

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)
Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

_____ Розробка моделі генерації сценаріїв у системі правил Dungeons & Dragons

(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-4

_____ Ксенія ГРИНЧЕНКО
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна
Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник проф. Марина КУДРЯВЦЕВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ІІІ

_____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Гринченко Ксенії Миколаївні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка моделі генерації сценаріїв у системі правил Dungeons & Dragons _____

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 8 липня 2026 р.

3. Вихідні дані до роботи Науково-технічні публікації та провідні дослідження у галузі процедурної генерації контенту (PCG), великих мовних моделей (LLM) та архітектур Retrieval-Augmented Generation (RAG); офіційна документація мови програмування Python та платформи Ollama; зводи правил ігрової системи Dungeons & Dragons (Player's Handbook, Dungeon Master's Guide, Monster Manual); структуровані набори даних у форматі JSON для формування бази сюжетних компонентів.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі _____

2) Розробка алгоритмів побудови пригоди _____

3) Програмна реалізація та експериментальне дослідження _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі, та постановка задачі	08.06.2026–09.06.2026	виконано
3	Огляд моделей обробки природної мови, семантичного пошуку та концепцій RAG	10.06.2026–13.06.2026	виконано
4	Проектування формальної моделі пригоди та алгоритмів перевірки сумісності компонентів	13.06.2026–15.06.2026	виконано
5	Програмна реалізація та експериментальне дослідження	15.06.2026–20.06.2026	виконано
6	Оформлення пояснювальної записки	20.06.2026–23.06.2026	виконано
7	Нормконтроль	24.06.2026	виконано
8	Підготовка презентації та доповіді	25.06.2026–30.06.2026	виконано
9	Попередній захист	01.07.2026	виконано
10	Захист перек ЕК	09.07.2026	

Дата видачі завдання 8 червня 2026 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

проф. Марина КУДРЯВЦЕВА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 88 с., 13 рис., 5 табл., 1 дод., 23 джерела.

АВТОМАТИЗОВАНА ГЕНЕРАЦІЯ, АДАПТИВНЕ ОЦІНЮВАННЯ, ВАНШОТ, ВЕЛИКА МОВНА МОДЕЛЬ, ГІБРИДНА АРХІТЕКТУРА, ДИЗАЙН ПРИГОД, КОМПОНЕНТНЕ МОДЕЛЮВАННЯ, КОСИНУСНА ПОДІБНІСТЬ, СЕМАНТИЧНИЙ ПОШУК, DUNGEONS & DRAGONS.

Об'єкт дослідження – процес автоматизованої генерації сценаріїв пригод для настільних рольових ігор.

Предмет дослідження – моделі, методи та алгоритми формування структурно узгоджених сценаріїв ваншотів у системі правил «Dungeons & Dragons».

Мета роботи – розробка моделі та програмної системи генерації сценаріїв пригод-ваншотів для системи правил «Dungeons & Dragons», що забезпечує логічну цілісність сценарію, відповідність ігровим правилам та можливість адаптації результату до побажань користувача.

Методи дослідження – теоретичний, алгоритмічний та експериментальний. У роботі використано методи обробки природної мови, семантичного пошуку, процедурної генерації контенту, евристичної фільтрації та великі мовні моделі. Програмну реалізацію виконано мовою Python із використанням формату JSON та платформи Ollama.

Значимість роботи полягає у створенні програмної системи, яка забезпечує автоматизовану генерацію структурно узгоджених сценаріїв пригод і може бути використана як основа для подальшого розвитку інтелектуальних систем автоматизованого наративного дизайну.

ABSTRACT

Bachelor's thesis: 88 pp., 13 fig., 5 tabl., 1 ann., 23 references.

ADAPTIVE EVALUATION, AUTOMATED GENERATION, COMPONENT-BASED MODELING, COSINE SIMILARITY, DUNGEONS & DRAGONS, HYBRID ARCHITECTURE, LARGE LANGUAGE MODEL, ONE-SHOT ADVENTURE, SEMANTIC SEARCH, STORY DESIGN.

Object of the research – the process of automated generation of adventure scenarios for tabletop role-playing games.

Subject of the research – models, methods, and algorithms for generating structurally consistent one-shot adventure scenarios within the «Dungeons & Dragons» ruleset.

Purpose of the thesis – to develop a model and a software system for generating one-shot adventure scenarios for the «Dungeons & Dragons» ruleset, ensuring logical consistency, compliance with game mechanics, and adaptation of the generated content to user preferences.

Research methods – theoretical, algorithmic, and experimental methods. The study employs natural language processing techniques, semantic search algorithms, procedural content generation methods, heuristic filtering algorithms, and large language models. The software implementation was developed in Python using the JSON data format and the Ollama platform for local deployment of language models.

Significance of the research lies in the development of a software system for the automated generation of structurally consistent adventure scenarios, which can serve as a foundation for the further development of intelligent automated narrative design systems.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі та постановка задачі.....	10
1.1 Особливості генерації сценаріїв у настільних рольових іграх.....	10
1.2 Аналіз існуючих підходів та рішень	12
1.2.1 Генератори на основі випадкового вибору та шаблонів.....	12
1.2.2 Правилоорієнтовані системи генерації сюжетів.....	13
1.2.3 ШІ-системи генерації тексту	15
1.2.4 Генеративно-пошукові системи та системи процедурної генерації контенту.....	16
1.2.5 Агентні системи	17
1.2.6 Порівняльний аналіз існуючих підходів та підсумки	18
1.3 Постановка задачі дослідження.....	21
2 Розробка алгоритмів побудови пригоди.....	23
2.1 Концептуальна модель процесу генерації пригод.....	23
2.2 Вибір сюжетних компонентів	27
2.3 Основи семантичного пошуку та моделей обробки природної мови	30
2.4 Формальна модель пригоди	34
2.5 Модель перевірки сумісності компонентів	37
2.6 Формування Adventure Blueprint на основі користувацького запиту	41
2.7 Алгоритм генерації текстового сценарію	43
2.7.1 Формування контексту генерації	43
2.7.2 Побудова інструкції генерації	44
2.7.3 Генерація текстового представлення сценарію	46
2.8 Модель адаптивного оцінювання сюжетних компонентів	52
3 Програмна реалізація та експериментальне дослідження	55
3.1 Реалізація програмного засобу	55
4.2 Реалізація бази сюжетних компонентів	58
4.3 Сценарії тестування системи	61

3.4 Аналіз результатів генерації	64
3.5 Дослідження впливу користувацького зворотного зв'язку на якість генерації	68
3.6 Реалізація інтелектуального компонента системи	71
3.7 Порівняння результатів генерації з прямим використанням великої мовної моделі.....	73
Висновки	83
Перелік джерел посилання	85
Додаток А Відомість кваліфікаційної роботи	88

ВСТУП

Сучасний розвиток інформаційних технологій, систем штучного інтелекту та методів обробки природної мови сприяє активному впровадженню інтелектуальних засобів автоматизованого створення цифрового контенту. Подібні технології широко застосовуються у сферах генерації текстів, інтерактивного сторітелінгу, комп'ютерних ігор, систем підтримки творчої діяльності та персоналізованих інформаційних сервісів. Одним із цікавих напрямів досліджень є автоматизація процесів створення сюжетного контенту для настільних рольових ігор із використанням методів штучного інтелекту та процедурної генерації.

Особливе місце серед настільних рольових ігор займає система «Dungeons & Dragons», яка протягом багатьох років залишається одним із найпоширеніших стандартів у даній галузі. Її популярність обумовлена гнучкістю ігрового процесу, великою кількістю доступного контенту та можливістю створення унікальних пригод. Водночас підготовка якісного сценарію вимагає значних часових витрат і високого рівня підготовки майстра гри. До його обов'язків входить формування сюжетної структури пригоди, створення локацій та неігрових персонажів, визначення конфліктів, побудова логічних зв'язків між подіями та забезпечення відповідності пригоди правилам ігрової системи.

Автоматизація процесу створення пригод є незвичайною науково-технічною задачею, яка поєднує елементи наративного дизайну, процедурної генерації контенту, семантичного аналізу тексту та інтелектуального синтезу природної мови. Для отримання якісного результату необхідно забезпечити узгодженість між усіма компонентами пригоди, зберегти логічну цілісність сюжетної лінії, підтримати відповідність тематиці та жанру, а також врахувати особливості ігрових механік «Dungeons & Dragons».

У сучасних дослідженнях для вирішення подібних задач активно використовуються методи процедурної генерації контенту, алгоритми аналізу природної мови, трансформаторні архітектури та великі мовні моделі. Їх використання дозволяє автоматизувати створення сюжетів, персонажів, діалогів та описів ігрового світу, суттєво розширюючи можливості сучасних систем генерації контенту.

Таким чином, розробка моделей автоматизованої генерації пригод для настільних рольових ігор є актуальною науково-практичною задачею, спрямованою на підвищення ефективності створення ігрового контенту та розширення можливостей використання сучасних методів штучного інтелекту в системах інтерактивного сторітелінгу.

У роботі використовуються методи обробки природної мови, алгоритми семантичного аналізу тексту, методи процедурної генерації контенту, евристичні алгоритми відбору та оцінювання компонентів, а також сучасні мовні моделі для формування текстового представлення пригод.

Практична цінність роботи полягає у створенні програмного засобу автоматизованої генерації сценаріїв ваншотів для системи правил «Dungeons & Dragons», який може використовуватися майстрами гри для швидкої підготовки ігрового контенту, у системах підтримки інтерактивного сторітелінгу та навчально-дослідницьких проєктах, а також слугувати основою для подальших досліджень у сфері автоматизованого наративного дизайну та інтелектуальних систем генерації сюжетів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Особливості генерації сценаріїв у настільних рольових іграх

Настільні рольові ігри є системами інтерактивної наративної взаємодії, у яких учасники формують спільний сюжет відповідно до визначеного набору правил [10]. Найбільш поширеною системою є «Dungeons & Dragons», яка поєднує механіки розвитку персонажів, бойових взаємодій, дослідження світу та рольової взаємодії [10]. При цьому ігровий процес спирається на майстра гри, який по суті є ведучим, що розповідає гравцям історію. Центральною фігурою гри є майстер, який керує перебігом пригоди на основі сценарію, що визначає структуру подій, конфлікти, персонажів і локації [9]. Якість сценарію безпосередньо впливає на цілісність ігрового процесу та рівень залучення гравців [10].

Пригоди можна поділити на два категорії: ваншоти та кампанії. Кампанія – це велика пригода, що розрахована на проходження протягом значного періоду часу. Тож для зручності в цій роботі використовується формат ваншотів, під яким розуміють завершений сценарій, розрахований на повне проходження протягом однієї ігрової сесії. На відміну від класичних довготривалих кампаній, такі односесійні пригоди характеризуються специфічними особливостями, серед яких варто виділити компактну сюжетну структуру з обмеженою кількістю паралельних ліній, швидке введення гравців у безпосередній конфлікт, а також високу інтенсивність ключових подій. З точки зору структурної організації, ваншоти спираються на лінійно-вузлову топологію проходження [10]. Зазначена лінійно-вузлова структура передбачає, що сценарій формалізується як послідовність дискретних сцен або сюжетних вузлів, між якими існують гнучкі альтернативні переходи, що дозволяє підтримувати загальну керованість історії та одночасно зберігати елемент свободи вибору для гравців [7].

У сучасному ігровому дизайні сценарій настільної рольової гри розглядається як композиція окремих функціональних компонентів, що призвело до формування концепції компонентного моделювання пригод [10]. За такого підходу сценарій формується як структурована комбінація окремих елементів – сюжетних зачіпок, вступних подій, досліджень локацій, соціальних взаємодій, бойових зіткнень, загадок та сюжетних поворотів, які логічно підводять гру до кульмінації та фінального завершення [4]. Використання подібної атомарної структури дозволяє забезпечити повторне використання окремих об'єктів бази даних, високу варіативність фінального контенту та підтримання загальної сюжетної цілісності [7].

Особливе значення у системах на кшталт «Dungeons & Dragons» мають бойові зіткнення, які виступають не лише елементом наративу, а й суворо формалізованою математичною механікою [10]. Відтак, задача формування бойових сцен трансформується у задачу фільтрації та відбору елементів із множини можливих супротивників відповідно до параметрів ігрової групи [10]. Крім бойових ситуацій, важливу роль відіграють соціальні взаємодії з неігровими персонажами для розвитку конфліктів та дослідницькі сцени, які сприяють формуванню атмосфери ігрового всесвіту [1].

Традиційні методи створення пригод базуються на ручному написанні сценаріїв або використанні випадкових таблиць, що супроводжується значними часовими витратами, складністю підтримання довгострокової сюжетної узгодженості, високою повторюваністю контенту, обмеженою адаптацією до запитів користувача та жорсткою залежністю від досвіду майстра [1], [7]. Для подолання цих обмежень у сучасних дослідженнях використовують методи штучного інтелекту та процедурної генерації контенту [7], [10]. Такі підходи дозволяють формувати сюжетні структури, підбирати тематично сумісні елементи сценарію та генерувати текстовий опис пригод відповідно до заданих параметрів.

Одним із перспективних напрямів є використання формалізованих моделей пригоди, які описують ключові характеристики майбутнього сценарію: жанр, тематику, тип локації, особливості конфлікту, антагоністів, цілі персонажів та інші структурні компоненти.

Додаткові можливості для автоматизації створення пригод забезпечують сучасні моделі обробки природної мови та великі мовні моделі. Їх використання дозволяє генерувати текстові описи сцен, персонажів і сюжетних подій на основі попередньо сформованої структури пригоди. Поєднання процедурної генерації, формалізованого представлення сценарію та мовних моделей створює передумови для побудови систем, здатних автоматично формувати завершені пригоди для настільних рольових ігор.

1.2 Аналіз існуючих підходів та рішень

1.2.1 Генератори на основі випадкового вибору та шаблонів

Одними з найбільш поширених допоміжних інструментів у практиці майстрів рольових ігор є генератори, що базуються на псевдовипадковому виборі текстових елементів із наперед підготовлених статичних масивів даних [7]. До відомих представників цієї групи належать платформи «Donjon RPG Tools» (рисунк 1.1) та «Kassoon RPG Generator». Подібні програмні рішення дозволяють в автоматичному режимі створювати ізольовані назви локацій, імена персонажів, описи таверн, карти підземель, найпростіші випадкові квести та фіксовані набори супротивників [10].

Головною перевагою використання таких генераторів є мінімальна складність їхньої програмної реалізації та висока швидкість отримання результату. Проте, оскільки обчислювальний процес спирається виключно на вибірку сутностей із зумовлених наборів даних за допомогою генераторів псевдовипадкових чисел, система повністю позбавлена можливості

аналізувати наративний контекст [7]. Унаслідок цього виникає низка критичних недоліків, таких як низька сюжетна узгодженість, слабкий причинно-наслідковий зв'язок між окремими сценами, висока повторюваність контенту, відсутність семантичного аналізу тексту та неможливість гнучкої адаптації системи під унікальний розгорнутий запит користувача [10]. Як наслідок, подібні алгоритми спроможні створювати лише фрагментарні компоненти, але не цілісний, логічно завершений сценарій пригоди. Таким чином, сфера застосування подібних генераторів обмежується створенням окремих допоміжних елементів пригоди, тоді як формування повноцінного сценарію потребує використання більш складних алгоритмів генерації та аналізу природної мови.

Random Adventure Generator	
Theme	Romance This sort of adventure (rarely played, but worthwhile anyway) has as its central plot the romance between two characters, usually a player-character and an NPC.
Goal	Retrieve Item This goal is like the Rescue, except the victim is an inanimate object. This item may be an artifact, a paper containing evidence against a character or patron, an antidote needed to save another NPC or player character, or just some item of sentimental value — an item which one NPC hires the characters to steal from another NPC.
Story Hook	Not in Kansas The PCs are minding their own business and find themselves transported to a strange place. They must figure out where they are, why they are there, and how to escape.
Plot	Series of Villains This is a very dramatic plot, and very well-suited to oriental campaigns. In it, the heroes have undertaken a quest, usually the finding and defeat of the Master Villain. They may have to travel to his citadel, or head off in another direction to find some artifact capable of defeating him, or run away from pursuing villains until they can figure out what's going on. All along their route, they are set upon by villains — each villain has a name and distinct personality, and each encounter is life-or-death for the heroes and villains; the villain never escapes to safety if the tide turns against him, he fights unto death.
Climax	Scattered Duels In this climax, the heroes have gotten to the end of their quest — they may have broken into, sneaked into, or escaped from imprisonment within the villain's citadel, or have marched into the little town where the villain is holed up — and they become separated. You can separate them by having traps and tricks break the party apart, by having them see two or three things they must resolve (such as danger to innocents or the appearance of minion villains) pop up simultaneously; they'll have to run in all directions at the same time or suffer failure. Once the party is broken down into bite-sized chunks, you confront each individual or small group with the enemy or enemies he most deserves to face — his personal enemy, the monster which defeated him before, etc. — for a grand series of climactic duels.
General Setting	Exotic Distant Land The adventure will take the heroes to some fascinating and exotic distant country, where they'll have to cope with new customs, monsters unfamiliar to them, and very colorful NPC encounters; choose one of the more fascinating foreign lands from your campaign world.

