящей статье представлены результаты разработки компонента интеграции технологии GPT[9] для генерации диалогов и предложена архитектура расширенной адаптивной системы, сочетающей большие языковые модели с принципами обучения с подкреплением для создания динамичных игровых квестов и диалогов.

ЭВОЛЮЦИЯ ИИ В ИГРАХ: ОТ СКРИПТОВ К НЕЙРОСЕТЯМ

Исторически ИИ в играх ограничивался примитивными алгоритмами поведения «врагов», которые следовали заранее определенным скриптам и паттернам. В 1990-е годы произошел значительный прорыв — игры серии *Baldur's Gate* представили NPC, способных запоминать действия игрока и соответствующим образом реагировать на них в будущем. Это стало важным этапом в развитии неигровых персонажей, которые превратились из статичных элементов декораций в активных участников игрового мира.

Текстовые игры стали важным полигоном для испытания нарративных технологий. Еще в 1970-е годы такие игры, как *Colossal Cave Adventure* и *Zork*, заложили основы интерактивного повествования. Особенностью текстовых форматов была их глубокая нелинейность: проще было реализовать сложную ветвистую структуру с сотнями развилок и множеством финалов, что до сих пор представляет сложность для современных AAA[10]-проектов.

[8] Все компьютерные игры, упомянутые в настоящем исследовании, описаны в разд. Лудография.

[9] GPT (Generative Pre-trained Transformer) — семейство больших языковых моделей (LLM), разработанных OpenAI.

[10] AAA (triple-A) — игры высшего уровня производства с большим бюджетом разработки и маркетинга, создаваемые крупными студиями и издателями (например, The Last of Us, Red Dead Redemption, Cyberpunk 2077).

Современные нейронные сети совершили качественный скачок — они не просто следуют инструкциям, а создают контент: от диалогов и квестов до текстур и озвучки.

Компании активно экспериментируют с различными подходами к интеграции ИИ, например:

- Ubisoft[11] использует Ghostwriter[12] для генерации реплик второстепенных персонажей и разрабатывает проект NEO NPC[13], позволяющий игрокам разговаривать с персонажами голосом;
- Epic Games[14] интегрировала ИИ для озвучки Дарта Вейдера в *Fortnite*, использовав нейросеть, обученную на голосе покойного Джеймса Эрла Джонса с согласия его семьи;
- разрабатываются специализированные инструменты для «оживления» виртуальных персонажей, такие как Inworld[15] и Bitpart[16], которые привлекают ученых и ветеранов игровой индустрии для решения задачи повышения последовательности и контекстной релевантности нейросетей.

Табл. 2 наглядно иллюстрирует трансформацию нарративных и контент-генерационных систем в индустрии, показывая, как ИИ развивался в играх, переходя от примитивных структур к адаптивным современным нейросетям.

[11] Ubisoft (ранее Ubi Soft Entertainment SA) — французская компания, специализирующаяся на разработке и издании компьютерных игр.

[12] Ubisoft Ghostwriter — инструмент на базе искусственного интеллекта, разработанный компанией Ubisoft для помощи сценаристам.

[13] Ubisoft NEO NPC — инструмент для разработчиков, который позволяет создавать более правдоподобное общение с неигровыми персонажами (NPC).

[14] Epic Games (ранее Epic MegaGames и Potomac Computer Systems) — американская компания, занимающаяся разработкой компьютерных игр и программного обеспечения.

[15] Inworld — движок искусственного интеллекта для создания динамичных неигровых персонажей (NPC) и игровых миров.

[16] Bitpart AI — интеллектуальная платформа для автоматизации создания контента. Предоставляет инструменты для легкой генерации текста и мультимедиа.

Табл. 2. Эволюция подходов к созданию игрового контента

Период	Технологии	Характер контента	Примеры
1970–1980-е	Текстовые парсеры	Статические деревья диалогов	*Colossal Cave Adventure, Zork*
1990–2000-е	Скриптовые системы	Ветвящиеся сценарии с ограниченной вариативностью	*Baldur's Gate, Fallout*
2010–2020-е	Процедурная генерация (CG)	Динамический контент по шаблонам	*Minecraft, The Elder Scrolls V: Skyrim*
2020-е и по настоящее время	Большие языковые модели (LLM)	Адаптивный контент, генерируемый в реальном времени	NEO NPC, *AI Dungeon*

СВЯЗАННЫЕ РАБОТЫ

Бурное развитие LLM в последние годы привело к активным исследованиям в области адаптивной генерации квестов и диалогов для цифровых игр. Актуальные публикации демонстрируют универсальность LLM при создании интерактивных сценариев и коммуникации персонажей, акцентируя внимание как на архитектурных новшествах, так и на оценке пользовательского опыта.

Обзорные статьи (например, [1]) освещают интеграцию LLM в генерацию сюжетов, диалогов NPC и адаптивных квестов, выделяя современные тренды и проблемы, связанные с созданием связного и увлекательного опыта для игрока. В работе [3] подчеркнуто, что новые модели генерируют более естественные и последовательные тексты, однако остаются открытыми вопросы по контролю качества и консистенции данных в коммерческих играх.

Внедрение LLM в игровые движки уже реализовано в ряде прикладных проектов. В статье [4] представлена архитектура генерации квестов и диалогов

на базе GPT-4o[17] и Claude 3.7[18], позволяющая динамично строить ветвящиеся сюжеты и подстраиваться под поведение игрока; также дан сравнительный анализ моделей и пользовательское тестирование. В исследовании [5] описан опыт интеграции LLM (Google Gemini API[19], Sentence-BERT[20]) для создания органичных диалогов и миссий для NPC с высокой степенью адаптивности.

Персонализированные сценарии успешно реализуются благодаря гибридным подходам. Например, в [6] LLM объединены с графами знаний для процедурной генерации индивидуализированных квестов и диалогов в RPG[21], что обеспечивает связность нарратива и высокий уровень кастомизации.

В работах отечественных исследователей также уделяется значительное внимание генеративным системам для игр. Например, в [7] представлен подход к генеративной симуляции игрового окружения в реальном времени, что согласуется с целями данного исследования по достижению минимальной задержки. В [8] представлена методология создания корпуса текстов видеоигр на основе универсальной структуры, что может служить основой для обучения и валидации специализированных языковых моделей для игрового контента.

Отметим, что современные работы также уделяют внимание пользовательской оценке новых технологий. В [9] проанализировано качество диалогов, сгенерированных LLM, на примере реальных коммерческих продуктов,

[17] GPT-4o – флагманская мультимодальная модель от OpenAI, выпущенная в мае 2024 г. Она основана на GPT-4, но дополнена поддержкой обработки сразу нескольких видов данных: текста, изображений, аудио и видео.

[18] Claude 3.7 Sonnet — передовая гибридная модель искусственного интеллекта от компании Anthropic, выпущенная в феврале 2025 г.. Она сочетает два режима работы: быстрые, мгновенные ответы для простых запросов и расширенное пошаговое рассуждение для сложных задач, таких как программирование и аналитика.

[19] Google Gemini API — канал взаимодействия с нейросетью Gemini от Google для разработчиков, занимающихся машинным обучением и созданием сервисов на базе больших языковых моделей.

[20] Sentence-BERT (SBERT) — модификация архитектуры BERT (Bidirectional Encoder Representations from Transformers), разработанная специально для генерации векторов фиксированной размерности, представляющих смысл целых предложений. Эти вектора (вложения предложений, эмбеддинги) позволяют сравнивать семантическую близость между предложениями.

[21] RPG (от англ. Role-Playing Game) — ролевая игра, жанр видеоигр, в которых игрок управляет одним или несколькими персонажами.

в частности *Disco Elysium*, с глубоким разбором отзывов игроков и сравнением с ручным написанием. Аналогично в [10] выполнена структурная оценка квестов, созданных LLM, выявлены типичные ошибки моделей и предпочтения аудитории.

В работе [11] предложена архитектура, сочетающая LLM с графами знаний, что позволяет достичь коэффициента персонализации до 0.87 против 0.62 у чистых LLM-решений [12].

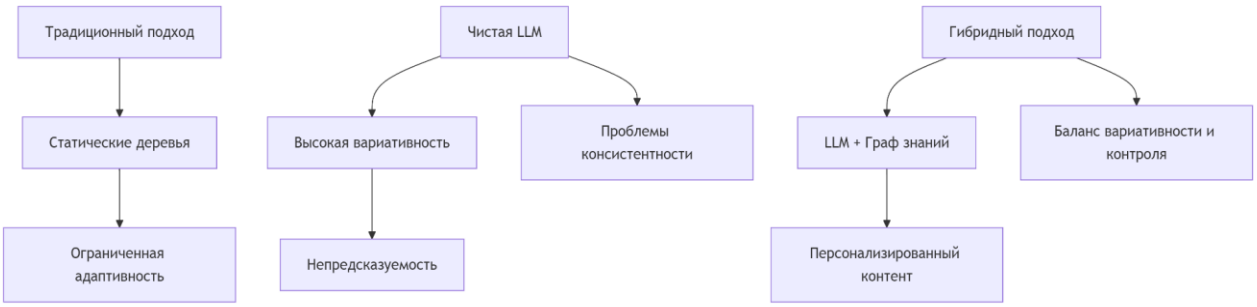

Рис. 1. Сравнение архитектур генерации квестов

На рис. 1 схематично представлено алгоритмическое различие в подходах (традиционный против чистого LLM-решения или гибридного подхода).

