

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Магистерская программа  
«Исследования и предпринимательство  
в искусственном интеллекте»

СОГЛАСОВАНО

Научный руководитель,  
доцент ФКН департамента  
программной инженерии, к.т.н.

\_\_\_\_\_ С. Л. Макаров  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

УТВЕРЖДАЮ

Академический руководитель  
образовательной программы  
«Исследования и предпринимательство  
в искусственном интеллекте»

\_\_\_\_\_ Д. С. Лялин  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**Курсовая работа  
(проектная)**

на тему: **Разработка мультиагентной текстовой стратегической игры на  
основе оркестрируемых языковых моделей**

по направлению подготовки 01.04.02 «Прикладная математика и информатика»

ВЫПОЛНИЛ

студент группы ИПИИ  
образовательной программы  
01.04.02 «Прикладная  
математика и информатика»

\_\_\_\_\_ Н. С. Пеганов  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

# Реферат

Работа посвящена разработке мультиагентной текстовой стратегической игры на основе оркестрируемых языковых моделей и исследованию применимости современных LLM в качестве автоматизированного мастера игры в жанре военно-политических игр (ВПИ).

В работе рассмотрен полный цикл создания подобной системы, включая проектирование архитектуры мультиагентной среды, оркестрацию языковых моделей для выполнения различных игровых функций, а также анализ результатов пилотного запуска с реальными пользователями. Особое внимание уделяется методам преодоления типичных проблем языковых моделей в контексте игрового процесса: галлюцинации, сохранение долгосрочного контекста и вычислительные ограничения.

Результаты исследования демонстрируют потенциал современных LLM для автоматизации роли вердера в текстовых стратегических играх, выявляют ключевые технические и игровые ограничения существующих подходов, а также предлагают набор технических решений для повышения качества игрового опыта, включая применение систем на основе RAG и локальных моделей для снижения стоимости эксплуатации игровой системы.

Данная работа состоит из 12 страниц, 2 глав, 5 листингов, 1 таблицы, 2 приложений. Использовано 9 источников.

**Ключевые слова:** мультиагентная система; языковые модели; оркестрация LLM; текстовые стратегические игры; искусственный интеллект в играх; RAG; автоматизация игрового мастера.

# Abstract

This paper is dedicated to developing a multi-agent text-based strategy game using orchestrated large language models, and investigating the applicability of modern LLMs as automated game masters in the military-political games (WPG) genre.

The work covers the complete cycle of creating such a system, including designing the architecture of a multi-agent environment, orchestrating language models to perform various game functions, and analyzing the results of a pilot launch with real users. Special attention is paid to methods for overcoming typical language model problems in the gaming context: hallucinations, maintaining long-term context, and computational limitations.

The research results demonstrate the potential of modern LLMs for automating the role of game arbiter in text-based strategy games, identify key technical and gameplay limitations of existing approaches, and propose a set of technical solutions to enhance gaming experience, including the application of RAG-based systems and local models to reduce the operational costs of the game system.

The paper contains 12 pages, 2 chapters, 5 listings, 1 table, 2 appendices. 9 sources are used.

**Keywords:** multi-agent system; large language models; LLM orchestration; text-based strategy games; artificial intelligence in games; RAG; automated game mastering.

# Содержание

Реферат . . . . .	2
Abstract . . . . .	3
Используемые определения и термины . . . . .	5
Введение . . . . .	7
Глава 1 Обзор источников . . . . .	8
1.1 Какая-то подглава . . . . .	8
1.1.1 Какая-то подподглава . . . . .	8
1.1.1.1 Какой-то параграф . . . . .	8
1.1.1.2 Какой-то параграф . . . . .	8
1.1.1.3 Какой-то параграф . . . . .	8
1.1.1.4 Какой-то параграф . . . . .	8
1.1.2 Какая-то подподглава . . . . .	8
1.1.2.1 Какой-то параграф . . . . .	8
1.1.2.2 Какой-то параграф . . . . .	8
1.1.2.3 Какой-то параграф . . . . .	8
1.1.2.4 Какой-то параграф . . . . .	9
Выводы по главе . . . . .	9
Глава 2 Какая-нибудь ещё глава . . . . .	10
Заключение . . . . .	11
Список использованных источников . . . . .	12
Приложение А . . . . .	13
Приложение Б . . . . .	16

## Используемые определения и термины

**RAG (Retrieval-Augmented Generation)** – метод улучшения генерации текста языковыми моделями путем предварительного извлечения релевантной информации из внешних источников.

**Большая языковая модель (Large Language Model, LLM)** – языковая модель значительного размера, способная генерировать когерентный текст и выполнять различные языковые задачи.

**Верд (вердикт)** – текстовое описание результатов выполнения приказов игроков, составляемое вердером.

**Виртуальное государство** – поджанр ВПИ, концентрирующийся на отыгрыше политики в рамках одного государства с детализацией механик принятия государственных решений.

**Военно-политические игры (ВПИ, Military-Political Games, WPG)** – жанр текстовых стратегических игр, в которых игроки управляют государствами, политическими фракциями или организациями, взаимодействуя посредством письменных приказов и получая ответные вердикты от мастера игры.