Рисунок 1.1 – Приклад пригоди згенерованої на платформі Donjon

1.2.2 Правилоорієнтовані системи генерації сюжетів

Одним із ранніх напрямів розвитку автоматизованої генерації сюжетів є правилоорієнтовані системи, які формують сценарії на основі заздалегідь

визначених логічних правил, обмежень та залежностей між подіями. На відміну від генераторів випадкового вибору, такі системи не лише обирають окремі елементи контенту, а й контролюють умови їх використання, забезпечуючи причинно-наслідкові зв'язки між сюжетними компонентами [7].

Основою правилоорієнтованих систем є продукційні правила, які задаються у вигляді конструкцій типу «якщо – то». Кожне правило визначає умови, за яких певна подія, персонаж або сюжетна ситуація можуть бути додані до сценарію. Завдяки цьому система здатна автоматично підтримувати внутрішню логіку розвитку пригоди та уникати суперечливих комбінацій сюжетних елементів.

Подальшим розвитком такого підходу стали експертні системи генерації квестів та автоматичного сторітелінгу, в яких знання про предметну область подаються у вигляді бази фактів і набору правил виведення. Подібні системи широко використовувалися для створення сценаріїв комп'ютерних рольових ігор, де необхідно забезпечити логічну послідовність подій та дотримання встановлених обмежень ігрового світу [10].

Окремим різновидом правилоорієнтованих підходів є сюжетні граматики, що розглядають історію як ієрархічну структуру взаємопов'язаних елементів. Згідно з цим підходом сценарій будується шляхом послідовного розгортання сюжетних шаблонів відповідно до визначених правил композиції. Така модель дозволяє формувати пригоди зі сталою драматургічною структурою, проте обмежує різноманітність можливих сюжетних комбінацій.

Для реалізації складніших сценаріїв також застосовуються скінченні автомати та системи планування сюжету.

Головною перевагою правилоорієнтованих систем є можливість жорсткого контролю логіки пригоди, підтримання причинно-наслідкових зв'язків між подіями та забезпечення узгодженості окремих сюжетних

компонентів. Водночас такі системи характеризуються високою трудомісткістю розробки, оскільки зі збільшенням кількості можливих сюжетних ситуацій суттєво зростає обсяг необхідних правил. Крім того, надмірна залежність від заздалегідь визначених умов обмежує творчість системи та її здатність адаптуватися до довільних запитів користувача природною мовою.

Незважаючи на зазначені обмеження, правилоорієнтовані підходи залишаються важливим компонентом сучасних систем генерації ігрового контенту, оскільки дозволяють реалізовувати механізми контролю сумісності сюжетних елементів, перевірки логічних обмежень та підтримання цілісності структури сценарію.

1.2.3 III-системи генерації тексту

Із розвитком архітектури Transformer та методів обробки природної мови широкого поширення набули III-системи генерації художнього тексту, засновані на великих мовних моделях (Large Language Models, LLMs) [9]. Провідними спеціалізованими платформами у цьому домені стали «AI Dungeon» та «NovelAI», а також універсальні моделі загального призначення класу ChatGPT. Використання нейромережових декодерів дозволяє створювати великі обсяги зв'язного тексту природною мовою, забезпечуючи високу якість художніх описів, стилістичну різноманітність, генерацію динамічних діалогів та адаптацію сюжету під контекст довільного запиту користувача [2].

Проте, спеціалізовані системи для рольових ігор, базованих на III, переважно мають вигляд чатів, в яких майстром виступає III-модель, якій гравець описує свої дії. Моделей, які використовуються саме для генерації сценарію, який може використовувати майстер немає або вони не розповсюджені і мають певні недоліки. Також для цього можна використовувати і класичні OpenAI моделі, але попри значні наративні

переваги, застосування чистих мовних моделей без зовнішнього детермінованого контролю для генерації сценаріїв за системою «Dungeons & Dragons» супроводжується серйозними обмеженнями [6]. Основною проблемою є схильність моделей до ефекту «галюцинацій», що виражається у створенні неіснуючих монстрів, вигадуванні заклинань чи здібностей та критичному порушенні балансу характеристик ігрових сутностей [4].

Іншою важливою вадою є математичний дисбаланс бойових сцен. За правилами D&D складність бойового зіткнення є функцією від багатьох параметрів, включаючи рівень персонажів, рівень супротивників, бюджет досвіду, кількість ворогів та економіку дій [10]. Оскільки чисті LLM оперують імовірнісними закономірностями розподілу токенів, а не суворими обчисленнями, вони генерують або летальні для ігрової групи зіткнення, або занадто тривіальні конфлікти [1], [2], що змушує майстра гри здійснювати тривале ручне редагування отриманого тексту.

1.2.4 Генеративно-пошукові системи та системи процедурної генерації контенту

Процедурна генерація контенту традиційно спирається на суворі математичні правила ігрової індустрії для автоматичного створення двовимірних або тривимірних просторів, ігрових подій та параметрів персонажів [7]. У контексті проєктування сценаріїв настільних рольових ігор методи PCG застосовуються переважно для алгоритмічної побудови топології підземель, автоматичного комбінування сюжетних блоків та формування випадкових подій під час дослідження світу.

Перевагою процедурного підходу є забезпечення високого рівня унікальності та варіативності контенту. Проте функціонування PCG-систем без інтегрованого контуру контекстного аналізу призводить до виникнення логічно суперечливих ігрових ситуацій, хаотичних сюжетних переходів, а

також до повної неузгодженості між геометрією локацій та супроводжуючим їх нарративним конфліктом [7], [10]. Через це чисті процедурні системи вимагають розробки додаткових складних механізмів семантичного контролю сюжетної цілісності [9].

Найбільш прогресивним напрямом розвитку сучасних NLP-систем у технічних та ігрових доменах є впровадження генеративно-пошукових архітектур [3], [20]. Вони поєднують у собі творчий потенціал трансформаторних моделей із детермінованою точністю пошуку інформації у структурованих базах даних. У межах RAG-архітектури первинний запит користувача проходить через retrieval-механізм семантичного пошуку на основі векторних представлень тексту – ембедінгів [5], [8].

Перетворення текстових описів компонентів майбутньої пригоди (таких як офіційно збалансовані монстри чи заздалегідь структуровані локації) у щільні вектори здійснюється неймережами, а вилучення найбільш релевантних сутностей відбувається за допомогою обчислення косинусної близькості [5].

Сформований у такий спосіб контекст інтегрується в інженерний запит, завдяки чому фінальний текстовий синтез мовною моделлю здійснюється з опорою на перевірені ігрові компоненти [2], [3]. Це дозволяє радикально знизити частоту логічних помилок і галюцинацій, забезпечити відповідність правилам ігрової системи та підтримувати структурну цілісність ваншоту за рахунок використання готових геймдизайнерських шаблонів [4], [9], [19].

1.2.5 Агентні системи

Також одним із перспективних напрямів розвитку сучасних систем автоматичної генерації сценаріїв є агентні архітектури (Agentic Narrative Systems), які набули значного поширення з розвитком великих мовних моделей та інструментів автономного

планування [12]. На відміну від традиційних генеративних систем, що виконують одноразову побудову тексту за єдиним запитом, агентні системи розглядають процес створення сценарію як послідовність взаємопов'язаних етапів, кожен з яких виконується окремим програмним агентом.

Доволі розповсюдженні в цій галузі багатоагентні системи генерації сюжетів, у яких різні агенти відповідають за окремі аспекти майбутньої історії [13]. Наприклад, один агент може формувати загальну структуру сюжету, інший – генерувати персонажів та їх мотивацію, третій – контролювати логічну узгодженість подій, а четвертий – перевіряти відповідність створеного контенту встановленим правилам предметної галузі. Обмін інформацією між агентами дозволяє поступово уточнювати сценарій та усувати суперечності, що виникають під час генерації.

Основною перевагою агентних підходів є можливість поєднання творчої гнучкості мовних моделей із механізмами планування, перевірки та самокорекції результатів. Завдяки цьому підвищується структурна узгодженість сценарію, покращується контроль за дотриманням заданих обмежень та зменшується кількість логічних помилок. Водночас такі системи характеризуються значною складністю архітектури, підвищеними обчислювальними витратами та потребують ретельної організації взаємодії між агентами.

Незважаючи на перспективність агентних архітектур, їх реалізація ускладнює програмну систему та збільшує витрати на генерацію, тому в межах даної роботи обрано гібридний підхід.

1.2.6 Порівняльний аналіз існуючих підходів та підсумки

Проведений аналіз існуючих підходів до автоматизованої генерації сюжетного контенту (таблиця 1.1) показав, що кожен із розглянутих методів має власні переваги та обмеження.

Таблиця 1.1 – Порівняльний аналіз підходів до автоматизованої генерації пригод

Технологічний підхід	Основні переваги	Основні недоліки
Генератори випадкового вибору	Простота реалізації; висока швидкість генерації; низькі обчислювальні витрати	Відсутність сюжетної узгодженості; висока повторюваність контенту; неможливість врахування складних користувацьких запитів
Правилоорієнтовані системи	Контроль логіки сценарію; підтримання причинно-наслідкових зв'язків; перевірка сумісності компонентів	Висока складність створення та підтримки правил; обмежена гнучкість; складність масштабування
Великі мовні моделі	Висока якість тексту; генерація описів, діалогів та сюжетів природною мовою; адаптація до довільних запитів користувача	Галюцинації; порушення ігрового балансу; недостатній контроль структури сценарію
Агентні системи генерації сюжетів	Багаторівневий контроль процесу генерації; підвищення узгодженості сценарію; можливість самоперевірки результатів	Складність архітектури; високі обчислювальні витрати; складність координації агентів
Генеративно-пошукові системи	Поєднання генерації тексту з перевіреними джерелами даних; зменшення кількості галюцинацій; підвищення достовірності контенту	Складність реалізації; залежність від якості бази знань та механізмів пошуку

Генератори випадкового вибору забезпечують високу швидкість створення контенту та простоту реалізації, проте не здатні підтримувати логічну цілісність сценарію. Правилоорієнтовані системи дозволяють контролювати причинно-наслідкові зв'язки між подіями та забезпечувати структурну узгодженість пригоди, однак характеризуються високою складністю підтримки великої кількості правил і обмеженою гнучкістю щодо довільних користувацьких запитів.

Великі мовні моделі забезпечують високу якість художнього тексту, здатність до інтерпретації природномовних запитів та генерації різноманітних сюжетів. Водночас використання чистих мовних моделей супроводжується ризиком виникнення галюцинацій, порушенням балансу

ігрових механік та недостатнім контролем структурної узгодженості сценарію. Агентні архітектури частково вирішують зазначені проблеми шляхом розподілу процесу генерації між декількома спеціалізованими агентами, проте потребують значних обчислювальних ресурсів та характеризуються підвищеною складністю реалізації.

Методи процедурної генерації контенту дозволяють автоматично формувати значну кількість унікальних об'єктів та ігрових ситуацій, однак без додаткових механізмів контролю часто породжують логічно неузгоджені результати. Генеративно-пошукові системи поєднують переваги мовних моделей із можливостями роботи зі структурованими джерелами даних, що дозволяє підвищити достовірність та якість сформованого контенту.

Аналіз також показав, що однією з ключових проблем автоматичної генерації пригод є забезпечення структурної та причинно-наслідкової узгодженості між окремими компонентами сценарію. Навіть за високої якості окремих текстових фрагментів система може формувати суперечливі сюжетні переходи, втрачати логіку розвитку конфлікту або створювати несумісні комбінації персонажів, локацій та подій. Тому сучасні системи генерації пригод дедалі частіше використовують компонентний підхід, за якого сценарій розглядається не як суцільний текст, а як сукупність взаємопов'язаних структурних елементів, що підбираються та узгоджуються відповідно до заданих обмежень.

Отже, результати проведеного аналізу свідчать, що найбільш перспективним підходом для розв'язання задачі генерації сценаріїв пригод у системі правил «Dungeons & Dragons» є використання гібридної архітектури, яка поєднує методи компонентного моделювання, процедурної генерації контенту, механізми контролю сумісності сюжетних елементів та сучасні мовні моделі. Такий підхід дозволяє забезпечити баланс між варіативністю контенту, структурною цілісністю сценарію та можливістю адаптації результату до довільного текстового запиту користувача. На основі

отриманих висновків у наступному розділі буде сформульовано вимоги до моделі генерації пригод та запропоновано архітектуру програмної системи.

1.3 Постановка задачі дослідження

Проведений аналіз існуючих підходів до автоматизованої генерації сюжетного контенту показав, що створення сценаріїв пригод для системи правил «Dungeons & Dragons» потребує комплексного підходу. Для отримання придатного до практичного використання результату необхідно забезпечити логічну узгодженість компонентів пригоди, причинно-наслідкові зв'язки між подіями та відповідність правилам предметної галузі.

Аналіз існуючих методів показав, що алгоритми процедурної генерації забезпечують різноманітність результатів, однак не гарантують їх структурної узгодженості, тоді як великі мовні моделі формують зв'язний текст, але не забезпечують достатнього контролю над логікою сюжету. Тому доцільним є поєднання структурованого моделювання пригод із сучасними методами обробки природної мови.

У межах даної роботи об'єктом дослідження є процес автоматизованої генерації сценаріїв пригод для настільних рольових ігор. Предметом дослідження виступають моделі, методи та алгоритми формування структурно узгоджених сценаріїв ваншотів у системі правил «Dungeons & Dragons».

Основною задачею дослідження є розробка моделі генерації пригоди, яка забезпечуватиме автоматичне створення завершеного сценарію на основі текстового запиту користувача. Така модель повинна виконувати аналіз побажань користувача, визначати основні характеристики майбутньої пригоди, здійснювати відбір сумісних сюжетних компонентів та формувати цілісну структуру сценарію перед етапом генерації текстового представлення.

Розроблена модель повинна забезпечувати структурну та сюжетну узгодженість пригоди, враховувати взаємозв'язки між її компонентами, підтримувати причинно-наслідкові зв'язки між сценами та відповідність правилам системи «Dungeons & Dragons». Сформований сценарій має бути придатним для практичного використання майстром гри без суттєвого ручного доопрацювання.

Таким чином, у роботі ставиться задача розробки моделі та програмної системи генерації сценаріїв пригод-ваншотів для системи правил «Dungeons & Dragons», яка забезпечуватиме баланс між варіативністю контенту, логічною цілісністю сценарію, відповідністю ігровим правилам та можливістю адаптації результату до індивідуальних побажань користувача.

2 РОЗРОБКА АЛГОРИТМІВ ПОБУДОВИ ПРИГОДИ

2.1 Концептуальна модель процесу генерації пригод

Однією з ключових проблем автоматизованої генерації сценаріїв настільних рольових ігор є забезпечення одночасно високої якості художнього тексту, логічної узгодженості сюжету та відповідності правилам ігрової системи. Найпростішим підходом до розв'язання цієї задачі є безпосередня генерація повного тексту пригоди за одним користувацьким запитом за допомогою великої мовної моделі. Однак на практиці такий підхід має низку суттєвих обмежень, які ускладнюють отримання стабільних та передбачуваних результатів.

Перш за все, під час генерації сценарію як єдиного текстового об'єкта виникають труднощі з контролем логічної цілісності пригоди. Повноцінний сценарій ваншоту складається з великої кількості взаємопов'язаних елементів: основного конфлікту, локацій, неігрових персонажів, фракцій, сюжетних подій, бойових зіткнень та фінального протистояння. У процесі створення великого обсягу тексту мовна модель може втрачати зв'язки між окремими елементами, що призводить до появи сюжетних суперечностей, нелогічних переходів між сценами або невідповідності між початковими передумовами та завершенням пригоди.

Додатковою проблемою є специфіка структури пригод формату ваншот. На відміну від довготривалих кампаній, які можуть розвиватися протягом багатьох ігрових сесій, ваншоти мають обмежений часовий ресурс і тому характеризуються відносно стабільною композиційною побудовою. У більшості випадків така пригода містить вступну подію, знайомство з конфліктом, декілька ключових сцен розвитку сюжету, кульмінацію та завершення історії. При генерації повного тексту без попереднього планування мовна модель не завжди дотримується подібної структури, що

може призводити до надмірного розширення окремих сюжетних ліній або, навпаки, до недостатнього розкриття важливих подій.

Ще одним важливим фактором є необхідність повторного використання окремих елементів пригоди. У процесі створення великої кількості сценаріїв доцільно використовувати заздалегідь підготовлені описи локацій, шаблони конфліктів, характеристики персонажів, монстрів та інші структурні компоненти. Генерація сценарію як суцільного тексту не дозволяє ефективно використовувати такі дані, оскільки кожна нова пригода створюється незалежно від попередніх результатів. Унаслідок цього знижується керованість процесу генерації та ускладнюється підтримка єдиного рівня якості контенту.

Особливе значення для системи «Dungeons & Dragons» має контроль відповідності сценарію правилам гри. На відміну від звичайного художнього тексту, пригода містить значну кількість формалізованих елементів, для яких існують визначені обмеження. До них належать рівні складності бойових зіткнень, характеристики супротивників, логіка побудови пригоди, типи доступних істот, фракцій та сюжетних ролей. У випадку прямої генерації тексту мовна модель не має гарантованих механізмів дотримання таких обмежень, що може призводити до створення некоректних або незбалансованих сценаріїв.