Однако необходимо отметить, что гибридный подход требует значительных вычислительных ресурсов для поддержания актуальности графа знаний в реальном времени.

Таким образом, современные исследования характеризуются инновационными архитектурными решениями, междисциплинарными методологиями и акцентом на тщательную практическую оценку LLM-генерации квестов и диалогов как важного инструмента для будущих игровых приложений.

РАЗРАБОТКА КОМПОНЕНТА ИНТЕГРАЦИИ GPT ДЛЯ ГЕНЕРАЦИИ ДИАЛОГОВ

Предлагаемый нами подход отличается от существующих решений тем, что стремится обеспечить реальную интерактивность и автономность в процессе генерации квестов и диалогов. Если большинство современных систем опирается на серверные LLM, требующие постоянного сетевого соединения и централизованной вычислительной инфраструктуры, то предлагаемый метод основан на модели Edge AI[22] — исполнении нейросетевых процессов непосредственно на пользовательском устройстве [13].

Концепция нового подхода

Основная цель такого подхода — это минимизация сетевой задержки и достижение отклика в реальном времени, что критически важно для игровых приложений, где скорость реакции напрямую влияет на восприятие интерактивности.

Итак, перечислим целевые метрики:

- снижение задержки до диапазона 200–500 мс по сравнению с ~1800 мс для облачных решений;
- автономность, которая позволяет работать в офлайн-режиме без зависимости от стабильности интернет-соединения;
- защита персональных данных, так как вычисления происходят локально, без передачи пользовательских данных во внешние сервисы [14].

[22] Edge AI (Artificial Intelligence at the Edge) — периферийный искусственный интеллект. Это парадигма вычислений, при которой задачи машинного обучения и инференса выполняются непосредственно на периферийных устройствах — смартфонах, камерах, датчиках и другом оборудовании. Вместо того чтобы полагаться на облачные вычисления, Edge AI обрабатывает данные локально, не отправляя их в облако.

Для реализации такой архитектуры требуются агрессивная оптимизация и адаптация LLM под устройства с ограниченными ресурсами. Здесь ключевыми направлениями выступают методы TinyML[23] и компрессии моделей (квантование[24], прунинг[25], дистилляция[26]), позволяющие существенно снизить объем вычислений без потери качества текстовой генерации.

Аналогичные идеи использовались в промышленных сценариях для автономных датчиков и систем компьютерного зрения, что подтверждает эффективность принципов Edge AI в условиях ограниченного аппаратного обеспечения [14].

Таким образом, в отличие от предыдущих решений, завязанных на серверные ресурсы и облачные API, предложенная концепция рассматривает генерацию квестов и диалогов как распределенный когнитивный процесс, выполняемый локально, что открывает возможности для взаимодействия с игроками, осуществляемого действительно в реальном времени, и использования новых форм автономного геймплейного поведения NPC.

[23] TinyML — раздел машинного обучения, который позволяет запускать модели искусственного интеллекта на миниатюрных устройствах с низким энергопотреблением, таких как микроконтроллеры.

[24] Квантование модели — мощный метод оптимизации модели, который уменьшает объем памяти и вычислительные затраты нейронной сети (NN), преобразуя ее веса и активации из чисел с плавающей запятой высокой точности (например, 32-битное число с плавающей запятой или FP32) в типы данных с более низкой точностью, такие как 8-битные целые числа (INT8).

[25] Прунинг нейронных сетей — метод сжатия (уменьшения расхода памяти и вычислительной сложности) сети за счет устранения части параметров в предобученной модели.

[26] Дистилляция знаний (Knowledge Distillation) — метод оптимизации моделей и сжатия в машинном обучении (ML), при котором компактная «студенческая» модель обучается воспроизводить производительность более крупной и сложной «учительской» модели.

Архитектура решения

В рамках предварительной работы [12] был разработан компонент Integrator для интеграции технологии GPT в видеоигры, представляющий собой библиотеку на C#, упакованную в NuGet-пакет[27].

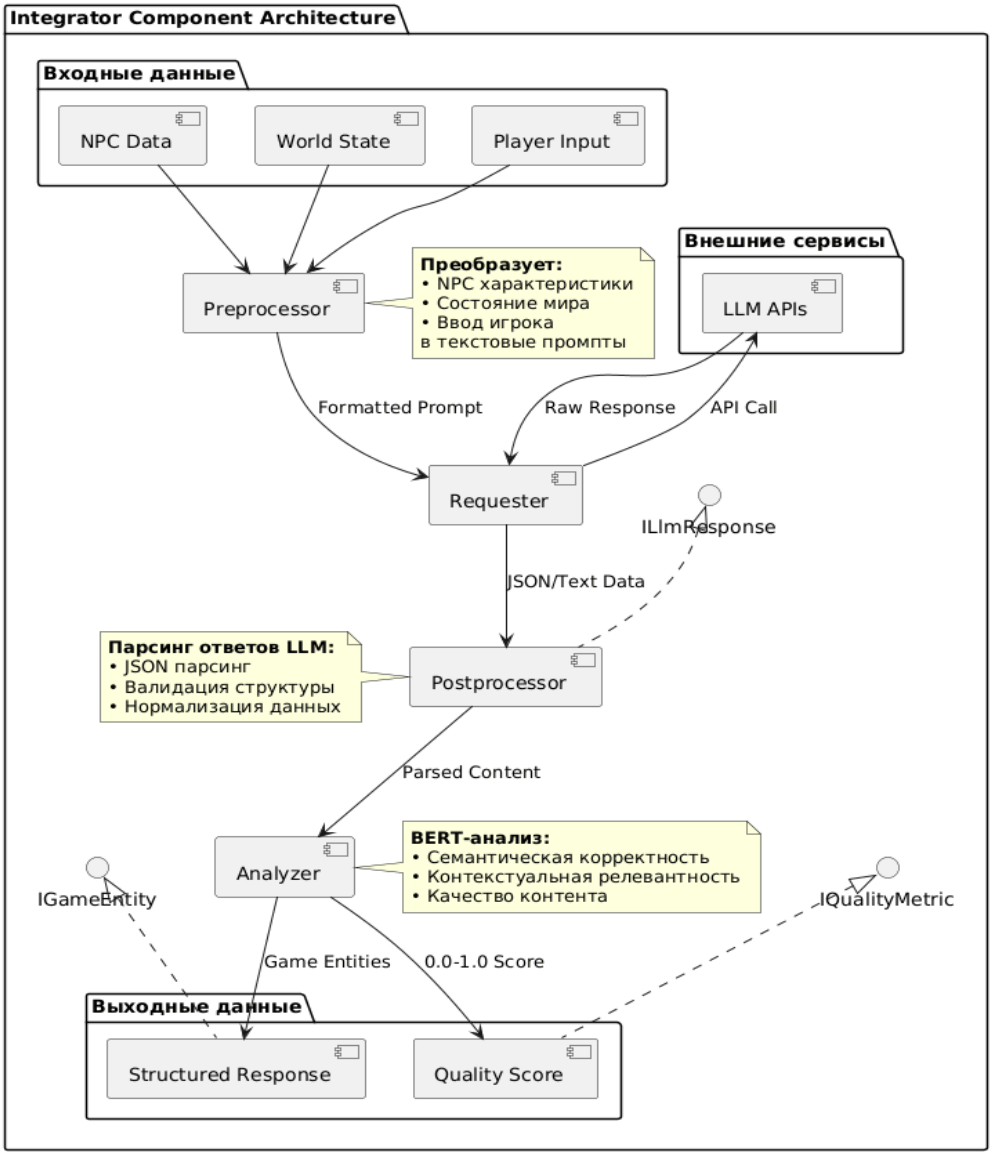

Рис. 2. Архитектура библиотеки

[27] NuGet-пакет – ZIP-файл с расширением .nupkg, содержащий скомпилированный код (DLL), другие связанные файлы и метаданные о пакете, такие как версия и описание. Он служит для управления зависимостями и повторного использования кода в .NET-проектах, позволяя разработчикам обмениваться библиотеками и инструментами через систему NuGet.

Архитектура компонента (см. рис. 2) включает четыре ключевых модуля:

- Preprocessor — преобразует сущности игрового мира (NPC, окружение) в текстовые промпты для LLM;
- Requester — осуществляет взаимодействие с API языковых моделей через RESTful-интерфейс;
- Postprocessor — выполняет парсинг ответов LLM и преобразует их в структурированные игровые сущности;
- Analyzer — нейросетевая модель на основе BERT, оценивающая корректность сгенерированного контента.