**Галлюцинации LLM** – явление, при котором языковая модель генерирует фактически неверную информацию, представляя её как достоверную.

**Калькулятор** – принцип организации игровой механики, при котором взаимодействие с миром осуществляется через численные переменные и формулы, обеспечивающие определенный уровень автоматизации.

**Классическая ВПИ** – проект, где в рамках игровой механики предусмотрен отыгрыш правителя государства с возможностью действий во всех сферах государственной политики.

**Командно-штабная игра (КШИ)** – разновидность ВПИ с акцентом на военной составляющей, где игроки делятся на команды, отыгрывающие офицерский состав заранее определенных армий.

**Мультиагентная система** – система, состоящая из нескольких взаимодействующих интеллектуальных агентов, каждый из которых выполняет определенную функцию.

**Оркестрация языковых моделей** – процесс координации нескольких языковых моделей или компонентов для последовательного выполнения сложных задач.

**Приказ** – сформулированная в текстовой форме воля игрока, использующая находящиеся под контролем игрока силы для преобразования внешнего мира в рамках установленных игрой правил.

**Рестарт** – перезапуск проекта и смена игровой сессии.

**Ролеплей (РП)** – принцип организации игровой механики, при котором взаимодействие с игровой реальностью осуществляется через прямое текстовое взаимодействие игрока и судьи.

**Сессия** – игровой процесс, проходящий в рамках одной «игровой реальности», непрерывный процесс отыгрыша в определённой вселенной без удаления игроков и обнуления прогресса.

**Сеттинг** – совокупность особенностей среды, в рамках которой протекает игра, включая историю мира, технологический уровень, географию и культурные особенности.

**Судья (вердер)** – человек, наделенный полномочиями определять реакцию внешнего мира на действия игрока и описывать в текстовом формате итоги приказов, аналог гейм-мастера.

**Языковая модель (Language Model, LM)** – алгоритмическая система, обученная предсказывать и генерировать текст на естественном языке.

# Введение

Современные достижения в области искусственного интеллекта и, в частности, больших языковых моделей (LLM) открывают новые перспективы для традиционных интерактивных развлечений [1; 2]. Одной из областей, где применение искусственного интеллекта имеет значительный потенциал, являются текстовые стратегические игры, такие как военно-политические игры (ВПИ) — жанр, сочетающий элементы стратегии, ролевой игры и коллективного сторителлинга [3].

ВПИ представляют собой текстовые игры, в которых игроки управляют сложными структурами (государствами, военными силами, политическими организациями) путем написания приказов, а судья (или вердер) оценивает эти приказы и формирует вердикты — текстовые описания результатов действий [4]. Этот процесс требует от судьи глубокого понимания игрового мира, механик, а также способности генерировать связные и логичные повествования, что делает эту роль одной из самых трудоемких в организации игры. Кроме того, традиционно судья ограничен в скорости обработки приказов, что создает естественный «потолок» для темпа игры и количества участников.

Большие языковые модели, такие как GPT-4, демонстрируют впечатляющие способности к пониманию контекста, следованию инструкциям и генерации связных текстов [5]. Эти характеристики потенциально позволяют им выполнять роль судьи в ВПИ, автоматизируя процесс создания вердиктов и значительно ускоряя игровой процесс. Однако использование LLM в таком качестве сопряжено с рядом технических и методологических вызовов, включая проблему галлюцинаций, ограничения контекстного окна и сложности в поддержании долговременной согласованности [6; 7].

В данной работе представлена разработка мультиагентной текстовой стратегической игры на основе оркестрируемых языковых моделей — системы, использующей несколько специализированных LLM-агентов для выполнения различных функций судьи в ВПИ. Исследование включает как теоретическое обоснование подхода, так и практическую реализацию в виде работающего прототипа, протестированного реальными игроками. Особое внимание уделяется механизмам обеспечения целостности игрового мира, преодоления ограничений LLM и создания интуитивно понятного пользовательского опыта.

Работа основывается на междисциплинарном подходе, объединяющем методы искусственного интеллекта, игрового дизайна и нарративных исследований [8; 9]. Представленная система не только демонстрирует практическое применение современных LLM в новой предметной области, но и открывает перспективы для создания более масштабных и динамичных текстовых игр, доступных широкой аудитории.