Для усунення зазначених недоліків у роботі використовується компонентний підхід до генерації пригод. Його сутність полягає в тому, що сценарій розглядається не як єдиний текстовий документ, а як сукупність окремих взаємопов'язаних компонентів, кожен із яких описує певний аспект майбутньої пригоди. До таких компонентів належать жанр, тематика, конфлікт, локації, персонажі, фракції, сюжетні сцени та бойові зіткнення.

Використання компонентної моделі дозволяє виконувати поетапне формування пригоди, контролювати узгодженість між її складовими, повторно використовувати структуровані елементи та здійснювати перевірку відповідності правилам «Dungeons & Dragons» ще до початку

генерації фінального тексту. У результаті мовна модель отримує не довільний запит, а вже підготовлену структуру пригоди, що суттєво підвищує передбачуваність та якість отриманого результату.

Таким чином, компонентний підхід виступає основою запропонованої моделі генерації сценаріїв та забезпечує можливість поєднання творчих можливостей великих мовних моделей із формалізованими механізмами контролю структури та логіки пригоди. Це дозволяє розглядати його як один із найбільш придатних підходів для автоматизованого створення ваншотів у системі правил «Dungeons & Dragons».

В основу розробленої моделі покладено компонентно-орієнтований підхід до генерації пригод, відповідно до якого фінальний сценарій розглядається не як єдиний текстовий документ, а як сукупність взаємопов'язаних структурних елементів. Такий підхід дозволяє розділити процес генерації на декілька послідовних етапів, кожен з яких відповідає за формування окремих характеристик майбутньої пригоди та забезпечує контроль її внутрішньої узгодженості.

Загальна архітектура системи складається з п'яти основних функціональних блоків: аналізу користувацького запиту, побудови формальної моделі пригоди, відбору сюжетних компонентів, формування структури сценарію та генерації фінального текстового представлення пригоди.

Робота системи починається з отримання текстового запиту користувача. На цьому етапі виконується аналіз природномовного опису майбутньої пригоди та визначаються її основні характеристики. Із тексту вилучаються відомості про бажану тематику пригоди, жанрові особливості, тип локації, стиль проходження, наявність соціальних взаємодій, загадок або бойових сцен. Результатом цього етапу є набір параметрів, які використовуються для подальшого формування структури сценарію.

Загальна послідовність роботи системи наведена на рисунку 2.1.

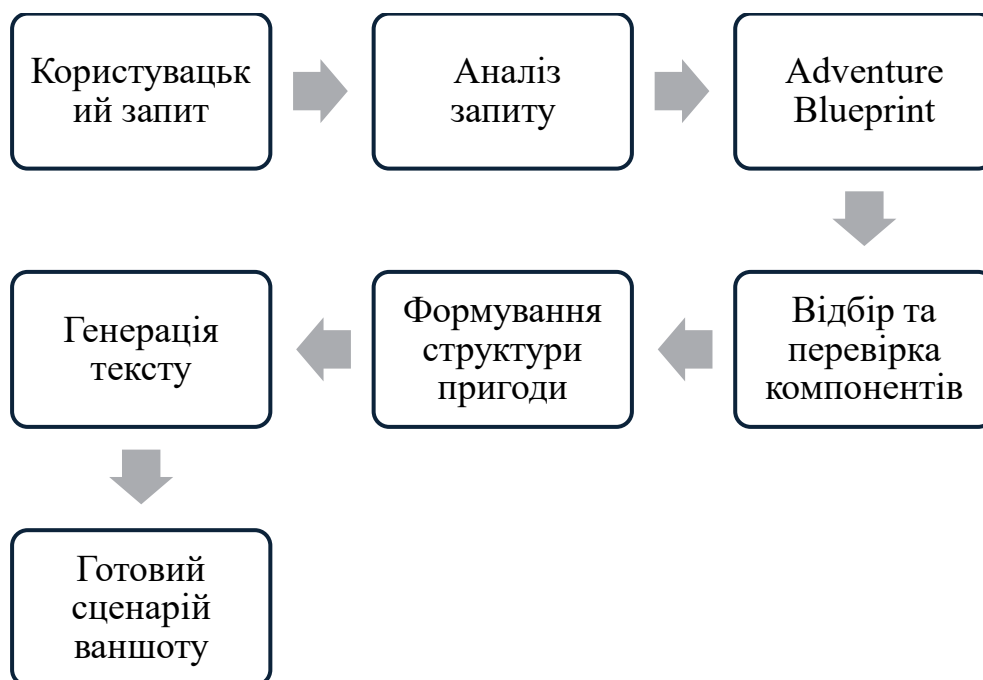


Рисунок 2.1 – Загальна послідовність роботи системи

На другому етапі здійснюється побудова формальної моделі пригоди. Дана модель виступає центральним елементом усієї архітектури та містить опис ключових характеристик майбутнього сценарію. До таких характеристик належать тема пригоди, тип конфлікту, цілі персонажів, тип локації, особливості антагоніста, склад доступних фракцій та інші параметри, необхідні для подальшого відбору компонентів. Формалізація пригоди у вигляді Blueprint дозволяє відокремити логіку побудови сценарію від процесу генерації тексту та забезпечити контроль узгодженості між окремими елементами.

Після побудови Blueprint система переходить до етапу відбору сюжетних компонентів. На основі характеристик пригоди виконується пошук та вибір відповідних елементів із структурованих баз даних. До таких елементів належать сюжетні зачіпки, неігрові персонажі, фракції, локації, дослідницькі сцени, соціальні взаємодії, загадки, бойові зіткнення та інші складові пригоди. Для запобігання появі логічно суперечливих комбінацій

використовується механізм перевірки сумісності компонентів, який відсікає невідповідні варіанти ще до етапу побудови сценарію.

Наступним етапом є формування структури пригоди. На цьому рівні відібрані компоненти об'єднуються у впорядковану послідовність сцен, які утворюють завершений сюжет. Формується вступна частина пригоди, розвиток основного конфлікту, ключові сюжетні події, кульмінація та завершення історії. Особлива увага приділяється підтриманню причинно-наслідкових зв'язків між окремими сценами та забезпеченню логічного розвитку сюжету.

На завершальному етапі сформована структура пригоди передається до мовної моделі, яка виконує генерацію текстового представлення сценарію. На основі отриманої структури створюються описи сцен, персонажів, локацій, сюжетних подій та інших елементів пригоди. Використання попередньо сформованої структури дозволяє забезпечити високу якість тексту при одночасному збереженні логічної цілісності сценарію та відповідності визначеним параметрам пригоди.

Запропонована архітектура поєднує переваги компонентного моделювання, процедурної генерації контенту, механізмів контролю сумісності та сучасних мовних моделей. Це дозволяє забезпечити баланс між варіативністю результатів, структурною узгодженістю сценарію та можливістю адаптації пригоди до довільних побажань користувача.

2.2 Вибір сюжетних компонентів

Після формування Adventure Blueprint система переходить до етапу відбору сюжетних компонентів, які використовуються для побудови майбутнього сценарію. Основною задачею даного етапу є автоматичний підбір структурних елементів пригоди відповідно до характеристик, визначених під час аналізу користувацького запиту. До таких елементів

належать антагоністи, локації, фракції, соціальні взаємодії, загадки, дослідницькі сцени та бойові зіткнення.

На відміну від традиційних генераторів випадкового вибору, відбір компонентів у розробленій системі виконується з урахуванням обмежень сумісності між елементами пригоди. Кожен компонент описується набором атрибутів, що характеризують його тематику, жанрову належність, тип середовища, рівень небезпеки та інші параметри. Adventure Blueprint виступає у ролі набору критеріїв, відповідно до яких здійснюється фільтрація доступних елементів бази даних.

Першим етапом відбору є визначення основного антагоніста пригоди. Антагоніст виступає центральним джерелом конфлікту та значною мірою визначає характер усієї історії. Для його вибору використовуються параметри тематики пригоди, типу конфлікту, рівня небезпеки та жанру сценарію. Наприклад, для пригод із містичною тематикою можуть обиратися культисти, некроманти або надприродні істоти, тоді як для політичних пригод перевага надається фракціям, корумпованим посадовцям або кримінальним угрупованням. Після визначення антагоніста його характеристики використовуються для подальшого відбору інших компонентів сценарію.

Наступним етапом є підбір локацій. Вибір середовища виконується відповідно до типу пригоди та характеристик антагоніста. Якщо основний конфлікт пов'язаний із діяльністю культу, система надає перевагу занедбаним храмам, катакомбам або таємним святилищам. Для пригод дослідницького типу можуть використовуватися руїни, стародавні підземелля чи ізольовані поселення. Таким чином забезпечується тематична відповідність між місцем подій та сюжетним конфліктом.

Після вибору локацій система виконує відбір фракцій та неігрових персонажів. Фракції використовуються для створення політичних, соціальних або військових взаємодій між учасниками пригоди. Для кожної фракції зберігається інформація про її цілі, ідеологію та можливі зв'язки з

іншими елементами сценарію. Під час генерації система відбирає лише ті фракції, які не суперечать визначеній тематиці пригоди та можуть бути логічно інтегровані у сформований конфлікт.

Особливе значення мають соціальні взаємодії, які використовуються для розвитку сюжету та передачі інформації гравцям. До таких взаємодій належать переговори, допити, дипломатичні зустрічі, отримання завдань, збирання чуток та інші форми спілкування з неігровими персонажами. Вибір конкретних соціальних сцен залежить від стилю пригоди та рівня соціальної складової, визначених у Blueprint.

Для пригод, що містять дослідницькі елементи, система додатково відбирає загадки та спеціальні випробування. Кожна загадка характеризується рівнем складності, тематикою та типом взаємодії з гравцями. Під час відбору враховується відповідність загадки поточній локації та загальній атмосфері сценарію. Наприклад, магичні головоломки можуть використовуватися в стародавніх руїнах, тоді як механічні загадки більше відповідають техномагічним або інженерним комплексам.

Окремим модулем реалізовано формування бойових сцен. Для кожного потенційного зіткнення система визначає допустимий набір супротивників відповідно до рівня складності пригоди та характеристик основного антагоніста. Відбір монстрів виконується із бази даних ігрових істот із урахуванням їхнього рівня небезпеки, типу середовища проживання та тематичної відповідності сценарію. Такий підхід дозволяє забезпечити баланс між наративною складовою пригоди та вимогами ігрової механіки.

Після завершення відбору всі компоненти проходять додаткову перевірку сумісності. На цьому етапі усуваються комбінації, які можуть призвести до логічних суперечностей або порушення цілісності сценарію. Лише після успішного проходження перевірки сформований набір компонентів передається до модуля побудови структури пригоди.

Запропонований механізм дозволяє розглядати сценарій як систему взаємопов'язаних елементів, що підбираються відповідно до формальної

моделі пригоди. Завдяки цьому забезпечується тематична узгодженість, логічна цілісність та висока варіативність сформованих ваншотів.

2.3 Основи семантичного пошуку та моделей обробки природної мови

Сучасні системи генерації текстового контенту активно використовують методи обробки природної мови (Natural Language Processing, NLP) та семантичного пошуку для аналізу текстових масивів, визначення вмісту користувацьких запитів та формування змістовно узгоджених результатів [9]. Особливого значення такі підходи набули у системах автоматичного синтезу текстів, рекомендаційних сервісах та генеративних архітектурах штучного інтелекту [3], [4]. У задачах автоматизованого проєктування сценаріїв для настільних рольових ігор NLP-моделі застосовуються для комплексного аналізу вхідних запитів користувача, ідентифікації жанрових і стилістичних ознак майбутньої пригоди, інтелектуального пошуку релевантних сюжетних компонентів у реляційних або векторних сховищах, а також для безпосередньої генерації зв'язного наративу із підтриманням тематичної узгодженості фінального сценарію ваншоту [2], [6].

Одним із ключових елементів сучасних лінгвістичних систем є семантичний пошук (semantic search) – підхід до знаходження інформації, заснований не на декларативному збігу окремих ключових слів чи лексем, а на аналізі глибинної змістовної близькості та контекстуальних зв'язків оброблюваних текстів [8].

Для технічної реалізації механізмів семантичного пошуку текстова інформація природної мови повинна бути трансформована у формалізовану структуру числових векторів фіксованої розмірності, які у комп'ютерних науках визначаються як щільні векторні представлення, або ембедінги (embeddings) [5], [8]. У класичних статистичних підходах до обробки природної мови текст подавався у вигляді розріджених векторів за

методологіями мішка слів (Bag-of-Words) або зважування частот TF-IDF [8]. Проте зазначені математичні інструменти виявляли критичну обмеженість, оскільки повністю ігнорували контекст розміщення слів, синтаксичну структуру речень та наявність синонімічних зв'язків у мові.

Сучасні NLP-моделі оперують багатовимірними неперервними векторними просторами, у яких кожному слову, реченню або цілісному текстовому фрагменту відповідає щільний числовий вектор високої розмірності [5]. Подібне представлення дозволяє з високою точністю визначати семантичну близькість довільних текстів, знаходити подібні за змістом фрази без прямих лексичних збігів, групувати схожі текстові елементи за тематичними кластерами та здійснювати ефективне вилучення даних (semantic retrieval) [3]. Головна закономірність функціонування векторного простору полягає в тому, що текстові об'єкти зі схожим семантичним навантаженням розташовуються математично ближче один до одного, тоді як вектори джерел із різною семантикою характеризуються значною відстанню між своїми координатами.

У сучасних прикладних архітектурах штучного інтелекту ембедінги формуються адаптивно для різних рівнів гранулярності тексту, включаючи окремі токени, закінчені речення, абзаци та повні документи [9], [18]. На відміну від згаданих класичних статистичних методів, нейромережеві ембедінги фіксують складні закономірності використання слів та латентні семантичні залежності між текстовими компонентами. Це забезпечує кардинальне підвищення якості семантичного пошуку, контекстного аналізу та механізмів semantic matching, що є критично важливим для retrieval-компонентів, які відповідають за семантичний відбір сумісних ігрових сутностей серед великих обсягів неструктурованих даних.

Однією з фундаментальних математичних задач у межах семантичного пошуку є кількісне оцінювання ступеня змістовної близькості між різними текстовими фрагментами, що реалізується за допомогою спеціальних метрик семантичної подібності (semantic similarity) [5, 8].

Зазначені метрики дозволяють обчислювально визначити, наскільки два довільні тексти близькі за своєю художньою чи логічною суттю, чи належать вони до єдиного тематичного кластера, та який рівень релевантності має знайдений у базі даних ігровий компонент відносно початкового текстового запиту користувача.

У сучасних NLP-системах обчислення семантичної подібності детерміновано зводиться до визначення геометричної відстані між відповідними embedding-векторами у багатовимірному просторі ознак [5]. Найбільш ефективною та поширеною в інженерії інформаційного пошуку метрикою для вимірювання близькості між такими векторами є косинусна подібність (cosine similarity), яка визначає косинус кута між двома спрямованими векторами у просторі 3D або вищих розмірностей [8].

Математично функція косинусної подібності формалізується таким виразом (2.1):

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}, \quad (2.1)$$

де A та B – embedding-вектори текстових описів;

$A \cdot B$ – скалярний добуток векторів;

$\|A\|$ та $\|B\|$ – довжини відповідних векторів.

Обчислене значення косинусної подібності строго належить неперервному інтервалу від -1 до 1. При цьому показники, що наближаються до 1, свідчать про максимальну семантичну тотожність текстів. Значення в коло 0 вказують на повну ортогональність векторів і відсутність будь-якого змістовного зв'язку між фразами, а від'ємні значення демонструють протилежність напрямків векторів у семантичному просторі. У межах розроблюваних retrieval-систем косинусна близькість слугує головним критерієм ранжування для відбору топ-N найбільш релевантних текстових елементів.

Базисом функціонування сучасних високоточних моделей обробки природної мови є нейромережева архітектура Transformer, яка спеціалізується на паралельній обробці послідовностей даних із урахуванням довгих дистрибутивних зв'язків [10], [13]. Головною відмінністю трансформаторних моделей від застарілих рекурентних нейронних мереж є математичний механізм уваги, що дозволяє динамічно обчислювати взаємну вагу та інформаційну значущість кожного слова в контексті всього аналізованого речення одночасно [9].

Базова функція механізму уваги (Scaled Dot-Product Attention) оперує концепцією трьох матриць – запитів (Query, Q), ключів (Key, K) та значень (Value, V), і формалізується таким чином (2.2):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.2)$$

де Q – матриця query-векторів;

K – матриця key-векторів;

V – матриця value-векторів;

d_k – розмірність key-векторів.

Завдяки механізму уваги трансформаторні архітектури забезпечують побудову контекстуалізованих ембедінгів (contextual embeddings) [5]. На відміну від статичних векторів першого покоління (таких як Word2Vec чи GloVe), де кожне слово мало лише один фіксований вектор незалежно від оточення, контекстуалізовані ембедінги адаптують числове представлення слова під його конкретну синтаксичну та наративну роль у поточному реченні. Це дозволяє диференціювати значення омонімів та полісемічних термінів, що є критично важливим у задачах генерації ігрових сценаріїв, де однакові сутності (наприклад, назви механік, типів монстрів чи елементів середовища) набувають зовсім різних значень залежно від специфіки сюжетного контексту ваншоту [9], [10].