Необходимые интерфейсы на рис. 2 помечены как i<class>.

Входные данные NPC Data, World State, Player Input составляют Narrative Environment — параметры окружающей среды, влияющие на формирование запроса, и будут детально описаны ниже.

Выходные данные состоят из:

- Structure Response — структурированный ответ, например ветвящийся диалог, полученный в результате диалогогенерации[28] (рис. 3).
- Quality Score — оценка синтаксической корректности и семантической согласованности.

[28] Диалогогенерация — это автоматическое создание диалогов и разговоров между персонажами с помощью технологий искусственного интеллекта, нейронных сетей и обработки естественного языка.

```
Entry text: Приветствую, странник! Я - сэр Альфред, рыцарь этих земель. Чем могу быть вам полезен?
Entry child count: 2

Dialogue varient: 1
Player name: Игрок
Player text: Здравствуйте, сэр Альфред. Я путешественник, и мне не помешает помощь опытного воина.
NPC name: Альфред
NPC text: Рад, что вы обратились ко мне. Как я могу помочь вам на вашем пути?
Dialogue node childs count: 2

Dialogue varient: 1.1
Player name: Игрок
Player text: Я бы хотел присоединиться к вашему отряду и помочь в борьбе с этими угрозами.
NPC name: Альфред
NPC text: Это благородное предложение, молодой человек. Я буду рад принять вас в наши ряды. Ваша помощь нам очень пригод
ится.
Dialogue node childs count: 0

Dialogue varient: 1.2
Player name: Игрок
Player text: Понимаю. Я постараюсь держаться подальше от этих мест, пока ситуация не станет более безопасной.
NPC name: Альфред
NPC text: Я понимаю ваши опасения. Однако, если вы решите остаться, я гарантирую вашу безопасность под моим присмотром.
Dialogue node childs count: 0
```

Рис. 3. Пример сгенерированного ветвистого диалога, выведенного в лог

Представленная архитектура предполагает минимальную задержку (кроме задержки самого облачного GPT API) и полную автономность за счет внедрения библиотеки прямо в код игрового движка.

Реализация и тестирование

Компонент был успешно интегрирован в демонстрационный проект на Unity (см. рис. 4), где была проиллюстрирована возможность генерации ветвящихся диалогов с учетом характеристик NPC (профессия, личностные черты, социальные связи).

Рис. 4. Демонстрационный проект использования библиотеки

Табл. 3. Синтаксическая корректность (грамматика)

Сценарий	Сгенерированный текст	Оценка (вероятность)	Комментарий
Корректно	*Я видел его вчера, когда он шел по мосту.*	$P(text) \approx 0.9$ (высокая)	Грамматически правильное предложение
Некорректно	*Я видел его вчера он мосту шел по.*	$P(text) \approx 0.07$ (низкая)	Нарушение порядка слов и пунктуации. LLM присваивает низкую вероятность такой последовательности токенов, что позволяет отфильтровать этот результат
Корректно	*Ты возьмешь этот меч?*	$P(text) \approx 0.88$ (высокая)	Корректная форма глагола и вопросительное предложение
Некорректно	*Ты этот меч брать?*	$P(text) \approx 0.15$ (низкая)	Ошибка согласования/ спряжения глагола

Для обеспечения корректности работы была реализована система валидации (детально описана в [12]), включающая:

- проверку синтаксической корректности P сгенерированного текста (см. табл. 3);
- оценку семантической согласованности с контекстом игры (см. табл. 4);
- контроль соответствия заданным параметрам NPC.

Табл. 4. Семантическая согласованность (Контекст игры)

Сценарий	Сгенерированный текст *(ответ Кузнеца)*	Оценка (Вероятность)	Комментарий
Корректно (согласованно)	*Отлично, это **золото** – то, что нужно для клинка! Благодарю тебя, эльф.*	$P(text)$ (золото) ≈ 0.92 (высокая)	Золото то, что нужно кузнецу
Некорректно (несогласованно)	*Ты принес мне **ржавые сапоги**? Это не поможет мне выковать серебряный клинок.*	$P(text)$ (ржавые сапоги) ≈ 0.13 (низкая)	Ржавые сапоги не то, что нужно кузнецу

Результаты этой валидации и являются Quality Score, определяющие качество сгенерированного контента.

Под семантической согласованностью в дальнейшем будем понимать согласованность сгенерированного контента с контекстом игры (*например, для NPC прописана история, что он является обычным деревенским кузнецом, а в контексте истории мира говорится, что мир находится в эпохе раннего средневековья, но не указано, что мир является магическим фэнтези, следовательно, магии, которая может имитировать или обходить современные технологические процессы, нет. Если такому кузнецу принести ржавые предметы и попросить переплавить, то он откажется, так как не сможет обработать данные предметы*).

Для оценки семантической согласованности часто используют внешние механизмы валидации (*например, Knowledge Graph[29] или Правила игры[30]*) в дополнение к внутренней вероятностной оценке LLM. Это гарантирует, что даже синтаксически корректный, но контекстно-неправильный ответ (*как в примере с «ржавыми сапогами»*) будет отброшен.

Проблемы интеграции

При практической интеграции возникли сложности с управлением зависимостями NuGet-пакетов внутри Unity, особенно для библиотек с нативными компонентами (например, ONNX Runtime[31] для оценочной нейросети BERT). Это создало значительный риск нестабильности и усложнило процесс сборки.

Анализ производительности

Тестирование с использованием облачного API GPT-3.5-Turbo[32] выявило критическую проблему с временем отклика (latency).

Для сценария генерации одного полного диалогового хода (фраза NPC и три варианта ответа для Игрока) были получены данные (табл. 5), позволившие выявить зависимости времени генерации.

Общее время задержки T_{total}, исходя из полученных данных, может быть выражено как

$$T_{\text{total}} = T_{\text{network}} + T_{\text{pre}} + T_{\text{API}} + T_{\text{post}} + T \ ,$$

где T_{network} — время выполнения отправки запроса нейросети, T_{pre} — время составления промта, T_{API} — время выполнения запроса нейросетью,

[29] Knowledge Graph (граф знаний) — семантическая сеть, в которой хранится информация о разных сущностях и взаимосвязях между ними.

[30] «Правила игры» (в контексте семантического анализа) — правила и соглашения, которые определяют, какие высказывания считаются правильными или адекватными в контексте Это понятие связано с концепцией «языковых игр» Людвига Витгенштейна, которая объясняет как язык функционирует в рамках социальных контекстов, которые рассматриваются как игры.

[31] ONNX (Open Neural Network Exchange) — открытая библиотека программного обеспечения для построения нейронных сетей глубокого обучения.

[32] GPT-3.5-Turbo — это улучшенная версия языковой модели GPT-3.5, выпущенная 1 марта 2023 г. Модель оптимизирована для разговорного чата и может имитировать человеческие ответы.

T_{post} — время парсинга ответа, пришедшего от нейросети, T — время выполнения синтаксического и семантического анализа.

Табл. 5. Параметры генерации

Параметр генерации	Значение	Описание параметра
Количество генерируемых вариантов (N_{variants})	3	Варианты ответа NPC
Длина генерируемого варианта (L_{variants})	3	Длина ответа NPC (в предложениях)
Среднее время генерации	30 с	Время от отправки запроса до получения итогового ответа

Конфигурация устройства, на котором проводились вычисления:

- CPU Intel Core i5 10300H;
- ОЗУ 32 ГБ памяти;
- GPU GeForce GTX 1650 от Nvidia.

Было установлено, что время выполнения запроса нейросетью T_{API} является доминирующим фактором и прямо пропорционально количеству вариантов и их длине:

$$T_{\text{API}} \sim N_{\text{variants}} \cdot L_{\text{variants}},$$

где N_{variants} – количество генерируемых вариантов, а L_{variants}– длина генерируемого варианта.

Задержка в 30 с является неприемлемой для интерактивного игрового процесса в реальном времени, для которого целевое время отклика должно составлять менее 1 с [15]. Последнее значение выбрано как следствие следующих критериев.

1. Разрушение иммерсивности и «магии» игры. Игры, особенно с глубоким повествованием, стремятся погрузить игрока в свой мир.

- При диалогогенерации, если Игрок задает вопрос NPC, но вместо мгновенного, живого ответа получает 30 с молчания, то пропадает эффект иммерсивности – впрочем, возможны решения, позволяющие скрыть эту паузу, например анимация idle[33]-действий, например, почесывание затылка или болтание руками, как будто NPC раздумывает, но эти «заплатки» будут явно вычисляться опытным Игроком.
- При кестогенерации[34] действие Игрока должно запустить новую цепочку событий, и снова ожидание в 30 с для выбора задания полностью разрушает *нарративный импульс*[35] и *драматический момент*[36]. Напряжение рассеивается, а интерес сменяется раздражением.