# Глава 1. Обзор источников

Текст главы 1

## 1.1. Какая-то подглава

Текст подглавы

### 1.1.1. Какая-то подподглава

Текст подподглавы

#### 1.1.1.1. Какой-то параграф

Текст параграфа

#### 1.1.1.2. Какой-то параграф

Текст параграфа

#### 1.1.1.3. Какой-то параграф

Текст параграфа

#### 1.1.1.4. Какой-то параграф

Текст параграфа

### 1.1.2. Какая-то подподглава

Текст подподглавы

#### 1.1.2.1. Какой-то параграф

Текст параграфа

#### 1.1.2.2. Какой-то параграф

Текст параграфа

#### 1.1.2.3. Какой-то параграф

Текст параграфа



#### **1.1.2.4. Какой-то параграф**

Текст параграфа

#### **Выводы по главе**

Текст Текст Текст Текст Текст Текст Текст

## Глава 2. Какая-нибудь ещё глава

Текст главы 2

## Заключение

Текст заключения

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Language Models are Few-Shot Learners / Т. В. Brown [и др.] // Advances in Neural Information Processing Systems. — 2020. — Т. 33. — С. 1877—1901.
2. Training language models to follow instructions with human feedback / L. Ouyang [и др.] // Advances in Neural Information Processing Systems. — 2022. — Т. 35. — С. 27730—27744.
3. Каталог ВПИ [Электронный ресурс] : Сообщество военно-политических игр. — URL: <https://vk.com/catalogwpg> (дата обр. 15.10.2023).
4. Что такое военно-политические игры (ВПИ)? [Электронный ресурс] : Введение в жанр текстовых стратегических игр. — URL: <https://vk.com/@catalogwpg-what-is-wpg> (дата обр. 20.11.2023).
5. *OpenAI*. GPT-4 Technical Report : тех. отч. / OpenAI. — 2023.
6. Sparks of Artificial General Intelligence: Early experiments with GPT-4 / S. Bubeck [и др.] // arXiv preprint arXiv:2303.12712. — 2023.
7. *Liu S., Ogren P., Peng N.* Evaluating the Factual Consistency of Large Language Models Through News Summarization // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. — 2023. — С. 10574—10587.
8. Wordcraft: Story Writing With Large Language Models / A. Yuan [и др.] // 27th International Conference on Intelligent User Interfaces. — ACM. 2022. — С. 841—852.
9. Generative Agents: Interactive Simulacra of Human Behavior / J. S. Park [и др.] // Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. — 2023.

## Пример приложения

Пример приложения. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст. Какой-то текст.

Ссылка на приложение Б.

Тут ссылка на листинг 1.

А тут ссылка на листинг 3.

```

1 | @Deprecated("Reason")
2 | fun findScriptDefinition(project: Project, script: SourceCode): ScriptDefinition?
   | {
3 |     val scriptDefinitionProvider = ScriptDefinitionProvider.getInstance(project) ?:
   |         return null
4 |     ?: throw IllegalStateException("Unable to get script definition: ...")
5 |
6 |     return scriptDefinitionProvider.findDefinition(script) ?:
   |         scriptDefinitionProvider.getDefaultDefinition() // Comment
7 | }

```

Листинг 1 — Пример какого-то кода на Kotlin

```

1 | class Main {
2 |     public static ScriptDefinition findScriptDefinition(Project project, SourceCode
   |         script) {
3 |         ScriptDefinitionProvider scriptDefinitionProvider = ScriptDefinitionProvider.
   |             getInstance(project);
4 |         if (scriptDefinitionProvider == null) {
5 |             if (null == null) {
6 |                 throw IllegalStateException("Unable to get script definition: ...");
7 |             } else {
8 |                 return null;
9 |             }
10 |        }
11 |
12 |        ScriptDefinition definition = scriptDefinitionProvider.findDefinition(script);
13 |        if (definition == null) {
14 |            return scriptDefinitionProvider.getDefaultDefinition(); // Comment
15 |        } else {
16 |            return definition;
17 |        }
18 |    }
19 | }

```

Листинг 2 — Пример какого-то кода на Java

```

...
13 aload_2
14 dup
15 ifnonnull 28
18 new #17 // NullPointerException
21 dup
22 ldc #19 // String null cannot be cast to non-null String
24 invokespecial #23 // NullPointerException."<init>"(String)
27 athrow
...
46 aload_2
47 dup
48 ifnonnull 61
51 new #17 // NullPointerException
54 dup
55 ldc #19 // String null cannot be cast to non-null String
57 invokespecial #23 // NullPointerException."<init>"(String)
60 athrow
...

```

Листинг 3 — Пример JVM-байткода

```

...
13: aload_2
14: dup
15: ifnonnull 28
18: new #17 // NullPointerException
21: dup
22: ldc #19 // String null cannot be cast to non-null String
24: invokespecial #23 // NullPointerException."<init>"(String)
27: athrow
...
46: aload_2
47: dup
48: ifnonnull 61
51: new #17 // NullPointerException
54: dup
55: ldc #19 // String null cannot be cast to non-null String
57: invokespecial #23 // NullPointerException."<init>"(String)
60: athrow
...

```

Листинг 4 — Пример JVM-байткода 2

А тут ссылка на таблицу 1.

Таблица 1 — Пример таблицы

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

```
1 procedure RUN(packages, hashes)
2   queue[svace.parallel_max]
3   for item ∈ zip(packages, hashes)
4     ps = create(item)
5     if !queue.full()
6       queue.put(ps)
7     else
8       first = queue.get()
9       first.wait()
10    end if
11  end for
12 end procedure
```

Листинг 5 — Пример псевдокода на алгоритмическом языке

## Ещё один пример приложения

Пример приложения