Системи вилучення інформації (retrieval systems) у сучасній комп'ютерній інженерії відповідають за інтелектуальний відбір, фільтрацію та ранжування релевантних даних із великих сховищ неструктурованої інформації [8]. У межах передових AI-архітектур механізми пошуку інтегруються безпосередньо з генеративними мовними моделями, формуючи гібридний технологічний стек класу Retrieval-Augmented Generation (RAG) [3], [4]. Основними функціональними задачами пошукового компонента в таких системах є первинна фільтрація синтаксичного шуму, семантичне порівняння користувацьких намірів із базою знань та підготовка структурованого контексту для подальшої передачі на вхід мовної моделі.

Технічна реалізація сучасного інформаційного пошуку спирається на використання спеціалізованих векторних баз даних (vector databases), алгоритмів пошуку найближчих сусідів (nearest neighbor search) та метрик семантичної подібності [5], [8]. Поєднання retrieval-компонента з генеративними нейромережами дозволяє нівелювати базові недоліки чистих мовних моделей, зокрема суттєво знизити рівень нейромережевих галюцинацій, гарантувати точну відповідність згенерованого тексту зовнішнім правилам і обмеженням предметної галузі, контролювати композиційну структуру сценарію та підвищити загальну логічну цілісність результату [3], [4]. Впровадження таких архітектурних рішень у процеси розробки сценаріїв для настільних рольових ігор забезпечує надійний автоматизований відбір ігрових компонентів за критеріями балансу та тематики, усуваючи необхідність подальшої ручної корекції сформованого матеріалу [1], [9].

2.4 Формальна модель пригоди

Однією з основних проблем автоматизованої генерації пригод для настільних рольових ігор є забезпечення узгодженості між окремими

елементами сценарію. У разі безпосередньої генерації тексту мовною моделлю без попередньої структуризації виникає ризик появи логічних суперечностей, тематично несумісних компонентів та порушення загальної композиції пригоди. Для усунення зазначених недоліків у роботі запропоновано формальну модель пригоди, яка використовується як проміжне структуроване представлення майбутнього сценарію.

Основою розробленої моделі є концепція Adventure Blueprint – формалізованого опису ключових характеристик пригоди, які визначають подальший процес відбору компонентів та побудови сценарію. Blueprint виступає центральним елементом системи та виконує роль єдиного джерела інформації для всіх наступних етапів генерації.

Формально пригоду можна представити у вигляді множини (2.3):

$$B=\{T,G,L,C,A, S,P,E\}, \quad (2.3)$$

де T – тема пригоди (Theme);

G – жанр пригоди (Genre);

L – тип локації або середовища (Location);

C – основний конфлікт (Conflict);

A – антагоніст або джерело загрози (Antagonist);

S – соціальні взаємодії (Social Encounters);

P – загадки та дослідницькі випробування (Puzzles);

E – бойові сцени та зіткнення (Encounters).

На відміну від класичних шаблонних генераторів, запропонована модель не визначає жорстко фіксований сценарій пригоди. Натомість вона задає простір допустимих рішень, у межах якого система може обирати конкретні сюжетні компоненти. Такий підхід дозволяє одночасно забезпечити варіативність результатів та контроль їх логічної узгодженості.

Кожен елемент множини (B) характеризується власним набором атрибутів. Наприклад, антагоніст описується типом загрози, рівнем

небезпеки, тематичною належністю та можливими зв'язками з іншими компонентами пригоди. Локація характеризується середовищем проведення пригоди, атмосферою, доступними типами сцен та допустимими сюжетними подіями. Для фракцій визначаються їх цілі, мотивація та взаємовідносини з іншими учасниками конфлікту.

Між окремими елементами Blueprint існують логічні залежності, які визначають допустимість їх одночасного використання в межах одного сценарію. Такі залежності можна подати у вигляді орієнтованого графа (2.4):

$$M = (V, R), \quad (2.4)$$

де M – модель пригоди;

V – Vertices – вершини графа;

R – Relations – зв'язки між вершинами.

У запропонованій моделі центральне місце займає конфлікт (C), який визначає характер розвитку всієї пригоди. Саме на основі конфлікту виконується вибір антагоніста, сюжетних подій, можливих союзників та структури кульмінації. Відповідно, більшість зв'язків між компонентами формуються навколо конфлікту як основного рушія сюжетної динаміки.

З наведених залежностей випливає, що тема пригоди впливає на вибір конфлікту, конфлікт визначає тип антагоніста, антагоніст формує набір доступних фракцій, а сукупність антагоніста та локації визначає характер бойових сцен. Водночас тип локації впливає на можливість використання певних загадок, а фракції формують соціальну складову пригоди.

Залежності між основними компонентами пригоди подані на рисунку 2.2.

Для забезпечення структурної цілісності сценарію модель також містить параметри композиційної побудови пригоди. Формально сценарій можна подати як впорядковану послідовність сцен (2.5):

$$S=\{s_1, s_2, \dots, s_n\}, \quad (2.5)$$

де S – сценарій;

s_i – належить до одного з функціональних типів: вступ, дослідження, соціальна взаємодія, загадка, бойова сцена, кульмінація або завершення пригоди.



Рисунок 2.2 – Залежності між основними компонентами пригоди

Таким чином, Adventure Blueprint виконує роль формальної моделі майбутнього сценарію, яка забезпечує єдине представлення характеристик пригоди, дозволяє контролювати сумісність окремих компонентів та виступає основою для побудови логічно узгодженої структури ваншоту. Використання такої моделі дозволяє відокремити процес проектування пригоди від генерації тексту та створює підґрунтя для подальшої автоматизації побудови сюжетів у системі правил «Dungeons & Dragons».

2.5 Модель перевірки сумісності компонентів

Однією з ключових проблем автоматизованої генерації пригод є забезпечення логічної узгодженості між окремими компонентами сценарію. Навіть за наявності коректно сформованої структури пригоди та якісних баз

даних система може створювати суперечливі комбінації елементів, які порушують внутрішню логіку сценарію. Прикладами таких ситуацій є поєднання несумісних типів локацій та антагоністів, використання сюжетних подій, що суперечать визначеній тематиці пригоди, або включення сцен, які не відповідають обраній атмосфері та жанру.

Для усунення подібних суперечностей у розробленій системі використовується модель перевірки сумісності компонентів, реалізована у вигляді модуля логічної фільтрації «BadCombinationsFilter». Основною задачею цього модуля є автоматичне виявлення та усунення конфігурацій компонентів, які можуть призвести до порушення цілісності сценарію.

У межах запропонованої моделі кожен сюжетний компонент розглядається як окремий об'єкт множини, а для кожної пари компонентів визначається функція сумісності (2.6):

$$Compatibility(c_i, c_j) = \begin{cases} 1, & \text{якщо компоненти сумісні;} \\ 0, & \text{інакше.} \end{cases} \quad (2.6)$$

де c_i, c_j представляє конкретний елемент пригоди: локацію, антагоніста, фракцію, загадку, соціальну сцену або бойове зіткнення.

Значення функції дорівнює одиниці у випадку, якщо одночасне використання компонентів не створює логічних суперечностей у сценарії. Якщо ж така комбінація порушує встановлені правила генерації, функція повертає нуль і відповідна комбінація виключається з подальшого розгляду.

У загальному випадку система перевіряє не лише окремі пари компонентів, а й узгодженість сформованої пригоди загалом. Для цього вводиться функція глобальної сумісності сценарію

Для кожної пари компонентів визначається функція сумісності (2.7):

$$G(B) = \prod_{i < j} C(c_i, c_j), \quad (2.7)$$

де $G(B)$ – GlobalCompatibility(B) – функція глобальної оцінки сумісності пригоди;

B – поточна конфігурація пригоди (Adventure Blueprint разом із відібраними компонентами);

c_i та c_j – окремі компоненти сценарію (антагоніст, локація, фракція, загадка, соціальна сцена, бойова сцена тощо);

n – загальна кількість компонентів, що входять до пригоди;

$C(c_i, c_j)$ – Compatibility(c_i, c_j) – функція перевірки сумісності пари компонентів;

Якщо хоча б одна перевірка повертає нульове значення, уся конфігурація вважається некоректною та підлягає модифікації або повторному відбору компонентів [18].

Основу роботи моделі складає набір логічних правил, що описують допустимі та заборонені поєднання елементів пригоди. Такі правила можна подати у вигляді предикатів (2.8):

$$Rule_k(c_i, c_j) \rightarrow \{True, False\}, \quad (2.8)$$

де $Rule_k$ – k -те правило перевірки логічної узгодженості;

k – номер або ідентифікатор правила з множини правил системи;

c_i та c_j – компоненти пригоди, для яких виконується перевірка;

$True$ – результат перевірки, що означає виконання відповідного правила та відсутність суперечностей між компонентами;

$False$ – результат перевірки, що свідчить про порушення встановленого обмеження або несумісність компонентів;

До основних типів обмежень належать:

- тематичні обмеження;
- жанрові обмеження;
- обмеження середовища;

- сюжетні обмеження;
- обмеження складності пригоди;
- обмеження логічної послідовності сцен.

Тематичні обмеження забезпечують відповідність компонентів обраний темі пригоди. Наприклад, сценарій із містичною тематикою не повинен містити елементи, характерні для гумористичних або технологічних пригод. Аналогічно пригода, побудована навколо політичних інтриг, потребує наявності соціальних взаємодій та фракцій, тоді як надлишкова кількість випадкових бойових сцен може порушити її композицію.

Жанрові обмеження використовуються для підтримання єдиного стилю пригоди. Детективний сценарій потребує наявності підказок, розслідування та збору інформації, тоді як пригода жанру survival повинна містити елементи виживання, обмежених ресурсів та небезпечного середовища.

Обмеження середовища перевіряють відповідність компонентів типу локації. Наприклад, морські істоти не повинні з'являтися у гірських фортецях, а підземні монстри – у міських соціальних пригодах без належного сюжетного обґрунтування.

Окрему категорію становлять сюжетні обмеження, що контролюють взаємозв'язок між антагоністом, конфліктом та кульмінацією пригоди. Основна загроза повинна бути безпосередньо пов'язана з фінальними подіями сценарію, а ключові сюжетні елементи мають підтримувати єдину лінію розвитку конфлікту.

Важливим елементом моделі є перевірка структурної узгодженості сценарію. Для цього кожна сцена пригоди розглядається як вершина орієнтованого графа (2.9):

$$G = (V, E), \quad (2.9)$$

де V – множина сцен;

E – множина допустимих переходів між ними.

Модуль перевіряє відсутність суперечливих переходів, логічних розривів та сцен, які не мають сюжетного зв'язку з основним конфліктом пригоди. Завдяки цьому забезпечується цілісність наративної структури та послідовність розвитку подій.

Таким чином, модель перевірки сумісності компонентів виконує роль механізму логічної валідації сценарію. Її використання дозволяє автоматично виявляти та усувати суперечливі комбінації елементів, підтримувати тематичну й структурну узгодженість пригоди та забезпечувати відповідність сформованого сценарію початковим параметрам Adventure Blueprint. Це суттєво підвищує якість фінального результату та зменшує потребу в ручному редагуванні згенерованих ваншотів.

2.6 Формування Adventure Blueprint на основі користувацького запиту

Першим етапом роботи запропонованої системи є аналіз текстового запиту користувача та автоматичне формування базової моделі пригоди (рисунок 2.3). Саме цей етап визначає загальний напрямок генерації та створює набір параметрів, який надалі використовується під час відбору сюжетних компонентів, побудови структури сценарію та формування текстового опису пригоди.

На відміну від підходу прямої генерації тексту за допомогою мовної моделі, де користувацький запит безпосередньо передається до LLM, у запропонованій системі він спочатку проходить етап структурного аналізу. У результаті цього аналізу не створюється готовий текст пригоди, а формується її формальна модель, яка описує ключові характеристики майбутнього сценарію.



Рисунок 2.3 – Схема утворення Blueprint

Основним джерелом інформації виступає довільний текстовий опис, введений користувачем. Запит може містити інформацію про місце подій, бажаний тип пригоди, особливості сюжету, атмосферу або окремі сюжетні елементи. Наприклад, запит: «Дослідження небезпечного лісу, де зникають мандрівники» містить відомості про тип локації, тематику пригоди та основний конфлікт.

На основі аналізу запиту система визначає набір характеристик, необхідних для побудови Adventure Blueprint. До них належать:

- тема пригоди;
- жанр;
- тип локації;
- тип пригоди;
- загальний настрій (тональність);
- очікувані сюжетні елементи.

Кожен параметр Blueprint визначає обмеження для наступних етапів генерації. Наприклад, обраний тип локації використовується під час відбору відповідних локацій, монстрів, екологічних особливостей та природних

небезпек. Жанр впливає на вибір сюжетних архетипів, а тональність визначає характер описів, соціальних взаємодій і фінальної розв'язки.

Сформований Adventure Blueprint не містить конкретних сюжетних компонентів або готового сценарію. Його призначення полягає у створенні узгодженого набору параметрів, який використовується всіма наступними модулями системи. Таким чином забезпечується єдина концепція пригоди незалежно від кількості використаних наборів даних.

Запропонований підхід дозволяє відокремити аналіз користувацького запиту від безпосередньої генерації тексту. Завдяки цьому мовна модель працює не з необробленим природномовним описом, а зі структурованою моделлю пригоди, що істотно зменшує кількість сюжетних суперечностей, покращує тематичну узгодженість сценарію та забезпечує можливість контролю всіх етапів його побудови.

2.7 Алгоритм генерації текстового сценарію

2.7.1 Формування контексту генерації

Після завершення етапів побудови моделі пригоди, відбору сюжетних компонентів та формування послідовності сцен система переходить до підготовки контексту генерації. Метою даного етапу є об'єднання всієї структурованої інформації про майбутню пригоду в єдине представлення, яке надалі використовується для створення текстового сценарію.

На попередніх етапах роботи формується набір взаємопов'язаних даних, що характеризують пригоду з різних аспектів. Для подальшої генерації вони об'єднуються у контекст генерації (2.10):

$$K=\{B,C,S\}, \quad (2.10)$$

де B – формальна модель пригоди (Adventure Blueprint);

C – множина відібраних сюжетних компонентів;

S – послідовність сюжетних сцен.

Формальна модель пригоди (B) містить основні характеристики майбутнього сценарію, зокрема тематику, жанр, тип локації, центральний конфлікт, антагоніста та інші параметри, що визначають загальну концепцію пригоди. Вона виступає джерелом високорівневих обмежень, яких необхідно дотримуватися під час подальшої генерації.

Множина компонентів (C) включає конкретні елементи пригоди, відібрані відповідно до параметрів Blueprint та перевірені на сумісність. До таких компонентів можуть належати локації, неігрові персонажі, фракції, бойові зіткнення, загадки та інші сюжетні об'єкти. Саме вони формують змістове наповнення майбутнього сценарію.

Послідовність сцен (S) визначає композиційну структуру пригоди та описує порядок розвитку сюжетних подій. Вона забезпечує логічний зв'язок між окремими компонентами та задає загальну динаміку розвитку історії від зав'язки до завершення.

У результаті формується цілісний контекст генерації, який містить як загальні характеристики пригоди, так і конкретні сюжетні елементи та їх взаємозв'язки. Використання такого підходу дозволяє зберегти структурну цілісність сценарію та забезпечити узгодженість між усіма його складовими на наступних етапах генерації.

2.7.2 Побудова інструкції генерації

Після формування контексту генерації виконується побудова інструкції генерації, яка використовується для подальшого створення текстового представлення пригоди. Основною метою цього етапу є перетворення структурованої моделі сценарію у форму, придатну для інтерпретації мовною моделлю.

На відміну від традиційного підходу, коли генерація виконується на основі короткого користувацького запиту, запропонована модель використовує розширену структуровану інструкцію, сформовану на основі параметрів пригоди та відібраних компонентів. Такий підхід дозволяє забезпечити відповідність результату заданій тематиці та підтримувати логічну цілісність сюжетної структури.

Джерелом даних для побудови інструкції виступає сформований контекст генерації K , який містить модель пригоди, набір сюжетних компонентів та послідовність сцен. На його основі створюється узагальнений опис майбутнього сценарію, що включає:

- жанр і тематику пригоди;
- опис основного конфлікту;
- інформацію про антагоніста та його цілі;
- перелік ключових неігрових персонажів;
- опис основних локацій;
- сюжетні події та їх послідовність;
- особливості бойових та соціальних взаємодій;
- вимоги до структури фінального сценарію.

Отримана інструкція задає межі та напрямок подальшої генерації, визначаючи змістовне наповнення майбутнього тексту. Фактично вона виконує роль проміжного представлення між формальною моделлю пригоди та її текстовим описом.

Використання структурованої інструкції дозволяє зменшити вплив випадкових факторів під час генерації та забезпечити збереження ключових характеристик пригоди. Завдяки цьому фінальний сценарій залишається узгодженим із параметрами, визначеними на попередніх етапах побудови моделі, а також підтримує логічні зв'язки між окремими сюжетними сценами та компонентами пригоди.