2. Нарушение игрового потока (Flow State)[37]. Игровой процесс строится на игровом цикле «действие → реакция → новое действие». Мозг игрока находится в *состоянии потока*, где он быстро реагирует на изменения в мире. Задержка в 30 с — это обрыв данного цикла. Игрок совершает действие, а обратная связь отсутствует. За это время Игрок потеряет фо-

[33] Idle (от англ. «бездействие», «простой») — состояние персонажа в игре, когда игрок не предпринимает никаких действий; idle-анимации — циклические движения персонажа в этом состоянии, придающие ему живость и естественность.

[34] Квестогенерация — процедурная генерация квестов, заданий и миссий в компьютерных играх с помощью заранее прописанных алгоритмов.

[35] Нарративный импульс — сочетание истории игры (сюжета, персонажей, мира) и игровых механик, которые работают вместе для создания глубокого погружения и эмоционального вовлечения игрока.

[36] Драматический момент — ключевая, напряженная сцена внутри самой игры, которая обладает сильной эмоциональной нагрузкой и зачастую является переломным пунктом в сюжете или игровом процессе.

[37] Игровой поток (Flow State) — это состояние полного погружения и сосредоточенности на игре, когда игрок теряет ощущение времени и чувствует глубокое удовлетворение от процесса.

кус и стратегический настрой. Вернуться в состояние потока после такого разрыва очень сложно. Игрок чувствует себя не активным участником, а пассивным наблюдателем, ожидающим загрузки.

3. Тактическое и стратегическое ожидание. В реальном времени каждая секунда на счету.
    - В диалоге: Игрок может ожидать ответа, чтобы принять решение: атаковать, торговаться, использовать убеждение. 30 с — это время, которое делает любое тактическое планирование бессмысленным.
    - В квестах: если квест генерируется на основе действий игрока (например, «враги захватили деревню, потому что вы ушли»), задержка в полминуты означает, что игровая ситуация может кардинально измениться еще до того, как квест будет получен. Это создает противоречивую и нелогичную ситуацию.

4. Психологический порог терпения и восприятие. Исследования UX-паттернов в IT-продуктах [15] выявили универсальные принципы, актуальные и для игровой индустрии:
    - 0.1 с: ощущается как мгновенная реакция;
    - 1.0 с: естественная, непрерывающая задержка; поток мысли пользователя не прерывается;
    - 10 с: максимальный предел для удержания внимания на задаче; пользователь начинает терять концентрацию;
    - 30 с и более: это воспринимается не как задержка, а как простой системы или сбой. Игрок с большой вероятностью решит, что игра «зависла», и попытается перезапустить ее или вовсе ее закроет.

Почему целевой показатель — менее 1 с? Именно показатель (от 100 мс до 1 с) соответствует требованиям для поддержания интерактивности в реальном времени, таким как:

- *непосредственность* – реакция системы происходит почти одновременно с действием игрока;
- *сохранение потока* – мозг не успевает переключиться на другую задачу;

- *естественность* – такая задержка сопоставима с реакцией живого собеседника или откликом интерфейса на нажатие кнопки.

Технические и практические следствия для разработки

Понимание временного ограничения (*100 мс на генерацию шага*) требует не прямолинейного, а гибридного и оптимизированного применения LLM, таких как

- кэширование и предварительная генерация – заранее (*но все же во время игрового процесса*) генерируются возможные реплики и варианты квестов, которые быстро подставляются в нужный момент;
- локальные и оптимизированные модели – использование меньших, более быстрых моделей, которые работают локально (или ближе к игровому серверу), чтобы минимизировать сетевые задержки;
- асинхронная генерация – генерация контента в фоне, пока игрок занят другими делами, а не в критический момент диалога;
- строгие ограничения по длине (tokens) – генерация очень коротких ответов, которые выполняются за доли секунды:
- offline-генерации (вместо online-генерации) – когда LLM используется для предварительного создания контента (offline), который затем используется игровым движком в режиме реального времени вместо синхронной генерации (online) во время игровой сессии.

Перечисленные рассуждения послужили основным аргументом для пересмотра архитектуры.

Переход к удаленному сервису (Server-Side AI)

Для обхода проблем с интеграцией зависимостей и изоляции игрового клиента от ресурсоемких процессов система была переведена в формат удаленного сервиса (Microservice Architecture).

Отметим преимущества серверной архитектуры.

- *Изоляция зависимостей*

  Игровой клиент (Unity) взаимодействует только по HTTP/HTTPS, полностью изолируясь от сложных C# библиотек и нативных зависимостей.

- *Централизация вычислений*

  Сервер, как единая точка обработки, выполняет все ресурсоемкие этапы (Preprocessor, Requester, Postprocessor, Analyzer).

- *Кросс-платформенность*

  Любой игровой клиент, поддерживающий HTTP-запросы, может использовать сервис.

Архитектура удаленного сервиса

На UML-диаграмме развертывания (рис. 5) представлена детальная архитектура удаленного сервера, взаимодействующего с микросервисами.

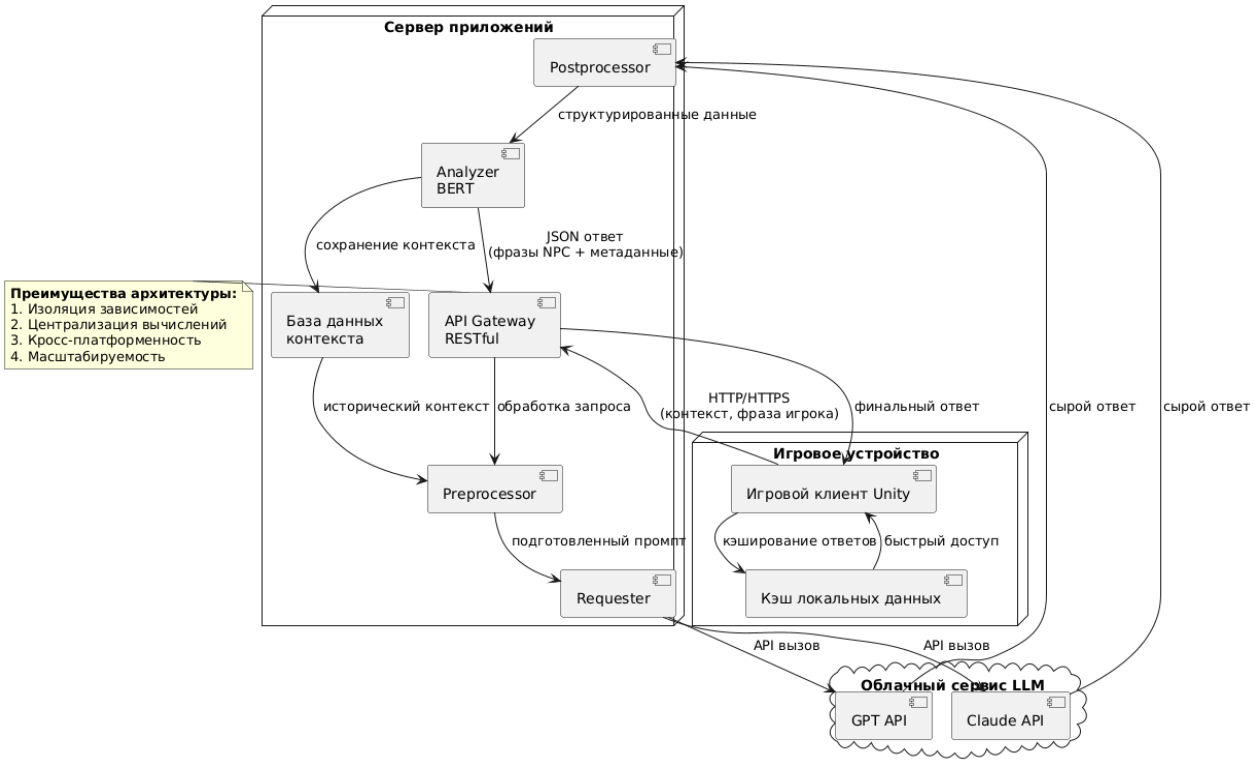

Рис. 5. Развертывание удаленного сервера с микросервисами

Игровой клиент формирует минимальный запрос, включающий контекстные данные (состояние мира, ID квеста, фраза игрока). Сервер принимает, обрабатывает запрос, вызывает облачный API LLM, анализирует ответ и возвращает клиенту финальный структурированный объект (например, JSON с готовыми фразами NPC и метаданными).

Несмотря на решение проблем интеграции, серверная архитектура попрежнему страдает от высокой задержки, вызванной облачным API LLM и необходимостью сетевого обмена с удаленным сервером. По приблизительным оценкам, даже с оптимизированным сервером задержка остается на прежнем уровне, что говорит о том, что нужно изменять подход к генерации данных, и требует дальнейших шагов для достижения целей в реальном времени.

ЛОКАЛЬНАЯ МОДЕЛЬ (EDGE AI)

В текущей стадии проекта использование облачного API GPT-3.5-Turbo для генерации диалогового хода дало среднюю задержку в 30 с. Это критическое препятствие: для поддержания погружения (иммерсивности) и сохранения состояния потока (Flow State) игрока время отклика системы на интерактивные действия должно быть менее одной с. Задержка, превышающая 1–2 с, психологически воспринимается пользователем не как ожидание ответа, а как системный сбой, что полностью разрушает интерактивный опыт, особенно в диалоговых системах.