Формально процес побудови інструкції генерації можна подати у вигляді відображення (2.11):

$$P = f(K), \quad (2.11)$$

де K – контекст генерації;

P – сформована інструкція генерації;

f – процедура перетворення структурованих даних у текстове представлення.

Використання детально сформованої інструкції дозволяє ефективніше спрямовувати процес генерації тексту трансформерною мовною моделлю, забезпечуючи кращу відповідність результату заданому контексту та вимогам до структури сценарію [13].

У результаті виконання даного етапу формується цілісна інструкція генерації, яка використовується мовною моделлю для створення фінального текстового сценарію пригоди.

2.7.3 Генерація текстового представлення сценарію

Після формування контексту генерації та побудови інструкції генерації виконується створення текстового представлення пригоди. На цьому етапі структурована модель сценарію перетворюється на цілісний текст природною мовою, придатний для безпосереднього використання майстром гри.

Вхідними даними для генерації виступає інструкція генерації (P), сформована на основі контексту пригоди. Вона містить опис основного конфлікту, характеристику ключових персонажів, відомості про локації, послідовність сюжетних подій та інші структурні елементи сценарію. Отримавши таку інформацію, мовна модель виконує інтерпретацію заданої структури та формує її текстове представлення.

На відміну від генерації довільного тексту, у запропонованій моделі мовна модель не створює сюжет з нуля. Її основною функцією є деталізація

вже побудованої пригоди шляхом розширення структурованих компонентів до рівня повноцінного сценарію. Зокрема, модель виконує:

- генерацію описів локацій та ігрового оточення;
- деталізацію сюжетних подій;
- формування характеристик неігрових персонажів;
- створення діалогів та соціальних взаємодій;
- опис бойових та дослідницьких сцен;
- формування вступу, кульмінації та завершення пригоди.

У процесі генерації окремі сюжетні сцени перетворюються на зв'язні текстові фрагменти, які доповнюються художніми описами та поясненнями. Завдяки цьому сценарій набуває цілісної наративної структури та стає придатним для проведення ігрової сесії.

Результатом роботи алгоритму є завершений сценарій пригоди, що містить опис передумов конфлікту, послідовність сюжетних сцен, характеристики ключових персонажів, описи локацій та рекомендації щодо проведення пригоди. Використання структурованого контексту на попередніх етапах дозволяє забезпечити узгодженість між окремими частинами сценарію та зберегти відповідність початковим параметрам пригоди.

2.6.4 Забезпечення узгодженості результату

Однією з основних проблем використання великих мовних моделей для автоматичної генерації пригод є відсутність гарантій логічної та структурної узгодженості отриманого результату. Під час прямої генерації сценарію виключно на основі користувацького запиту мовна модель самостійно формує сюжет, персонажів, локації та послідовність подій. Такий підхід забезпечує високу творчість і різноманітність тексту, проте часто призводить до виникнення сюжетних суперечностей, порушення

причинно-наслідкових зв'язків та появи елементів, що не відповідають встановленій тематиці пригоди або правилам ігрової системи.

У запропонованій моделі генерація текстового сценарію не виконується безпосередньо з користувацького запиту. Перед зверненням до мовної моделі здійснюється побудова формальної моделі пригоди, відбір сюжетних компонентів та перевірка їх логічної сумісності. У результаті формується структурований контекст генерації, який містить усі ключові характеристики майбутньої пригоди.

Формально процес генерації можна подати у вигляді послідовності перетворень (2.12):

$$U \rightarrow B \rightarrow C \rightarrow S \rightarrow K \rightarrow P \rightarrow T, \quad (2.12)$$

де U – початковий запит користувача;

B – Adventure Blueprint;

C – множина відібраних компонентів пригоди;

S – сформована структура сценарію;

K – контекст генерації;

P – інструкція генерації;

T – фінальний текст пригоди.

На відміну від прямої генерації запропонований підхід містить декілька проміжних етапів контролю, кожен з яких виконує окрему функцію перевірки та узгодження даних. Завдяки цьому мовна модель працює не з довільним текстовим описом, а з попередньо підготовленою структурою пригоди.

Важливу роль у забезпеченні узгодженості результату відіграє система перевірки сумісності компонентів. Перед включенням до сценарію кожен сюжетний елемент проходить перевірку відповідності тематиці пригоди, типу локації, жанру, антагоністу та іншим параметрам Adventure Blueprint.

Компоненти, що порушують встановлені правила, відхиляються ще до початку генерації тексту.

Використання структурованої моделі пригоди дозволяє суттєво зменшити кількість галюцинацій мовної моделі. Оскільки ключові сюжетні елементи визначаються до етапу генерації, модель не створює їх самостійно, а лише деталізує вже сформовану структуру. Це знижує ймовірність появи неузгоджених персонажів, суперечливих сюжетних подій або об'єктів, що не відповідають загальній концепції пригоди.

Додатковою перевагою запропонованого підходу є підтримання відповідності правилам «Dungeons & Dragons». Підбір монстрів, формування бойових сцен, вибір локацій та інших сюжетних компонентів виконується до етапу текстової генерації з урахуванням встановлених обмежень предметної галузі. Завдяки цьому фінальний сценарій не лише зберігає наративну цілісність, але й залишається узгодженим із механіками ігрової системи.

Таким чином, використання Adventure Blueprint, механізмів перевірки сумісності компонентів та структурованої сюжетної моделі дозволяє перетворити мовну модель з інструмента створення історії «з нуля» на засіб деталізації попередньо побудованого сценарію. Це забезпечує підвищення логічної узгодженості пригоди, зменшення кількості галюцинацій та покращення відповідності фінального результату вимогам системи правил «Dungeons & Dragons».

2.7 Алгоритм формування бойових сцен

Бойові сцени є одним із ключових компонентів пригод у системі правил «Dungeons & Dragons», оскільки вони поєднують наративну складову сценарію з формалізованими механіками ігрової системи. На відміну від більшості інших сюжетних елементів, формування бойового зіткнення вимагає одночасного врахування тематичної відповідності

пригоді, структурної ролі сцени у розвитку сюжету та математичного балансу між учасниками конфлікту.

У запропонованій моделі генерація бойової сцени виконується після формування Adventure Blueprint та визначення основних характеристик пригоди. На цьому етапі система використовує параметри тематики, типу локації, жанру пригоди, типу антагоніста та очікуваного рівня складності для відбору потенційних супротивників із бази даних монстрів.

Формально множину доступних монстрів можна подати у вигляді (2.13):

$$M = \{m_1, m_2, \dots, m_n\}, \quad (2.13)$$

де m_i – окремий монстр, що характеризується набором атрибутів (2.14):

$$m_i = \{CR, T, E, XP\}, \quad (2.14)$$

де CR – показник складності монстра (Challenge Rating);

T – Type – тип істоти;

E – Environment – середовище існування;

XP – кількість досвіду за перемогу над монстром [15].

На першому етапі система виконує тематичну фільтрацію кандидатів. Для кожного монстра перевіряється відповідність його типу та середовища обраній тематиці пригоди, характеристикам локації та образу головного антагоніста. Наприклад, для пригоди в занедбаному некрополі перевага надається нежиті, культистам або демонічним істотам, тоді як використання морських чудовиськ або лісових створінь вважається небажаним.

Після тематичного відбору виконується оцінювання складності бойового зіткнення. У системі «Dungeons & Dragons» складність бою визначається співвідношенням між силою групи гравців та сукупною

небезпекою супротивників [12]. Для цього використовується бюджет досвіду зіткнення (2.15):

$$EncounterXP = \sum_{i=1}^k XP(m_i), \quad (2.15)$$

де k – кількість супротивників у сцені;

$XP(m_i)$ – досвід, що відповідає обраному монстру.

Отримане значення порівнюється з допустимим діапазоном складності для групи гравців відповідного рівня. Якщо сукупна складність виходить за межі встановлених обмежень, система виконує повторний відбір монстрів або коригує їх кількість.

Додатково враховується сюжетна роль бойової сцени. Проміжні сутички використовуються для підтримання динаміки пригоди та поступового нарощування напруги, тоді як фінальне зіткнення виконує функцію кульмінації сценарію. У фінальному бою один або декілька супротивників безпосередньо пов'язуються з головним антагоністом пригоди, що забезпечує логічне завершення основного конфлікту.

Результатом роботи алгоритму є множина бойових сцен (2.16):

$$Combat = \{e_1, e_2, \dots, e_k\}, \quad (2.16)$$

де кожна сцена e_i містить опис місця події, склад супротивників, сюжетний контекст зіткнення та його роль у структурі пригоди.

Таким чином, запропонований алгоритм поєднує механізми тематичної фільтрації, перевірки структурної узгодженості та балансування складності бойових сцен. Це дозволяє забезпечити відповідність згенерованих зіткнень правилам «Dungeons & Dragons» та підтримувати їх логічний зв'язок із загальною сюжетною структурою пригоди.

2.8 Модель адаптивного оцінювання сюжетних компонентів

Після завершення генерації пригоди взаємодія користувача із системою не припиняється. Для підвищення якості наступних генерацій пропонується використання механізму накопичення користувацького зворотного зв'язку, який дозволяє поступово адаптувати процес відбору сюжетних компонентів без зміни самої мовної моделі.

На відміну від класичних генеративних систем, де кожна нова генерація виконується незалежно від попередніх результатів, у запропонованій моделі інформація про оцінку сформованої пригоди використовується для коригування пріоритетів окремих компонентів бази знань. Таким чином, система поступово накопичує статистику успішності використання різних сюжетних елементів і враховує її під час формування наступних пригод.

Під час побудови нової пригоди накопичений рейтинг використовується як один із критеріїв ранжування кандидатів після завершення процедур семантичного пошуку та перевірки логічної сумісності компонентів. Якщо декілька сюжетних елементів однаково відповідають параметрам Adventure Blueprint і не порушують встановлених обмежень, система надає перевагу компонентам, що мають вищу середню оцінку користувачів. Таким чином, адаптація впливає лише на вибір між допустимими альтернативами та не змінює загальної структури пригоди, визначеної попередніми етапами генерації. Подібний підхід відповідає принципам використання механізмів зворотного зв'язку для покращення якості інформаційних систем без зміни базової моделі генерації [3, 4].

Основною метою адаптивного механізму є підвищення релевантності сценаріїв щодо очікувань користувачів при одночасному збереженні структурної цілісності пригоди, яка забезпечується попередніми етапами генерації.

Після завершення пригоди користувач оцінює результат за визначеними критеріями, наприклад:

- відповідність початковому запиту;
- логічність розвитку сюжету;
- цікавість пригоди;
- загальна якість сформованого сценарію.

Отримані оцінки пов'язуються з компонентами, які використовувалися під час побудови пригоди. Для кожного сюжетного компонента формується накопичуваний рейтинг, який відображає його ефективність у попередніх генераціях.

Для кожного окремого компонента обчислюється середній рейтинг (2.17):

$$Score(c_i) = \frac{1}{m} \sum_{k=1}^m Rating_k(c_i), \quad (2.17)$$

де $Score(c_i)$ – інтегральна оцінка компонента;

$Rating_k(c_i)$ – оцінка, отримана компонентом під час k -ї генерації;

m – кількість пригод, у яких використовувався відповідний компонент.

Отримане значення використовується як додатковий критерій під час подальшого відбору компонентів. За інших однакових умов система віддає перевагу елементам із вищим накопиченим рейтингом, оскільки вони демонстрували кращі результати у попередніх генераціях.

На кожній ітерації система накопичує нові оцінки користувачів, оновлює рейтинги сюжетних компонентів та використовує їх під час наступних генерацій. Завдяки цьому релевантність сформованих пригод поступово підвищується, тоді як структурна узгодженість зберігається завдяки використанню формальної моделі пригоди та системи перевірки логічних обмежень.

Для накопичення інформації про оцінювання система зберігає історію використання сюжетних компонентів разом із відповідними

користувацькими оцінками. Для кожного компонента фіксується кількість його використань, середній рейтинг та дата останнього оновлення оцінки. Такий підхід дозволяє аналізувати ефективність окремих елементів бази знань у довгостроковій перспективі та використовувати накопичену статистику під час формування нових пригод.

Оскільки середній рейтинг є статистичною характеристикою, його значення поступово стабілізується зі збільшенням кількості використань компонента. Це дозволяє зменшити вплив випадкових оцінок окремих користувачів і формувати більш об'єктивну оцінку якості сюжетних елементів. У результаті процес адаптації ґрунтується не на одиничних відгуках, а на накопиченому досвіді використання компонентів у багатьох згенерованих пригодах.

Запропонований механізм адаптивного оцінювання має низку практичних переваг порівняно з донавчанням великих мовних моделей. По-перше, він не потребує значних обчислювальних ресурсів і може застосовуватися без повторного навчання моделі після кожного оновлення бази знань. По-друге, накопичені оцінки можуть використовуватися незалежно від конкретної мовної моделі, що забезпечує можливість її заміни без втрати отриманого досвіду експлуатації системи. Крім того, відокремлення механізму адаптації від генератора тексту забезпечує модульність архітектури програмного засобу та полегшує його подальший розвиток і масштабування відповідно до сучасних підходів побудови інтелектуальних систем [10].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Реалізація програмного засобу

Для перевірки працездатності запропонованої моделі було реалізовано програмний засіб автоматизованої генерації сценаріїв пригод-ваншотів для системи правил «Dungeons & Dragons». Основною метою програмної реалізації є автоматизація повного циклу побудови пригоди – від аналізу текстового запиту користувача до формування завершеного сценарію природною мовою.

Архітектура програмного засобу (рисунок 3.1) побудована відповідно до розробленої у попередніх розділах моделі генерації.

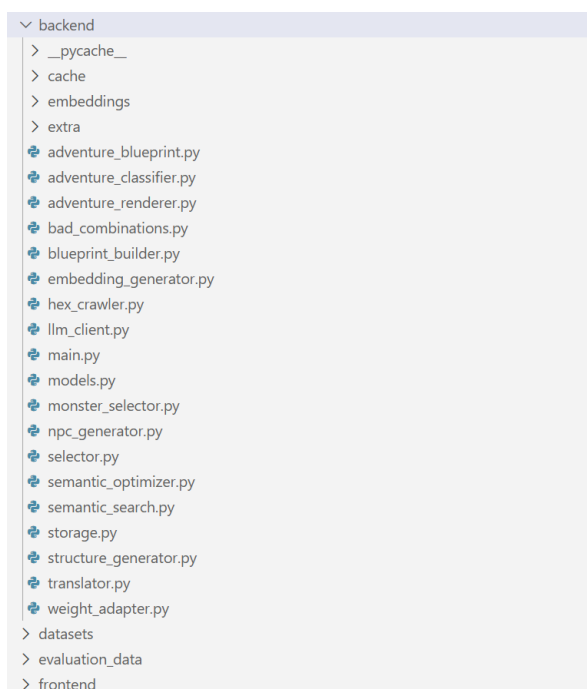


Рисунок 3.1 – Структура програмного проєкту

Робота системи починається з отримання користувацького запиту, у якому описується бажана тематика пригоди, її атмосфера, жанрові

особливості або інші побажання щодо майбутнього сценарію. Отриманий текстовий запит надходить до модуля аналізу природної мови, який виконує визначення ключових характеристик пригоди та формує формальну модель Adventure Blueprint.

На основі сформованого Blueprint система здійснює підбір сюжетних компонентів із внутрішньої бази даних. Для цього використовуються спеціалізовані набори даних, що містять інформацію про локації, антагоністів, фракції, сюжетні конфлікти, загадки, соціальні взаємодії, архетипи пригод, структури підземель та монстрів. Кожен компонент має набір атрибутів, які використовуються для оцінювання його відповідності параметрам майбутньої пригоди.

Після первинного відбору компонентів виконується перевірка їх логічної сумісності за допомогою модуля BadCombinationsFilter. Даний модуль реалізує набір правил предметної галузі, які дозволяють виявляти та усувати суперечливі поєднання елементів сценарію. Завдяки цьому забезпечується узгодженість між тематикою пригоди, характеристиками локацій, мотивацією антагоністів, фракціями та іншими сюжетними складовими.

Наступним етапом є побудова структури пригоди. На основі Blueprint та відібраних компонентів формується послідовність сюжетних сцен, що визначає розвиток пригоди від вступу до кульмінації та завершення. Додатково система може генерувати структуру підземель із використанням архетипів зон та кімнат, що дозволяє створювати пригоди дослідницького типу.

Особливу роль у системі відіграє модуль формування бойових сцен. Під час його роботи здійснюється відбір монстрів із урахуванням рівня персонажів, тематичних особливостей пригоди та характеристик антагоніста. Це дозволяє забезпечити відповідність бойових зіткнень загальній концепції сценарію та підтримувати баланс складності відповідно до правил «Dungeons & Dragons».

Після завершення формування структури пригоди всі отримані дані об'єднуються у контекст генерації. На його основі створюється структурована інструкція для мовної моделі, яка використовується для формування текстового представлення сценарію. На відміну від традиційної генерації тексту за коротким запитом, мовна модель працює з уже сформованою структурою пригоди та виконує її деталізацію шляхом створення описів локацій, персонажів, сюжетних подій, соціальних взаємодій та бойових сцен. У таблиці 4.1. наведено основні модулі програмного засобу, які приймають участь у цьому процесі.