Именно поэтому центральной задачей дальнейшего развития является достижение интерактивности в реальном времени. Это достижимо исключительно через радикальное устранение сетевой задержки и переход к парадигме Edge AI — исполнению LLM непосредственно на вычислительном устройстве пользователя.

Обоснование необходимости Edge AI и целевые показатели

Переход на локальную архитектуру Edge AI является не просто оптимизацией, а сменой фундаментального подхода, позволяющей решить три критические проблемы облачных решений:

1. *Проблема задержки*

   Облачное решение всегда подвержено вариативности сетевого трафика, времени обработки на сервере и скорости токенизации. Локальное исполнение исключает эти переменные. Наша цель — 200–500 мс на генерацию, что вплотную приближается к порогу человеческого восприятия «мгновенного» ответа. Достижение этого показателя требует не только быстрой модели, но и тесной интеграции с аппаратными ускорителями (GPU/NPU).

2. *Проблема автономности и стабильности*

   Локальная модель гарантирует работу в офлайн-режиме, что критически важно для портативных устройств или пользователей с нестабильным интернет-соединением. Система становится независимой от внешних сервисов, их тарифов и потенциальных сбоев.

3. *Проблема приватности данных*

   Вычисления, происходящие на устройстве пользователя, полностью исключают передачу чувствительных данных (контекст игры, история диалогов) на внешние серверы, обеспечивая высокий уровень конфиденциальности.

Стратегия оптимизации на основе методологии TinyML

Для внедрения LLM в игровое окружение с ограниченными ресурсами (VRAM среднего игрового ПК ≤8 GB) необходима агрессивная стратегия микрооптимизации, вдохновленная принципами TinyML.

1. Выбор и адаптация базовой модели

Выбор начнется с анализа Open-Source LLM (например, Llama 3 8B[38], Mistral 7B[39]), которые показали высокую производительность при относительно малом размере.

- *Критерий качества* – модель должна обладать достаточной «глубиной

[38] Meta-Llama-3-8B — языковая модель с 8 млрд параметров, разработанная компанией Meta в рамках семейства моделей Meta Llama 3.

[39] Mistral 7B — LLM с 7.3 млрд параметров от французского стартапа Mistral AI.

знаний» для поддержания согласованного ролевого поведения NPC и генерации логичных квестов.

- *Критерий ресурсов* **–** модель должна быть способна работать в режиме Inference[40] на пользовательском оборудовании (например, на GPU с объемом VRAM 8 ГБ или даже на мощном CPU в режиме *CPU Offload*[41]).

2. Технологии компрессии

Ключевым шагом было применение квантования. Модели с миллиардами параметров в формате FP32 занимают десятки гигабайт.

- *4-битное квантование (INT4)*

  Это наиболее радикальный и необходимый метод. Он позволяет сократить требования к памяти на 75%. При этом может быть проведен тщательный анализ потерь качества (NLG Evaluation[42]) для гарантирования сохранения «голоса» NPC и нарративной логики после компрессии. Мы использовали техники GPTQ[43] или QLoRA[44] для пост-тренировочного квантования (PTQ).

[40] Inference — рабочий режим эксплуатации обученной нейросети, при котором она выполняет свою основную функцию (генерацию диалогов и квестов) на основе входных данных в реальном времени.

[41] CPU Offload — стратегия работы с большими нейросетевыми моделями, которая позволяет запускать их на GPU с ограниченной памятью за счет хранения части данных в оперативной памяти CPU, что спасает от ошибок нехватки памяти, но значительно снижает скорость выполнения.

[42] NLG Evaluation — совокупность методов (как автоматических, так и основанных на оценке человеком) для измерения качества текста, созданного нейросетями, что особенно важно для таких задач, как генерация игровых диалогов и квестов, где принципиальны связность, увлекательность и соответствие игровому миру.

[43] GPTQ — метод послеобученного квантования, который позволяет сжать большие языковые модели до 3–4 бит на параметр с минимальной потерей качества.

[44] QLoRA — метод эффективной тонкой настройки, который позволяет адаптировать большие модели на одном GPU с ограниченной памятью.

- *Прунинг и дистилляция*

  В качестве дополнительного шага были рассмотрены методы структурного прунинга и дистилляции для создания специализированной, легковесной модели, настроенной исключительно на задачи генерации диалогов и квестов.

3. Технологический стек для игровой интеграции

Для достижения максимальной скорости Inference и кросс-платформенной совместимости необходимо использовать специализированные движки вывода нейросетей (*Inference Engines)* и низкоуровневые API:

- *Inference Engine*

  Модель была конвертирована в формат, оптимизированный для Inference (например, GGUF[45] или ONNX). Использование библиотек-оберток (например, C#-обертки над llama.cpp) позволило нам получить доступ к высокопроизводительному коду на C++, который использует SIMD-инструкции[46] и CPU Offload для эффективной работы даже при отсутствии мощного GPU.
- *Аппаратное ускорение* (Hardware Acceleration)

  Внедрение фреймворков (таких как DirectML[47] на Windows или Metal Performance Shaders[48] на macOS/iOS) позволило задейство-

[45] GGUF (GPT-Generated Unified Format) — формат файла для хранения моделей машинного обучения, специально разработанный для быстрой загрузки и выполнения на центральных процессорах (CPU).

[46] SIMD-инструкции — «одна инструкция, множество данных» (single instruction, multiple data), что означает возможность выполнять одну и ту же операцию параллельно над несколькими элементами данных с помощью одной команды.

[47] DirectML — это низкоуровневый API для машинного обучения (МО) от Microsoft, который использует аппаратное ускорение на графических процессорах (GPU) для выполнения задач МО.

[48] Metal Performance Shaders (MPS) — набор высокооптимизированных шейдеров, предоставляемых Apple в виде фреймворка для приложений, созданных с использованием Metal API.

вать весь потенциал GPU для параллельных вычислений, что критически важно для снижения задержки с 500 мс до целевых 200 мс.

- *Асинхронный стриминг*

  Для снижения воспринимаемой задержки был реализован механизм Token Streaming[49], при котором текст начинает отображаться на экране сразу после генерации первого токена, а не после полного формирования ответа моделью. Это создает иллюзию мгновенного отклика, даже если полная генерация занимает до полус.

АРХИТЕКТУРА АДАПТИВНОЙ СИСТЕМЫ ГЕНЕРАЦИИ КВЕСТОВ И ДИАЛОГОВ

На основе разработанного компонента Edge AI нами предложена гибридная, многоуровневая архитектура (см. рис. 6 и 7) для адаптивной генерации игрового контента.

[49] Token Streaming (потоковая передача токенов) — технология генерации и отображения текста языковой моделью, при которой токены (минимальные единицы текста) передаются и отображаются последовательно по мере их генерации, а не после завершения формирования полного ответа.

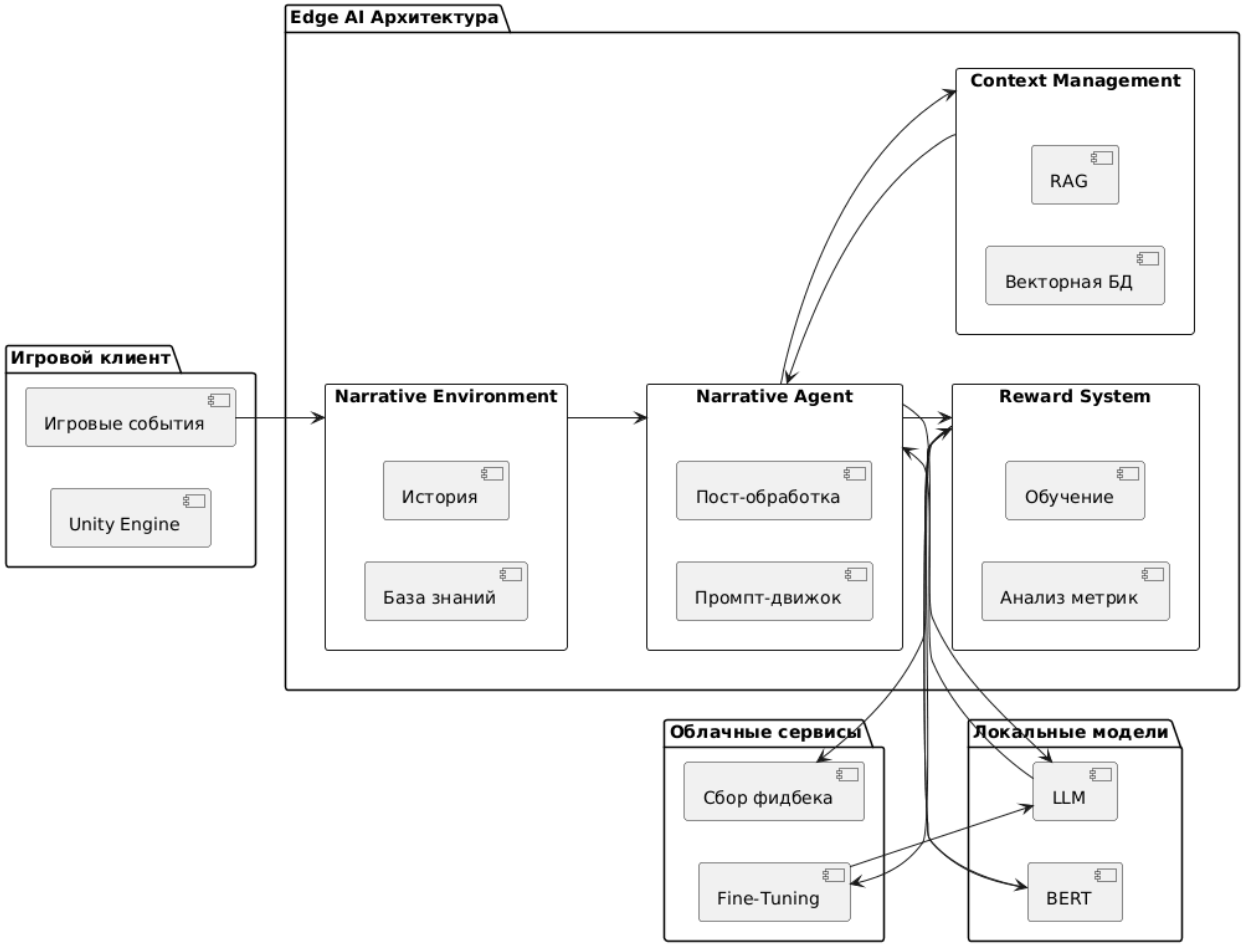

Рис. 6. Упрощенная схема усовершенствованной архитектуры

Система, представленная на рис. 6, уходит от простого промптинга и сочетает мощь LLM с традиционными подходами управления состоянием мира и элементами обучения с подкреплением (Reinforcement Learning, RL).

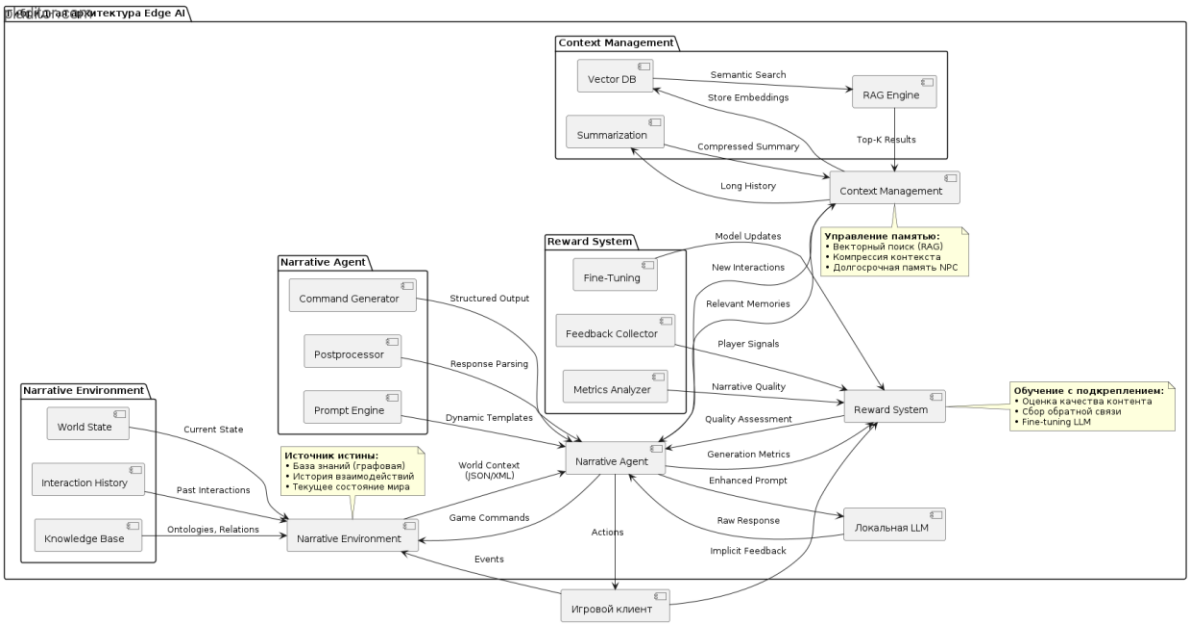

Рис. 7. Подробная схема усовершенствованной архитектуры

Архитектура на рис. 7 состоит из четырех ключевых (Narrative Environment, Narrative Agent, Context Management, Reward System for Narrative), тесно интегрированных модулей, каждый из которых выполняет свою роль в цикле адаптивной генерации.

Далее рассмотрим каждый из четырех модулей.

Модуль управления нарративной средой (Narrative Environment)

Этот модуль является источником истины для всей системы. Его основная задача — хранить, обрабатывать и предоставлять актуальное, консистентное состояние игрового мира.

- *База знаний* (Knowledge Base)

  Хранит структурированные данные о мире — онтологии (иерархия существ, предметов), карты отношений (дружба, вражда между NPC и фракциями) и ключевые нарративные точки. Используется графовая база данных (например, Neo4j или реляционная база с графовой структурой) для быстрого извлечения контекста. Для формализации пространственно-временных отношений в игровом мире и структурирования данных о состояниях может быть применен математический аппарат [16], позволяющий создать

строгую формальную модель эволюции игровых систем во времени и пространстве. Это особенно важно для отслеживания положения NPC, изменения состояний локаций и временных зависимостей между игровыми событиями.

- *История взаимодействий* (Interaction History)

  Ведет журнал всех действий игрока и сгенерированных диалогов/квестов. Этот журнал критически важен для долгосрочной памяти и предотвращения повторения контента.

- *Формат вывода*

  Модуль преобразует сложное состояние мира в структурированный, лаконичный формат (например, JSON или XML) для передачи в Narrative Agent, минимизируя «шум» промпта для LLM.

Нарративный агент (Narrative Agent)

Это интеллектуальный посредник между состоянием мира и LLM. Его задача — превратить игровые события в эффективные инструкции для генеративной модели и интерпретировать ее «сырой» вывод обратно в игровую логику.

- *Контекстуализация и промптинг.*

  Агент принимает данные о состоянии мира от Narrative Environment и, используя динамические шаблоны промптов, формирует запрос к локальной LLM. Этот промпт включает:

  - «ролевую» характеристику NPC (характер, цель, знания);
  - краткий исторический контекст (из Системы Памяти);
  - текущее событие или запрос игрока.

- *Пост-обработка и парсинг.*

  После получения ответа от LLM агент использует Postprocessor для парсинга текста и преобразования его в формализованные игровые команды (например, QUEST_START {id: "..."}, DIALOGUE_LINE {text: "..."}, NPC_ACTION {move: "..."}). Это обеспечивает безопасную и предсказуемую интеграцию генеративного контента.

Система управления контекстом и памятью (Context Management)

Этот модуль решает фундаментальную проблему всех LLM — ограниченное окно контекста и потерю долгосрочной памяти.

- *Векторная база данных* (Vector Database)

  Используется для хранения всех прошлых диалогов и ключевых событий в виде векторных эмбеддингов.
- *Retrieval-Augmented Generation* (RAG[50])

  При получении нового запроса от игрока система использует RAG-подход: ищет в векторной базе данных наиболее релевантные «воспоминания» (например, 2–3 ключевых диалога) и включает их в промпт для LLM. Это позволяет LLM генерировать ответы, которые согласуются с прошлыми взаимодействиями.
- *Модуль суммаризации*

  При переполнении окна контекста или для создания «резюме» долгой истории используется отдельная, более легкая модель для автоматической компрессии старых воспоминаний в краткое, высокоуровневое описание (например, «Игрок выполнил квест по доставке артефакта 5 дней назад»).

Модуль позволяет NPC «помнить» события, произошедшие часы назад.

Система оценки и наград (Reward System for Narrative)

Данный модуль обеспечивает обучение системы и ее адаптацию, используя принципы Reinforcement Learning from Human Feedback (RLHF) или Preference Optimization.

- *Нарративные метрики*

  Включает в себя автоматический Analyzer (возможно, на базе BERT или другой классификационной модели), который оценивает:

[50] Retrieval-Augmented Generation (RAG) — это архитектурный подход в области искусственного интеллекта, который объединяет большие языковые модели (LLM) с внешней базой знаний для повышения точности и актуальности генерируемых ответов.

1) естественность и ролевую консистентность диалога;
2) сбалансированность сгенерированного квеста (сложность, награда);
3) нарративную согласованность (отсутствие противоречий с Narrative Environment).