Таблиця 3.1 – Основні модулі програмного засобу

Модуль	Призначення
Prompt Analyzer	Аналіз запиту користувача
Adventure Blueprint Builder	Формування моделі пригоди
Component Selector	Відбір сюжетних компонентів
BadCombinationsFilter	Перевірка узгодженості
Story Generator	Побудова структури пригоди
Encounter Generator	Формування бойових сцен
LLM Generator	Створення тексту пригоди
Web Interface	Взаємодія з користувачем

Програмний засіб реалізовано мовою Python, що обумовлено широкими можливостями цієї мови для обробки текстових даних, роботи з JSON-документами та інтеграції сучасних моделей штучного інтелекту. Для організації даних використовуються структуровані JSON-файли, які містять опис сюжетних компонентів та елементів пригод. Взаємодія з мовною

моделлю здійснюється через API, що забезпечує автоматичне формування фінального сценарію на основі підготовленого контексту.

Для забезпечення зручності використання реалізовано веб-інтерфейс користувача (рисунок 3.2), який дозволяє вводити параметри майбутньої пригоди та переглядати результати генерації. Інтерфейс забезпечує взаємодію між користувачем і серверною частиною системи, що виконує всі етапи побудови сценарію.

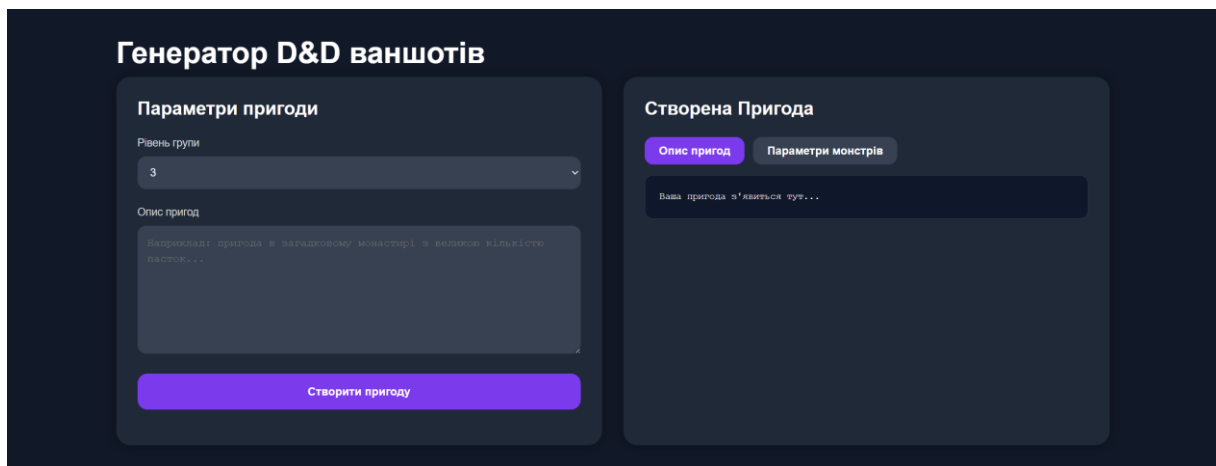


Рисунок 3.2 – Веб-інтерфейс користувача

Таким чином, реалізований програмний засіб забезпечує повний цикл автоматизованого створення пригод-ваншотів: від аналізу користувацького запиту до генерації завершеного сценарію, придатного для використання під час проведення ігрової сесії.

4.2 Реалізація бази сюжетних компонентів

Одним із ключових елементів розробленої системи генерації пригод є база сюжетних компонентів, яка виступає джерелом структурованих даних для формування сценарію. На відміну від підходів, що повністю покладаються на генерацію тексту мовною моделлю, запропонована система використовує попередньо підготовлений набір сюжетних елементів,

що дозволяє контролювати зміст пригоди, забезпечувати її тематичну узгодженість та зменшувати кількість логічних помилок під час генерації.

База сюжетних компонентів реалізована у вигляді набору JSON-файлів, кожен з яких відповідає окремій категорії даних. Така організація забезпечує незалежне редагування та розширення окремих наборів компонентів без необхідності зміни програмної логіки системи. Крім того, використання формату JSON спрощує інтеграцію даних із програмними модулями та забезпечує зручне представлення ієрархічних структур [20].

Для підтримки компонентного підходу база даних містить декілька тематичних груп компонентів, що використовуються на різних етапах побудови пригоди. До основних категорій належать сюжетні конфлікти, антагоністи, фракції, локації, загадки, соціальні взаємодії, архетипи пригод, структури підземель та монстри. Основні з них наведені в таблиці 3.2:

Таблиця 3.2 – Основні набори даних системи

Набір даних	Призначення
Conflicts	Центральні конфлікти пригод
Antagonists	Антагоністи та їх мотивації
Locations	Локації та середовища
Dungeons	Архетипи підземель
Puzzles	Загадки та дослідницькі сцени
Social Encounters	Соціальні взаємодії
Monsters	Супротивники для бойових сцен

Сюжетні конфлікти виступають основою пригоди та визначають її центральну проблему. Для кожного конфлікту зберігаються характеристики тематики, жанрової спрямованості, можливих антагоністів та рекомендованих типів локацій. Завдяки цьому система може підбирати компоненти, які відповідають загальній концепції сценарію.

Окрему групу становлять антагоністи пригоди. Для кожного антагоніста в базі зберігаються відомості про його тип, мотивацію,

характерні цілі, тематичну належність та можливі зв'язки з іншими компонентами пригоди. Подібна структура дозволяє формувати логічно обґрунтовані конфлікти та забезпечувати причинно-наслідкові зв'язки між подіями сценарію.

Для створення соціальної складової пригоди використовуються набори фракцій, неігрових персонажів та соціальних взаємодій. Вони містять інформацію про політичні угруповання, організації, культу, торговельні союзи та інші структури, які можуть брати участь у розвитку сюжету. Наявність подібних даних дозволяє формувати складніші сценарії з дипломатичними або детективними елементами.

Важливим компонентом бази є набір локацій. Кожна локація характеризується типом місцевості, тематикою, рівнем небезпеки та можливими сценарними подіями. Система використовує ці характеристики для підбору середовища, що відповідає параметрам Adventure Blueprint та обраному жанру пригоди.

Для урізноманітнення ігрового процесу база також містить набори загадок, дослідницьких сцен та сюжетних подій. Такі компоненти використовуються під час побудови структури пригоди та дозволяють створювати сценарії, які не обмежуються виключно бойовими взаємодіями.

Особливе значення для системи «Dungeons & Dragons» має база монстрів. Вона містить інформацію про характеристики супротивників, рівень складності, тип істоти, середовище існування та тематичну належність. Під час генерації бойових сцен система використовує ці дані для формування збалансованих зіткнень, що відповідають рівню персонажів та загальній концепції пригоди.

Для забезпечення логічної узгодженості між окремими компонентами кожен запис бази даних містить набір атрибутів та тегів. Саме вони використовуються під час роботи механізмів відбору компонентів та перевірки сумісності. Наявність формалізованих характеристик дозволяє автоматично визначати допустимість використання конкретного елемента в

межах сформованої пригоди. На рисунку 3.3 наведено приклад структури запису компонента.

```
{
  "goal": "Rescue missing villagers kidnapped by creatures beneath the old mine.",
  "category": "global",
  "levels": [1, 4],
  "tags": ["rescue", "village", "underground", "monsters", "mystery"],
  "dmg_type": "Location-Based",
  "expected_elements": ["captives", "monsters", "mine_shafts"]
},
```

Рисунок 3.3 – Приклад структури запису компонента

Структура бази сюжетних компонентів була спроектована з урахуванням можливості подальшого розширення. Нові типи пригод, додаткові категорії локацій, монстрів або сюжетних подій можуть бути додані шляхом поповнення відповідних JSON-файлів без внесення змін до основних алгоритмів генерації. Такий підхід підвищує масштабованість системи та спрощує її подальший розвиток.

Таким чином, реалізована база сюжетних компонентів виступає фундаментом усієї системи генерації пригод. Вона забезпечує структуроване зберігання даних предметної галузі, підтримує механізми автоматичного відбору компонентів та створює основу для побудови логічно узгоджених сценаріїв відповідно до правил «Dungeons & Dragons».

4.3 Сценарії тестування системи

Після завершення реалізації програмного засобу було проведено експериментальне дослідження його роботи. Основною метою тестування є перевірка здатності системи формувати логічно узгоджені пригоди різних типів відповідно до заданих користувачем параметрів. Особлива увага приділялася оцінці коректності побудови Adventure Blueprint, якості відбору

сюжетних компонентів, узгодженості сформованої структури пригоди та відповідності фінального сценарію початковому запиту.

Для забезпечення об'єктивності дослідження було сформовано набір тестових запитів, які охоплюють різні жанри та стилі пригод, характерні для системи «Dungeons & Dragons». Тестування виконувалося для кількох основних типів сценаріїв: класичних пригод у підземеллях, пригод із дослідженням дикої місцевості, сценаріїв пограбування та пригод виживання.

Першу групу становили пригоди типу «Dungeon Adventure». Для таких сценаріїв характерним є дослідження підземель, пошук артефактів, подолання пасток та протистояння головному антагоністу в межах замкненої локації. Основною метою тестування даної категорії було оцінювання здатності системи формувати структуру підземель, коректно підбирати тематичних супротивників та підтримувати послідовність розвитку пригоди від вступу до фінального зіткнення.

До другої групи належали пригоди типу «Wilderness Adventure», орієнтовані на дослідження відкритих територій. У цьому випадку перевірялася здатність системи створювати маршрути подорожей, підбирати природні локації, формувати випадкові події та забезпечувати логічний зв'язок між окремими етапами дослідження світу.

Окрему категорію тестових сценаріїв становили пригоди типу «Heist Adventure», основною метою яких є проникнення на захищений об'єкт і викрадення певного предмета або інформації. Такі сценарії дозволяють перевірити ефективність роботи механізмів генерації соціальних взаємодій, використання фракцій та побудови складніших сюжетних структур із декількома альтернативними шляхами досягнення мети.

Для оцінювання роботи системи в умовах обмежених ресурсів використовувалися пригоди типу «Survival Adventure». У подібних сценаріях головний акцент робиться на виживанні персонажів у ворожому

середовищі, пошуку ресурсів та подоланні природних або надприродних загроз. Генерація таких пригод дозволяє перевірити здатність системи формувати нетипові конфлікти, які не зводяться виключно до бойових взаємодій.

Під час тестування кожен сценарій генерувався декілька разів із використанням однакових початкових параметрів. Це дозволило оцінити рівень варіативності результатів та перевірити, чи зберігає система загальну концепцію пригоди при зміні окремих сюжетних компонентів. Окремо аналізувалася робота механізму перевірки сумісності компонентів, який повинен був виключати виникнення логічно суперечливих комбінацій.

Для кожної згенерованої пригоди оцінювалися такі критерії:

- відповідність сформованого сценарію початковому запиту;
- логічна узгодженість сюжетних подій;
- відповідність обраних компонентів тематиці пригоди;
- відсутність суперечностей між елементами;
- коректність формування бойових сцен;
- цілісність композиційної структури сценарію;
- різноманітність отриманих результатів.

Наведені критерії дозволяють комплексно оцінити не лише якість сформованого текстового сценарію, а й ефективність роботи окремих алгоритмічних модулів системи, відповідальних за побудову Adventure Blueprint, відбір компонентів і забезпечення структурної узгодженості пригоди.

Для систематизації результатів тестування було сформовано набір контрольних запитів, наведений у таблиці 3.3.

Таблиця 3.3 – Набір тестових сценаріїв

№	Тип пригоди	Приклад запиту
1	Dungeon Adventure	Стародавнє прокляте підземелля з культом некромантів

Продовження таблиці 3.3

№	Тип пригоди	Приклад запиту
2	Dungeon Adventure	Пошук легендарного артефакту у забутій фортеці
3	Wilderness Adventure	Дослідження небезпечного лісу, де зникають мандрівники
4	Wilderness Adventure	Подорож через заражені болота до покинутого храму
5	Heist Adventure	Викрадення магічного артефакту з резиденції аристократа
6	Heist Adventure	Проникнення до таємного сховища злочинної організації
7	Survival Adventure	Вживання після корабельної аварії на таємничому острові
8	Survival Adventure	Порятунок із засніжених гір під час магічної бурі

Результати проведеного тестування показали, що система здатна успішно формувати пригоди різних жанрових категорій, зберігаючи логічну цілісність сценарію та відповідність початковим параметрам генерації. Отримані сценарії демонстрували достатній рівень різноманітності, а використання формальної моделі пригоди та механізму перевірки сумісності компонентів дозволило уникнути більшості сюжетних суперечностей, характерних для прямої генерації тексту мовними моделями.

3.4 Аналіз результатів генерації

Після проведення експериментального тестування було виконано аналіз сформованих сценаріїв пригод з метою оцінювання якості роботи запропонованої моделі генерації (таблиця 4.4). Основна увага приділялася перевірці відповідності сформованих пригод початковим параметрам користувацького запиту, логічній узгодженості сюжету, тематичній

цілісності компонентів, а також відповідності сформованих бойових сцен правилам системи «Dungeons & Dragons».

Таблиця 3.4 – Результати оцінювання роботи системи

Критерій оцінювання	Результат
Відповідність початковому запиту	Повністю забезпечується
Узгодженість сюжетної структури	Висока
Логічність розвитку подій	Висока
Тематична сумісність компонентів	Забезпечується
Відсутність суперечливих комбінацій	Забезпечується механізмом перевірки
Відповідність правилам D&D	Забезпечується для основних компонентів
Якість текстового представлення	Висока зв'язність та читабельність

Для аналізу було використано набір тестових сценаріїв різних жанрових категорій, що включав пригоди типу «Dungeon Adventure», «Wilderness Adventure», «Heist Adventure» та «Survival Adventure». Кожний сценарій генерувався незалежно, після чого здійснювався якісний аналіз отриманого результату.

Першим критерієм оцінювання стала відповідність сформованого сценарію початковому запиту користувача. У всіх протестованих прикладах система коректно визначала основні характеристики майбутньої пригоди, зокрема тематику, тип локації, характер конфлікту та загальний жанр сценарію. Побудова Adventure Blueprint на початковому етапі дозволила сформувати єдину концепцію пригоди ще до початку текстової генерації, що забезпечило відповідність фінального результату початковим вимогам.

Другим критерієм стала структурна узгодженість пригоди. У сформованих сценаріях простежується послідовний розвиток сюжету: вступ поступово переходить до дослідження локацій, розвитку конфлікту, кульмінаційної події та завершення пригоди. Побудова сценарію на основі структурованої послідовності сцен дозволила уникнути хаотичних переходів між подіями та забезпечила логічний розвиток історії.

Як приклад роботи системи наведено фрагмент однієї зі згенерованих пригод. У сценарії присутні всі основні структурні елементи, сформовані на попередніх етапах генерації: вступ, сюжетна зав'язка, опис ключових локацій, неігрових персонажів, послідовність сюжетних сцен, бойові та соціальні взаємодії, а також завершення пригоди. Незважаючи на те, що кінцевий текст створюється мовною моделлю, його зміст визначається сформованою структурною моделлю пригоди та відібраними компонентами.

Аналіз наведеного прикладу на рисунку 3.4 демонструє, що всі ключові елементи пригоди логічно пов'язані між собою. Центральний конфлікт узгоджується з тематикою сценарію, антагоніст відповідає обраному жанру, а бойові сцени інтегровані у розвиток сюжету. Це підтверджує ефективність використання структурної моделі пригоди та системи перевірки сумісності компонентів.

Наступним критерієм було оцінювання тематичної сумісності сюжетних компонентів. Усі відібрані локації, антагоністи, фракції, загадки, неігрові персонажі та бойові сцени відповідали загальній тематиці пригоди. Застосування моделі перевірки сумісності компонентів дозволило автоматично виключити комбінації, що суперечили обраній концепції сценарію або порушували логіку розвитку подій.

Окрему увагу було приділено бойовим сценам, оскільки саме вони найбільше залежать від дотримання правил «Dungeons & Dragons». Аналіз показав, що система формує бойові зіткнення, які відповідають тематиці пригоди та логічно інтегруються в розвиток основного конфлікту. Підбір монстрів здійснюється з урахуванням рівня персонажів, характеристик

супротивників та ролі бойової сцени у сюжеті, що дозволяє уникати випадкових або невмотивованих сутичок.

Для кількісного аналізу результатів було проведено експертне оцінювання згенерованих пригод за двома критеріями якості. Кожен критерій оцінювався за п'ятибальною шкалою, де 1 відповідає найнижчій якості, а 10 – найвищій.