- *Обратная связь* (Feedback Loop)

  Система собирает неявные (время, проведенное в диалоге, выбор действий игрока) и явные (опросы игрока) данные для формирования функции награды (Reward Function). Эта награда используется для тонкой настройки (Fine-Tuning) локальной LLM, улучшая ее способность генерировать контент, который максимально соответствует ожиданиям и стилю игрока, делая систему по-настоящему адаптивной.

МЕТРИКИ ОЦЕНКИ ПЕРСОНАЛИЗИРОВАННОГО КОНТЕНТА

Для оценки качества персонализации генерируемого контента необходим ряд комплексных метрик. В отличие от традиционных метрик, ориентированных исключительно на технические параметры, предлагаемые нами метрики позволяют количественно оценивать степень адаптации контента под индивидуальные особенности игрока.

Метрика персонализации контента для оценки адаптивности игровых систем

Для количественной оценки эффективности персонализации игрового контента в разработанной системе предложена комплексная метрика, построенная по аналогии с подходами, описанными в исследованиях по адаптивным системам и рекомендательным алгоритмам. В работе [6] подчеркнута важность измерения степени соответствия генерируемого контента индивидуальным характеристикам пользователя, однако конкретная математическая формализация не приведена. Восполняет этот пробел предлагаемая метрика персонализации (от 0 до 1)

$$P = \sum_i (R_i \cdot W_i)/N$$

где R_i – релевантность контента i-му параметру пользователя, W_i – вес параметра, N – количество учитываемых параметров.

Интерпретация значений метрики персонализации осуществляется следующим образом:

- $P < 0.3$ – низкий уровень персонализации, контент практически не адаптируется под пользователя;
- $0.3 \leq P < 0.7$ – средний уровень персонализации, учитываются основные предпочтения пользователя;
- $P \geq 0.7$ – высокий уровень персонализации, контент значительно адаптирован под индивидуальные особенности игрока.

Представленная метрика позволяет оценить способность системы адаптировать генерируемые квесты и диалоги под индивидуальные особенности игрока, включая его игровой стиль, принятые ранее решения, предпочтения в прохождении и другие релевантные характеристики.

Метрика динамической адаптации для оценки соответствия контента игровому прогрессу

Для оценки способности системы адаптировать генерируемый контент в соответствии с прогрессом игрока и изменяющимся игровым контекстом предлагаем метрику динамической адаптации

$$D = 1 - \frac{|C_a - C_e|}{\mathrm{C_{max}}},$$

где C_a – фактическая сложность генерируемого контента, C_e – ожидаемая сложность на основе текущего прогресса игрока, $C_{\max}$ – максимально возможная сложность в системе.

Метрика построена на основе анализа подходов, представленных в работе [17], где отмечена важность соответствия сложности контента текущим возможностям игрока.

Интерпретация значений коэффициента динамической адаптации:

- $D < 0.6$ – недостаточная адаптация, дисбаланс сложности;
- $0.6 \leq D < 0.8$ – удовлетворительная адаптация;
- $D \geq 0.8$ – высокая степень адаптации сложности контента.

Дополнительные метрики для комплексной оценки

Можно рассмотреть и другие специализированные метрики, такие как:

- *метрика контекстной непрерывности* – измерение согласованности генерируемого контента с предысторией и текущим состоянием игрового мира;
- *метрика игрового баланса* – оценка сбалансированности наград и сложности в генерируемых квестах;
- *интегральный показатель качества контента* – общий показатель оценки качества, объединяющий уже реализованные и еще не реализованные оценки.

Такая система метрик релевантна гибридной архитектуре, когда появляется возможность тесной интеграции модуля оценки качества с механизмами генерации контента. Это позволит создать замкнутый цикл улучшения качества персонализации на основе объективных количественных данных, что особенно важно для обеспечения долгосрочной вовлеченности и качества игрового опыта.

ЗАКЛЮЧЕНИЕ

Представлено решение критической проблемы современной игровой индустрии — создания динамических и адаптивных нарративных систем, способных генерировать персонализированный контент в реальном времени. Разработанный компонент интеграции GPT продемонстрировал возможность генерации ветвящихся диалогов с учетом характеристик NPC и контекста игрового мира, при этом практическое тестирование выявило критическое ограничение облачных решений — задержку в 30 с, неприемлемую для интерактивных приложений.

Предложенная гибридная архитектура адаптивной системы представляет собой эволюцию от простой интеграции LLM к комплексному решению, сочетающему управление контекстом, векторную память и систему оценки качества. Модульная структура системы обеспечивает технологическую независимость и возможность интеграции различных языковых моделей, а меха-

низм управления контекстом на основе RAG-подхода решает фундаментальную проблему долгосрочной памяти NPC. Переход к парадигме Edge AI с целевой задержкой 200–500 мс открывает путь к достижению интерактивности в реальном времени, что критически важно для сохранения иммерсивности игрового опыта.

Дальнейшее развитие системы связано с несколькими ключевыми направлениями. Разработка и тонкая настройка специализированных языковых моделей для игровых сценариев позволят повысить качество генерируемого контента при сохранении компактности модели. Интеграция инструментов автоматической балансировки обеспечит оптимальное соответствие сложности контента уровню игрока. Реализация локальных оптимизированных моделей на основе методов квантования и прунинга устранит зависимость от облачных сервисов и связанные с ними задержки. Расширение системы мультимодальными возможностями, включая генерацию голосовых реплик и анимаций, создаст основу для следующего поколения интерактивных игровых персонажей.

Результаты работы демонстрируют практическую осуществимость создания адаптивных игровых миров, где нарративный контент генерируется динамически и персонализируется под каждого игрока, что открывает новые возможности для геймдизайна и повышения вовлеченности аудитории.

Благодарности

Работа выполнена за счет средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»).

ЛУДОГРАФИЯ

Bethesda Game Studios (2011). The Elder Scrolls V: Skyrim [Action RPG] [Multiplatform], Bethesda Softworks.

BioWare (1998). Baldur's Gate [RPG] [Windows, macOS], Interplay Entertainment, Black Isle Studios.

Epic Games (2017). Fortnite [Battle Royale, Sandbox] [Multiplatform], Epic Games.

Infocom (1980). Zork [Text adventure] [Apple II, TRS-80, PDP-10], Infocom.

Mojang Studios (2011). Minecraft [Sandbox, survival] [Multiplatform], Mojang Studios, Xbox Game Studios.

Nick Walton (2019). AI Dungeon [Text adventure, AI-generated] [Web, iOS, Android], Latitude.

Will Crowther, Don Woods (1976). Colossal Cave Adventure [Text adventure] [PDP-10].

ZA/UM (2019). Disco Elysium [RPG] [Windows, macOS, PlayStation, Xbox, Nintendo Switch], ZA/UM.

СПИСОК ЛИТЕРАТУРЫ

1. *Gallotta R. et al.* Large language models and games: A survey and roadmap // IEEE Transactions on Games. 2024.

2. Inworld, Future of NPCs report // inworld [Электронный ресурс] – February 2023. URL: https://www.inworld.ai/blog/future-of-npcs-report

3. *Sweetser P.* Large language models and video games: A preliminary scoping review // Proceedings of the 6th ACM Conference on Conversational User Interfaces. 2024. P. 1–8.

4. *Wang Q. et al.* GenQuest: An LLM-based Text Adventure Game for Language Learners // arXiv preprint arXiv:2510.04498. 2025.

5. *Hardiman J.P.W. et al.* AI-powered dialogues and quests generation in role-playing games using Google's Gemini and Sentence BERT framework // Procedia Computer Science. 2024. Vol. 245. P. 1111–1119.

6. *Ashby T. et al.* Personalized quest and dialogue generation in role-playing games: A knowledge graph-and language model-based approach // Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023. P. 1–20.

7. *Большаков Э.С., Кугуракова В.В.* Генеративная симуляция игрового окружения в реальном времени // Электронные библиотеки. 2025. Т. 28, № 2. С. 188–212.

8. *Нурлыгаянов Н.Р., Кугуракова В.В.* Подход к созданию корпуса текстов видеоигр на основе универсальной структуры // Электронные библиотеки. 2024. Т. 27, № 4. С. 578–597.

9. *Akoury N., Yang Q., Iyyer M.* A framework for exploring player perceptions of llm-generated dialogue in commercial video games // Findings of the Association for Computational Linguistics: EMNLP 2023. 2023. P. 2295–2311.

10. *Jin C., Cao P., Zaïane O.* Role-Playing Based on Large Language Models via Style Extraction // International Conference on Neural Information Processing. Singapore: Springer Nature Singapore, 2024. P. 433–447.

11. *Tseng Y.M. et al.* Two tales of persona in llms: A survey of role-playing and personalization // arXiv preprint arXiv:2406.01171. 2024.

12. *Трофимчук В.Т.* Разработка компонента для интеграции GPT в видеоигры: выпуск. квалиф. раб. на с.з. Бакалавр, спец. 09.03.04 – Программная инженерия, науч. рук. Хафизов М.Р., Казанский федеральный университет, Институт информационных технологий и интеллектуальных систем, 2024. 54 с. URL: https://kpfu.ru/student_diplom/10.160.178.20_FLP3APBW54SAM3JYPPT73DBLRFUS75DXQEBT_Z5F6LD7O0KAF7_F_Trofimchuk.pdf

13. *Abdelrahman E.* Edge AI and Edge Computing: Powering Real-Time Intelligence [Электронный ресурс] // Ultralytics.
URL: https://www.ultralytics.com/ru/blog/edge-ai-and-edge-computing-powering-real-time-intelligence

14. *Кудерин Д.* Edge AI: как работают нейросети на устройствах с ограниченными ресурсами [Электронный ресурс] // TProger.
URL: https://tproger.ru/articles/edge-ai--kak-rabotayut-nejroseti-na-ustrojstvah-s-ogranichennymi-resursami

15. *Martindale J.* Input lag and response time aren’t the same. Here’s which is more important [Электронный ресурс]. 2024.
URL: https://www.digitaltrends.com/computing/input-lag-vs-response-time/

16. *Кугуракова В.В.* Формальный подход к пространственно-временному моделированию игровых систем // Ученые записки Казанского университета. Серия Физико-математические науки. 2024. Т. 166, №. 4. С. 532–554.

17. *Chen B.* Optimization Strategies for Role-Playing Games Based on Large Language Models // Proceedings of the 2nd International Conference on Data Science and Engineering: ICDSE 2025. P. 632–637.

18. *Сахибгареева Г.Ф., Кугуракова В.В., Большаков Э.С.* Инструменты балансирования игр // Электронные библиотеки. 2023. Т. 26, № 2. С. 225–251.

DEVELOPMENT OF AN ADAPTIVE SYSTEM FOR GENERATING GAME QUESTS AND DIALOGUES BASED ON LARGE LANGUAGE MODELS

V. T. Trofimchuk[1] [0009-0001-9106-9614], V. V. Kugurakova[2] [0000-0002-1552-4910]

[1, 2] Institute of Information Technologies and Intelligent Systems, Kazan Federal University, 35 Kremlyovskaya st., Kazan, 420008

[1] vselord.beta@gmail.com, [2] vlada.kugurakova@gmail.com

Abstract

This article addresses the problem of creating dynamic narrative systems for video games with real-time interactivity. It presents the development and testing of a GPT integration component for dialogue generation, which revealed a critical limitation of cloud-based solutions – a 30-second latency unacceptable for gameplay. A hybrid architecture of an adaptive system is proposed, combining LLMs with reinforcement learning mechanisms. Particular attention is given to solving the problems of game world consistency and managing long-term context of NPC interactions through a RAG approach. The transition to the Edge AI paradigm with the application of quantization methods to achieve a target latency of 200–500 ms is substantiated. Metrics for evaluating personalization and dynamic content adaptation have been developed.

Keywords: *video games, large language models, LLM, dialogue generation, quest generation, adaptive quests, procedural content generation, agent behavior, game AI, machine learning in games.*

Acknowledgment: This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY–2030").

LUDOGRAPHY

Bethesda Game Studios (2011). The Elder Scrolls V: Skyrim [Action RPG] [Multiplatform], Bethesda Softworks.

BioWare (1998). Baldur's Gate [RPG] [Windows, macOS], Interplay Entertainment, Black Isle Studios.

Epic Games (2017). Fortnite [Battle Royale, Sandbox] [Multiplatform], Epic Games.

Infocom (1980). Zork [Text adventure] [Apple II, TRS-80, PDP-10], Infocom.

Mojang Studios (2011). Minecraft [Sandbox, survival] [Multiplatform], Mojang Studios, Xbox Game Studios.

Nick Walton (2019). AI Dungeon [Text adventure, AI-generated] [Web, iOS, Android], Latitude.

Will Crowther, Don Woods (1976). Colossal Cave Adventure [Text adventure] [PDP-10].

ZA/UM (2019). Disco Elysium [RPG] [Windows, macOS, PlayStation, Xbox, Nintendo Switch], ZA/UM.

REFERENCES

1. *Gallotta R. et al.* Large language models and games: A survey and roadmap // IEEE Transactions on Games. 2024.

2. Inworld. Future of NPCs report // inworld [Electronic resource]. – February 2023. URL: https://www.inworld.ai/blog/future-of-npcs-report

3. *Sweetser P.* Large language models and video games: A preliminary scoping review // Proceedings of the 6th ACM Conference on Conversational User Interfaces. 2024. P. 1–8.

4. *Wang Q. et al.* GenQuest: An LLM-based Text Adventure Game for Language Learners // arXiv preprint arXiv:2510.04498. 2025.

5. *Hardiman J.P.W. et al.* AI-powered dialogues and quests generation in role-playing games using Google's Gemini and Sentence BERT framework // Procedia Computer Science. 2024. Vol. 245. P. 1111–1119.

6. *Ashby T. et al.* Personalized quest and dialogue generation in role-playing games: A knowledge graph-and language model-based approach // Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023. P. 1–20.

7. *Bolshakov E.S., Kugurakova V.V.* Generative simulation of a game environment in real time // Russian Digital Libraries Journal. 2025. Vol. 28, No. 2. P. 188–212 (In Russian).

8. *Nurygayanov N.R., Kugurakova V.V.* An approach to creating a corpus of video game texts based on a universal structure // Russian Digital Libraries Journal. 2024. Vol. 27, No. 4. P. 578–597 (In Russian).

9. *Akoury N., Yang Q., Iyyer M.* A framework for exploring player perceptions of llm-generated dialogue in commercial video games // Findings of the Association for Computational Linguistics: EMNLP 2023. P.2295–2311.

10. *Jin C., Cao P., Zaïane O.* Role-Playing Based on Large Language Models via Style Extraction // International Conference on Neural Information Processing. Singapore: Springer Nature Singapore, 2024. P. 433–447.

11. *Tseng Y.M. et al.* Two tales of persona in llms: A survey of role-playing and personalization // arXiv preprint arXiv:2406.01171. 2024.

12. *Trofimchuk V.T.* Development of a component for GPT integration into video games: Bachelor's qualifying work, spec. 09.03.04 – Software Engineering, scientific supervisor Khafizov M.R., Kazan Federal University, Institute of Information Technology and Intelligent Systems, 2024. 54 p.
URL: https://kpfu.ru/student_diplom/10.160.178.20_FLP3APBW54SAM3JYPPT73DBLRFUS75DXQEBT_Z5F6LD7O0KAF7_F_Trofimchuk.pdf (In Russian)

13. *Abdelrahman E.* Edge AI and Edge Computing: Powering Real-Time Intelligence [Electronic resource] // Ultralytics.
URL: https://www.ultralytics.com/ru/blog/edge-ai-and-edge-computing-powering-real-time-intelligence

14. *Kuderin D.* Edge AI: how neural networks work on devices with limited resources [Electronic resource] // TProger.

URL: https://tproger.ru/articles/edge-ai--kak-rabotayut-nejroseti-na-ustrojstvah-s-ogranichennymi-resursami (In Russian).

15. *Martindale J.* Input lag and response time aren't the same. Here's which is more important [Electronic resource]. 2024.
URL: https://www.digitaltrends.com/computing/input-lag-vs-response-time/

16. *Kugurakova V.V.* A formal approach to spatio-temporal modeling of game systems // Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki. 2024. Vol. 166, No. 4. P. 532–554 (In Russian).

17. *Chen B.* Optimization Strategies for Role-Playing Games Based on Large Language Models // Proceedings of the 2nd International Conference on Data Science and Engineering: ICDSE 2025. P. 632–637.

18. *Sakhibgareeva G.F., Kugurakova V.V., Bolshakov E.S.* Game balancing tools // Russian Digital Libraries Journal. 2023. Vol. 26, No. 2. P. 225–251 (In Russian).

СВЕДЕНИЯ ОБ АВТОРАХ

ТРОФИМЧУК Всеволод Тарасович – лаборант-исследователь научно-исследовательской лаборатории Digital Media Lab Института ИТИС КФУ. Область научных интересов – использование LLM для видеоигр в реальном времени.

Vsevolod Tarasovich TROFIMCHUK – laboratory research assistant at Digital Media Lab of the Institute of ITIS KFU. Area of research interests - using LLM for real-time video games.

email: vselord.beta@gmail.com

ORCID: 0009-0001-9106-9614

КУГУРАКОВА Влада Владимировна – кандидат технических наук, и. о. зав. кафедрой индустрии разработки видеоигр Института ИТИС КФУ, руководитель НИЛ Digital Media Lab. Область научных интересов – формальные методы верификации видеоигр.

Vlada Vladimirovna KUGURAKOVA – Ph.D. of Engineering Sciences, Head of the Video Game Development Industry Department of ITIS KFU, Head of Laboratory «Digital Media Lab». The area of scientific interest is formal methods of video game verification.

email: vlada.kugurakova@gmail.com

ORCID: 0000-0002-1552-4910

Материал поступил в редакцию 10 сентября 2025 года