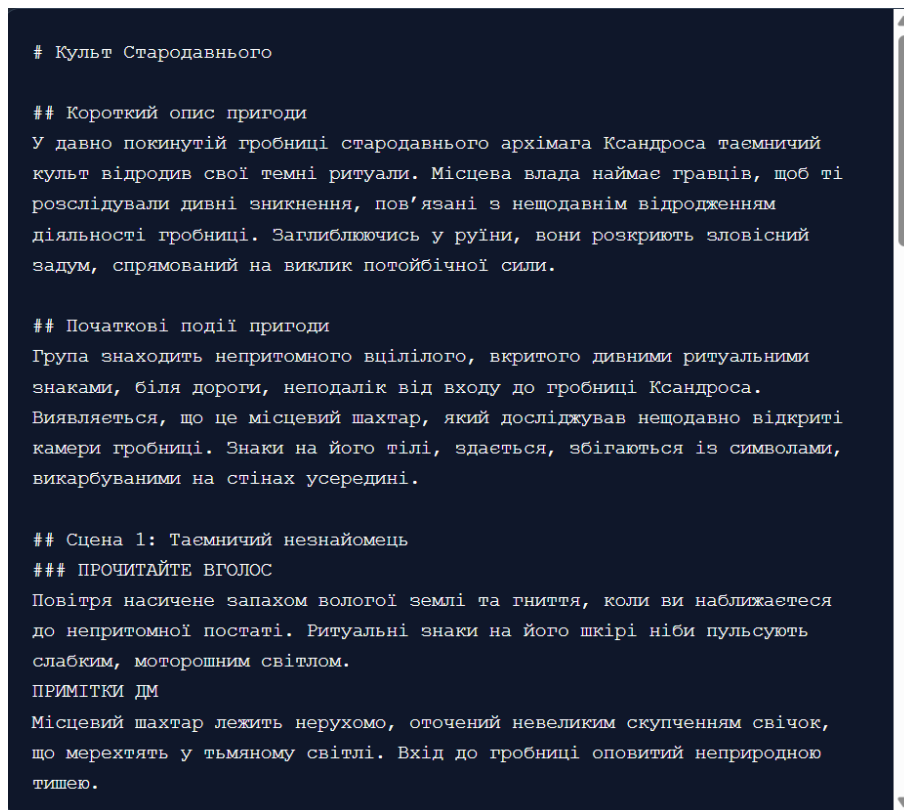


Рисунок 3.4 – Фрагмент згенерованої пригоди

Під час дослідження також було оцінено якість фінального текстового представлення пригоди. Використання мовної моделі на завершальному етапі генерації забезпечило формування зв'язного художнього тексту, який містить описи локацій, характеристики персонажів, сюжетні переходи та рекомендації для майстра гри. При цьому мовна модель виконувала не створення сценарію «з нуля», а деталізацію вже побудованої структурної моделі пригоди, що позитивно вплинуло на узгодженість результату.

Проведений аналіз показав, що запропонована модель забезпечує стабільне формування структурно узгоджених пригод різної тематики. Поєднання формальної моделі Adventure Blueprint, механізму перевірки сумісності компонентів, структурованої побудови сценарію та генерації тексту за допомогою мовної моделі дозволяє створювати пригоди, які одночасно відповідають користувацькому запиту, підтримують логічну цілісність сюжету та узгоджуються з основними правилами системи «Dungeons & Dragons».

3.5 Дослідження впливу користувацького зворотного зв'язку на якість генерації

Однією з особливостей розробленої системи є можливість поступового вдосконалення процесу генерації за рахунок накопичення користувацьких оцінок. На відміну від більшості генеративних систем, у яких кожна генерація виконується незалежно від попередніх результатів, запропонована модель використовує механізм зворотного зв'язку, що дозволяє адаптувати процес вибору сюжетних компонентів відповідно до накопиченого досвіду використання.

Після завершення генерації користувач оцінює сформовану пригоду за двома основними критеріями:

- релевантність (Relevance), яка характеризує відповідність пригоди початковому запиту;
- структурна узгодженість (Consistency), що відображає логічність побудови сценарію, взаємозв'язок сюжетних подій та відповідність правилам предметної галузі.

Отримані оцінки зберігаються у системі та використовуються для оновлення рейтингів сюжетних компонентів. Надалі ці рейтинги враховуються під час вибору альтернативних компонентів при побудові нових пригод. При цьому формальна модель пригоди, система логічних

обмежень та алгоритми перевірки сумісності залишаються незмінними, що гарантує збереження структурної цілісності сценарію.

Для оцінювання ефективності запропонованого підходу було проведено серію із 26 послідовних генерацій пригод різної тематики. Після кожної генерації виконувалося оцінювання результату, а отримані значення використовувалися для коригування рейтингів відповідних сюжетних компонентів.

Для кожної генерації фіксувалися значення показників релевантності та структурної узгодженості, після чого будувалися графіки зміни цих оцінок у процесі накопичення користувацького зворотного зв'язку. Це дозволило оцінити вплив адаптивного механізму на стабільність роботи системи.

Аналіз результатів, відображених на рисунку 4.5 показує, що показник структурної узгодженості протягом експерименту залишається стабільно високим. Середнє значення узгодженості становить приблизно 8,1 бала, а більшість пригод отримують оцінки від 8 до 10 балів.

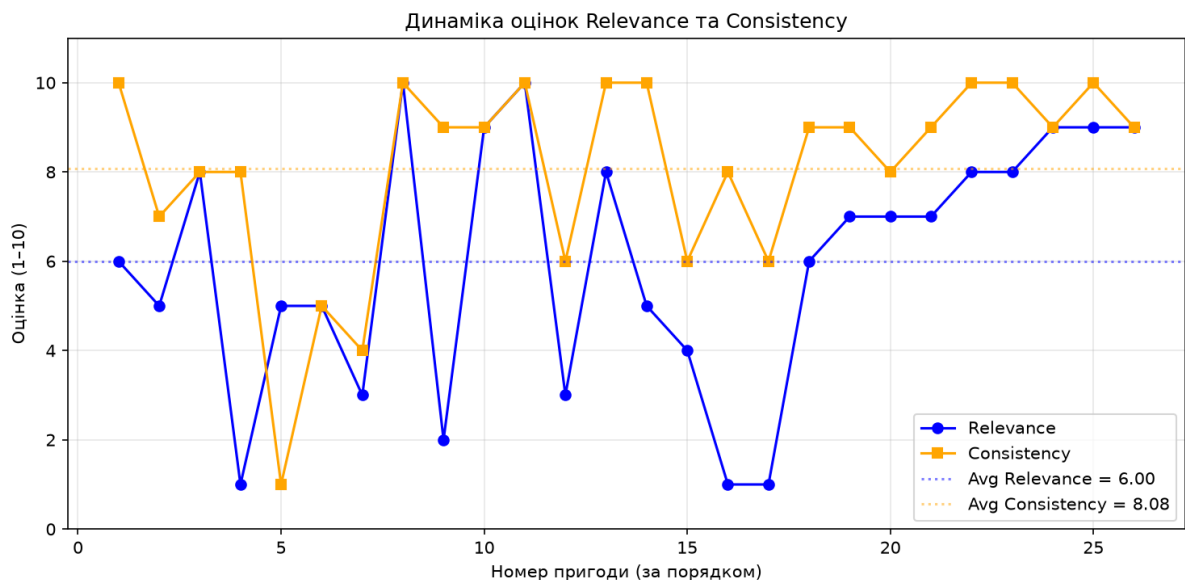


Рисунок 3.5 – Динаміка оцінок релевантності та структурної узгодженості пригод

Подібний результат пояснюється особливостями архітектури розробленої системи. Перед початком генерації тексту формується план пригоди, виконується відбір сумісних компонентів, здійснюється перевірка логічних обмежень та будується структура майбутньої пригоди. Таким чином, мовна модель працює не з довільним текстовим запитом, а з уже підготовленою структурованою моделлю сценарію. Саме це забезпечує високу логічну узгодженість навіть під час перших генерацій.

Іншу тенденцію демонструє показник релевантності. На початковому етапі експерименту його значення характеризуються значною варіативністю, що пояснюється відсутністю накопиченої статистики щодо якості використання окремих сюжетних компонентів. У міру накопичення користувацьких оцінок система отримує можливість віддавати перевагу компонентам, які раніше демонстрували кращі результати. Це приводить до поступового зростання релевантності сформованих пригод.

На завершальних етапах експерименту оцінки релевантності стабілізуються на рівні 8-9 балів, що свідчить про ефективність використання механізму адаптивного ранжування компонентів. При цьому структурна узгодженість сценаріїв залишається практично незмінною, оскільки забезпечується незалежними механізмами формальної перевірки.

Отримані результати підтверджують, що використання користувацького зворотного зв'язку дозволяє покращувати якість генерації без зміни параметрів мовної моделі та без її повторного навчання. Адаптація відбувається виключно на рівні вибору сюжетних компонентів, що значно спрощує реалізацію системи та зменшує обчислювальні витрати.

Отже, проведене дослідження підтверджує ефективність запропонованого механізму адаптивного оцінювання компонентів. Поєднання формальної моделі пригоди, системи логічного контролю та накопичення користувацьких оцінок дозволяє одночасно забезпечити високу структурну узгодженість сценаріїв і поступове підвищення їхньої відповідності запитам користувачів.

3.6 Реалізація інтелектуального компонента системи

Особливістю розробленого програмного засобу є використання великої мовної моделі не як самостійного генератора пригод, а як одного з компонентів гібридної системи генерації. На відміну від традиційного підходу, за якого мовна модель отримує лише текстовий запит користувача і самостійно формує повний сценарій, у запропонованій архітектурі більшість рішень приймається алгоритмічними модулями системи ще до звернення до моделі.

Взаємодія із мовною моделлю реалізована у вигляді окремого програмного модуля, який інкапсулює всі операції, пов'язані з формуванням запиту, передаванням його до сервера мовної моделі та обробкою отриманої відповіді. Такий підхід забезпечує незалежність інших компонентів системи від конкретної реалізації моделі та дозволяє легко замінити використовувану LLM без зміни логіки роботи програмного засобу.

Для локального виконання великої мовної моделі використовується платформа Ollama, яка забезпечує запуск моделей без використання зовнішніх хмарних сервісів. Це дозволяє уникнути передачі користувацьких даних стороннім серверам, зменшує залежність від мережевої інфраструктури та забезпечує можливість використання різних мовних моделей шляхом зміни лише конфігураційних параметрів системи.

Перед початком генерації виконується аналіз користувацького запиту, після чого формується модель пригоди, яка визначає основні характеристики майбутнього сценарію. Далі система здійснює пошук відповідних сюжетних компонентів, перевіряє їх логічну сумісність та будує структуру пригоди. Лише після завершення цих етапів формується структурований запит до мовної моделі.

Таким чином, мовна модель не створює сценарій «з нуля», а виконує деталізацію вже сформованої структури пригоди. Це суттєво зменшує

ймовірність появи логічних суперечностей, неузгоджених сюжетних переходів і невідповідностей правилам «Dungeons & Dragons».

Інтелектуальний компонент виконує такі основні функції:

- інтерпретацію структурованого контексту пригоди;
- генерацію художніх описів локацій;
- формування текстових описів неігрових персонажів;
- деталізацію сюжетних сцен;
- створення діалогів;
- побудову плавних переходів між окремими етапами пригоди;
- стилістичне оформлення фінального сценарію.

Завдяки використанню мовної моделі система отримує природний текст із високою мовною якістю без необхідності ручного написання шаблонів для кожного можливого сценарію.

Після завершення генерації англomовний текст автоматично передається до окремого модуля перекладу.

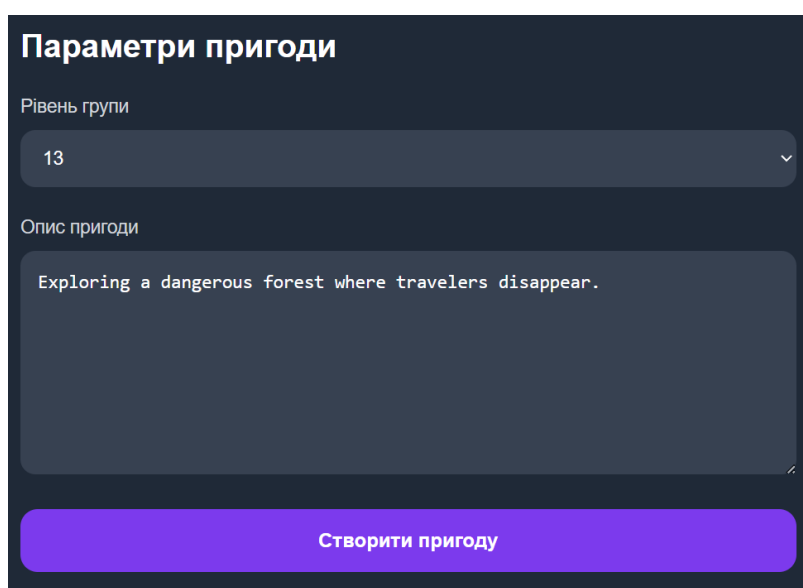
Важливою особливістю реалізації є використання структурованого промптингу. До мовної моделі передається не лише опис теми пригоди, а повний набір даних, сформований попередніми алгоритмами. Запит містить інформацію про жанр пригоди, основний конфлікт, антагоніста, ключові локації, неігрових персонажів, сюжетні сцени, бойові зіткнення та вимоги до структури фінального документа. Такий підхід дозволяє значно підвищити керованість процесом генерації.

Після завершення роботи користувач має можливість оцінити якість сформованої пригоди за кількома критеріями, зокрема релевантністю до початкового запиту та логічною узгодженістю сценарію. Отримані оцінки зберігаються у базі даних і можуть бути використані для подальшого вдосконалення системи, аналізу її ефективності та реалізації механізмів адаптивного налаштування генерації.

3.7 Порівняння результатів генерації з прямим використанням великої мовної моделі

Одним із ключових завдань даної роботи є визначення доцільності використання розробленої моделі генерації пригод порівняно з прямим використанням великої мовної моделі без попередньої структуризації даних. Незважаючи на високі можливості сучасних мовних моделей щодо створення художнього тексту, вони не забезпечують повного контролю над структурою сценарію, взаємною узгодженістю його компонентів та відповідністю вимогам системи правил «Dungeons & Dragons». Тому виникає необхідність у застосуванні додаткових механізмів, які забезпечують структуризацію вхідних даних та контроль процесу генерації до моменту створення фінального тексту.

Для оцінювання ефективності запропонованого підходу було проведено експериментальне порівняння двох способів генерації пригод за однакових початкових умов. Для забезпечення коректності порівняння в обох випадках використовувався один і той самий користувацький запит (рисунок 3.6).



Параметри пригоди

Рівень групи

13

Опис пригоди

Exploring a dangerous forest where travelers disappear.

Створити пригоду

Рисунок 3.6 – Користувацький запит для порівняння підходів

У першому випадку користувацький запит безпосередньо передавався великій мовній моделі. Усі етапи формування сюжету, вибору персонажів, створення локацій та побудови пригоди виконувалися виключно засобами мовної моделі без використання додаткових алгоритмів контролю структури. Інструкція моделі наведена на рисунку 3.7.

```
You are a professional D&D 5e adventure writer.

Create a complete one-shot adventure based ONLY on the following request.

User request:
{user_prompt}

Write:
- Adventure Summary
- Adventure Hook
- 4-6 scenes
- Final battle
- Rewards

Do not ask questions.
Invent everything yourself.

""""
```

Рисунок 3.7 – Промпт для моделі без Adventure Blueprint

В цьому випадку було отримано наступний результат:

«Назва модуля: «Розбиті віддзеркалення»

Сюжет: Село Елдріда розташоване на березі містичного озера, поверхня води якого, як кажуть, є дзеркалом душі. Село охопили дивні події: мешканці бачать видіння своїх найпотаємніших страхів і бажань, а віддзеркалення в озері почало розбиватися, вивільняючи темні енергії, які загрожують поглинути селище.

Мета: Гравцям доручено розкрити джерело цих заворушень і покласти їм край, перш ніж село буде розірване на шматки власними темними поривами.

Місце дії: Елдріда та містичне озеро Ель'гороф.

Село Елдрида: Невелике сільське поселення, збудоване навколо жвавої ринкової площі. Селяни не розголошують своїх проблем, але атмосфера тут напружена.

Озеро Ель'горот: Священне місце з таємничими властивостями. Поверхню води можна розглядати як відображення душі людини.

Сюжет: Гравці прибувають до Елдриди бурхливої ночі, саме тоді, коли селяни повідомляють про раптове зростання кількості дивних подій. Місцевий лорд, сер Харлон, тепло вітає їх і доручає їм розкрити джерело цих заворушень.

Під час розслідування гравці зіткнуться з:

Привидами минулого Елдриди: Гравці дізнаються, що село має темну історію вбивств, зрад і трагічних нещасних випадків. Ці мстиві духи проявляються у вигляді видінь у свідомості селян.

Темна енергія озера: Гравці знайдуть докази того, що озеро випромінює дивну енергетичну сигнатуру, яка приваблює привидів і підживлює їхній гнів.

Основне завдання:

1. Знайти джерело енергетичної сигнатури: Гравці повинні відстежити джерело темної енергії, що випромінюється з озера.

2. Визначте місце походження джерела: Гравці з'ясують, що енергетичний сигнал пов'язаний із стародавнім ритуалом, який пішов не так, як треба, і пробудив злісну істоту.

3. Перемогти істоту та зняти прокляття: Гравці повинні зіткнутися з істотою та перемогти її, знявши прокляття, яке тяжіє над селом.

Додаткові завдання:

Повернути зниклу кохану людину: Один із селян попросить гравців допомогти йому повернути зниклу кохану людину, яку викрали мстиві духи.

Розкрити приховані таємниці: Гравці натраплять на стародавні тексти та артефакти, що містять підказки про темне минуле Елдриди.

Теми:

Зустріч зі страхами: Гравці повинні зіткнутися зі своїми власними страхами та слабкостями, щоб подолати виклики, що чекають на них попереду.

Сила відображення: Містичне озеро слугує дзеркалом душі, підкреслюючи важливість самоусвідомлення та самоаналізу.

Тон: Зловісний, тривожний, з прихованим відчуттям терміновості та небезпеки, що чатує за кожним поворотом.»

Отриманий сценарій є прикладом генерації, що повністю базується на можливостях великої мовної моделі без використання додаткових механізмів структуризації. Незважаючи на логічну зв'язність тексту, користувач не має можливості контролювати вибір сюжетних компонентів, а система не виконує перевірку їх взаємної сумісності, що може призводити до появи суперечностей або невідповідності окремих елементів правилам предметної галузі. Незважаючи на достатньо високу якість згенерованого тексту, вже на цьому етапі можна помітити, що сценарій містить елементи, сформовані виключно на основі статистичних закономірностей мовної моделі. Зокрема, відсутні гарантії використання офіційних монстрів, збалансованості бойових зіткнень, узгодженості між окремими сюжетними компонентами та відповідності структури пригоди вимогам системи «Dungeons & Dragons». Саме ці обмеження стали підставою для розробки запропонованої гібридної моделі генерації. Запропонований підхід поєднує переваги великих мовних моделей із детермінованими алгоритмами відбору та перевірки даних.

У другому випадку той самий запит оброблявся розробленою системою генерації пригод. На початковому етапі виконувалося формування Adventure Blueprint, після чого система здійснювала підбір сюжетних компонентів, перевірку їх сумісності, побудову структури пригоди та формування контексту генерації. Така послідовність етапів забезпечує попередню валідацію сюжетних компонентів ще до початку генерації тексту, що істотно зменшує ймовірність появи логічних суперечностей у

фінальному сценарії. Лише після завершення цих етапів структуровані дані передавалися мовній моделі для створення фінального текстового сценарію.

Інструкція моделі відображена на рисунку 3.8.

```
Your task is to create a professional one-shot D&D 5e adventure module.

CRITICAL REQUIREMENT: You MUST strictly adapt all components below to fit the user's core idea: "{user_prompt}".
If the components contain generic names, re-skin and re-flavor them to match the user's requested setting and plot.
**Adventure Theme:** {theme}
**Genre:** {genre}
**Tone:** {tone_str}
**Expected length:** {length_hint}

**Adventure Blueprint (summary):**
- Location type: {blueprint.get('location_type', 'unknown')}
- Goal type: {blueprint.get('goal_type', 'unknown')}
- Expected elements: {'', '.join(blueprint.get('expected_elements', []))}

**Key Components:**
- Hook: {components.get('hook', {}).get('hook', '')}
- Goal: {components.get('goal', {}).get('goal', '')}
- Location: {components.get('location', {}).get('location', '')}
- Antagonist: {components.get('antagonist', {}).get('name', '')}
- Puzzles: [{p.get('name', '') for p in components.get('puzzles', [])}]
- Social encounters: [{s.get('name', '') for s in components.get('socials', [])}]
- Monsters: {monsters_str}

**Deep World Data:**
{ecology_text}
{topology_text}
{challenges_text}
{env_state_text}
{rooms_text}

**Scene-by-Scene Structure:**
{scenes_text}

**Social & Contextual Content (Formatted for easy reading):**
{npcs_text}
{hazards_text}
{encounters_text}
```

Рисунок 3.8 – Промпт для моделі з Adventure Blueprint

Результат ускладненої генерації:

«Короткий опис пригоди

У самому серці «Зелених пусток» щонаочі лісом лунає таємнича серія криків, що наводить жах на мешканців сусідніх сіл. Кажуть, що в цій дикій місцевості заховано рідкісний і цінний природний ресурс, відомий як «Кристали Зоряного Світла». Група повинна подолати небезпечну місцевість, перемогти древніх охоронців і зіткнутися з грізними ворогами, щоб виявити джерело криків і заволодіти бажаними кристалами.

Початок пригоди

До групи підходить старійшина села, її очі широко розширені від тривоги. «Будь ласка, ви мусите нам допомогти. Крики в лісі стають дедалі гучнішими, і ми боїмося, що сталося щось жахливе. Ми намагалися відправити туди своїх мисливців, але вони не повернулися. Якщо ви зможете розібратися з цими звуками та знайти їх джерело, ми запропонуємо вам нагороду, про яку ви навіть не мріяли». Вона вручає їм грубу карту околиць, вказуючи на кілька місць, де чулися крики.

Сцена 1: Свідок, якого переслідують привиди

[ПРОЧИТАЙТЕ ВГОЛОС]

Коли ви прибуваєте до села, у повітрі витає моторошна тиша. Селяни туляться один до одного, їхні обличчя освітлюють лише мерехтливі смолоскипи. Слова старійшини все ще лунають у ваших головах, поки ви прямуєте до призначеного місця зустрічі.

[ПРИМІТКИ ДМ]

Свідок — травмована торговка на ім'я Елара — чекає на ваше прибуття. Відтоді, як почалися крики, вона бачить дивні видіння та страждає від кошмарів. Її страх відчутний, але вона поділиться своїми знаннями з групою, якщо ті зможуть завоювати її довіру.

Сцена 2: Дослідження — Залишки мегаструктури

[ЧИТАТИ ВГОЛОС]

Коли ви заглиблюєтеся в ліс, крони дерев над головою стають густішими, фільтруючи сонячне світло й занурюючи ліс у смарагдову пітьму. Повітря насичене запахом вологої землі та гнилого листя. Ваш шлях веде до величезної стародавньої споруди, частково похованої під чагарниками.

[ПРИМІТКИ ДМ]

Залишки мегаструктури приховують таємниці загадкового минулого «Зелених пусток». Розкидані артефакти та дивні символи натякають на стародавню цивілізацію, яка колись процвітала тут. Будьте обережні:

місцевість підступна — тут є круті схили та приховані пастки, що чекають на необережних.

Сцена 3: Дослідження — Лігво стародавнього аболета

[ЧИТАТИ ВГОЛОС]

Густа листя розступається, відкриваючи галявину, оповиту туманом. У центрі стоїть величезне, покручене дерево, на корі якого викарбувані пульсуючі руни, що здаються поєднанням магії та технології. Повітря насичене смородом гниття та розкладу.

[ПРИМІТКИ ДМ]

У цьому лігві мешкає Стародавній Аболет — супротивник з CR 11,0. Його присутність зіпсувала навколишнє середовище, змушуючи дерева корчитися й викручуватися від болю. Будьте готові до його здібностей «Зловісний чар».

Сцена 4: Бій, Кульмінація — Мисливські угіддя дракона

[ЧИТАТИ ВГОЛОС]

На галявині спалахує хаос, коли молодий дракон розриває землю своїми величезними пазурами. Менші істоти — примарні духи, що, здається, народилися з самої темряви, — роєм оточують групу, наступаючи їм на п'яти.

[ПРИМІТКИ ДМ]

«Страж у темряві», конструкт рівня CR 7.0, виконує роль охоронця дракона. Його здатність «Прокляття мандрівника» змушує блукаючих мандрівників спотикатися та втрачати рівновагу. Будьте обережні: дракон атакуватиме зверху, використовуючи свою здатність «Багаторазова атака». Група повинна адаптувати свою стратегію, щоб протистояти як повітряним атакам молодого дракона, так і наземним атакам Вартового.

Наслідки та нагороди

Битва виграна, але якою ціною? Кристали Зоряного Світла знайдені, і джерело криків нарешті замовкло. Однак дії групи порушили рівновагу природи в Зелених Диких Землях. Як наслідок:

За свою хоробрість герої отримують 100 золотих монет.

Ті, хто доведе свою гідність, можуть отримати карту, що веде до додаткового прихованого місця, де знаходяться ще кристали.

Старійшина знову підходить до вас, щоб подякувати.»

Аналіз сформованого сценарію показує, що використання Adventure Blueprint та механізмів попереднього відбору компонентів забезпечує вищий рівень керованості процесом генерації. Усі ключові елементи пригоди: сюжетні сцени, локації, персонажі та бойові зіткнення – відповідають заздалегідь сформованій структурі, а використання компонентів бази знань дозволяє уникнути довільного створення об'єктів, які можуть суперечити правилам предметної галузі. У результаті фінальний сценарій є більш структурованим і придатним до практичного використання майстром гри.

Аналіз результатів показав, що в обох випадках мовна модель формує зв'язний художній текст, однак отримані сценарії суттєво відрізняються за рівнем структурної узгодженості. У випадку прямої генерації структура пригоди формується виключно на основі внутрішніх статистичних закономірностей мовної моделі. Внаслідок цього окремі сюжетні події можуть бути недостатньо пов'язаними між собою, а розвиток конфлікту іноді відбувається нерівномірно. Також спостерігається тенденція до додавання нових персонажів, сюжетних елементів або деталей, які не були передбачені початковим задумом пригоди. Подібна поведінка є характерною для великих мовних моделей та пов'язана з їх генеративною природою.

На відміну від цього, у запропонованій системі основні параметри пригоди визначаються ще до початку текстової генерації. Використання Adventure Blueprint дозволяє зафіксувати жанр пригоди, її тематику, конфлікт, тип локації та ключових учасників подій. Завдяки цьому всі подальші етапи побудови сценарію виконуються в межах заздалегідь визначеної концепції.

Окремою перевагою запропонованої моделі є використання офіційної бази монстрів «Dungeons & Dragons». На відміну від прямої генерації, де

мовна модель може створювати неіснуючих істот або змінювати характеристики відомих монстрів, у розробленій системі всі противники відбираються із попередньо сформованої бази даних. Це забезпечує відповідність офіційним правилам гри та дозволяє використовувати пригоду без додаткової перевірки майстром. Тож, у випадку такої генерації, ми отримаємо ворогів, підібраних згідно офіційної документації гри, а не вигаданий моделлю (рисунок 3.9).

Дорослий мідний дракон			
Дракон • Величезний • CR 14 • XP 11500			
ОСНОВНА ІНФОРМАЦІЯ			
Світгляд	хаотичне зло	КД	18 (природна броня)
ОЗ	252	Швидкість	пройти 12 метрів, пролетіти 24 метри, проплисти 6 метрів, піднятися 12 метрів, повзти 6 метрів.
ОЦІНКИ ЗДІБНОСТЕЙ			
сил	22	СПР	22
тл	30	ІНТ	18
муд	24	ХАР	26

Рисунок 3.9 – Приклад монстра підбраного для пригоди

Додаткову роль відіграє механізм відбору та перевірки сумісності компонентів. На відміну від прямої генерації, де персонажі, локації та сюжетні події створюються без попередньої перевірки взаємозв'язків, у розробленій системі кожен компонент проходить етап узгодження з іншими елементами пригоди. Це дозволяє зменшити кількість логічних суперечностей та забезпечити цілісність сюжетної структури.

Ще однією важливою відмінністю є наявність явної композиційної структури сценарію. Під час роботи системи попередньо формується послідовність сюжетних сцен, що включає вступ, розвиток конфлікту,

ключові події, кульмінацію та завершення пригоди. Завдяки цьому фінальний текст має більш передбачувану структуру та краще відповідає особливостям формату ваншоту.

Проведений експеримент підтвердив, що використання лише великої мовної моделі дозволяє отримувати якісний художній текст, проте не забезпечує достатнього рівня контролю над внутрішньою структурою пригоди. Натомість запропонована модель поєднує творчі можливості мовної моделі з формалізованим описом пригоди та механізмами контролю її побудови.

Таким чином, проведений експеримент демонструє, що запропонована система не замінює велику мовну модель, а розширює її можливості шляхом попередньої алгоритмічної обробки даних. Використання Adventure Blueprint, семантичного пошуку, перевірки сумісності компонентів і побудови структури пригоди дозволяє перетворити мовну модель із самостійного генератора тексту на керований інструмент формування сценаріїв. Це забезпечує вищу структурну узгодженість пригод, кращу відповідність правилам «Dungeons & Dragons» та підвищує стабільність результатів генерації.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальну задачу автоматизації процесу створення пригод формату ваншотів для настільної рольової гри «Dungeons & Dragons» шляхом розроблення програмного засобу, що поєднує алгоритмічні методи відбору сюжетних компонентів із можливостями великих мовних моделей.

У процесі виконання роботи було проведено аналіз існуючих підходів до автоматизованої генерації сюжетного контенту, методів процедурної генерації, сучасних технологій обробки природної мови та застосування великих мовних моделей у задачах інтерактивного сторітелінгу. Встановлено, що використання лише мовної моделі не забезпечує достатнього рівня структурної узгодженості пригод, тому доцільним є застосування гібридного підходу, який поєднує алгоритмічне формування структури сценарію та генерацію природного тексту.

У роботі запропоновано модель формування Adventure Blueprint, що забезпечує перетворення користувацького запиту у формалізований опис майбутньої пригоди із визначенням жанру, тематики, тональності, типу пригоди та інших структурних характеристик. Розроблено модель відбору сюжетних компонентів із використанням семантичного пошуку, механізмів перевірки логічної сумісності та побудови послідовності сюжетних сцен.

Запропоновано модель побудови структурованої інструкції генерації, яка об'єднує параметри Adventure Blueprint, відібрані сюжетні компоненти та структуру пригоди у єдиний контекст для мовної моделі. Використання структурованого промптингу дозволило підвищити керованість процесом генерації та забезпечити відповідність сформованого сценарію заданим параметрам.

Розроблено модель адаптивного оцінювання сюжетних компонентів, яка використовує користувацький зворотний зв'язок для накопичення рейтингів окремих елементів бази знань. Запропонований механізм дозволяє

поступово покращувати якість відбору компонентів без необхідності донавчання великої мовної моделі, що забезпечує простоту реалізації та можливість подальшого розвитку системи.

Реалізовано програмний засіб автоматизованої генерації пригод, що включає модулі аналізу користувацького запиту, формування Adventure Blueprint, семантичного пошуку компонентів, побудови структури сценарію, генерації тексту за допомогою великої мовної моделі та автоматичного перекладу сформованої пригоди українською мовою. Архітектура системи побудована за модульним принципом, що забезпечує можливість незалежного вдосконалення окремих компонентів і заміни мовної моделі без зміни загальної логіки роботи програмного засобу.

Результати роботи підтверджують ефективність запропонованого гібридного підходу до генерації пригод. Попередня алгоритмічна побудова структури сценарію, використання семантичного пошуку та логічної перевірки компонентів дозволяють суттєво підвищити узгодженість сформованих пригод порівняно з генерацією виключно на основі користувацького запиту.

Практична цінність роботи полягає у створенні програмного засобу, який може використовуватися майстрами гри для швидкої підготовки сценаріїв, а також слугувати основою для розроблення інтелектуальних систем підтримки інтерактивного сторітелінгу та автоматизованого наративного дизайну.

Перспективами подальших досліджень є розширення бази сюжетних компонентів, удосконалення механізмів адаптивного відбору на основі накопиченого користувацького досвіду, порівняльне оцінювання різних великих мовних моделей у процесі генерації пригод, а також впровадження агентних підходів для автоматичного моделювання поведінки неігрових персонажів і розвитку сюжету.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Dungeons and Dragons as a dialog challenge for artificial intelligence / C. Callison-burch et al. *Proceedings of the 2022 conference on empirical methods in natural language processing*, Abu Dhabi, United Arab Emirates. Stroudsburg, PA, USA, 2022. URL: <https://doi.org/10.18653/v1/2022.emnlp-main.637> (date of access: 16.06.2026).

2) Delafuente P., Honraopatil A., J. Martin L. Does reasoning help llm agents play dungeons and dragons? A prompt engineering experiment. *Arxiv*. 2025. URL: <https://doi.org/10.48550/arxiv.2510.18112>. (date of access: 16.06.2026).

3) Lewis P. Perez E. Piktus A. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Arxiv*. 2020. URL: <https://doi.org/10.48550/arxiv.2005.11401>. (date of access: 16.06.2026).

4) Shuster K. Poff S. Chen M. Retrieval augmentation reduces hallucination in conversation. *Arxiv*. 2021. URL: <https://doi.org/10.48550/arxiv.2104.07567>. (date of access: 16.06.2026).

5) Riedl M. O., Bulitko V. Interactive narrative: an intelligent systems approach. *AI magazine*. 2012. T. 34, № 1. P. 67. URL: <https://doi.org/10.1609/aimag.v34i1.2449> (дата звернення: 14.06.2026).

6) Zammit M., Liapis A., Yannakakis G. N. CrawlLM: an llm-based pipeline for game asset generation. *IEEE transactions on games*. 2026. P. 1–12. URL: <https://doi.org/10.1109/tg.2026.3655612> (date of access: 02.07.2026). (date of access: 16.06.2026).

7) Procedural content generation for games / M. Hendrikx et al. *ACM transactions on multimedia computing, communications, and applications*. 2013. Vol. 9, no. 1. P. 1–22. URL: <https://doi.org/10.1145/2422956.2422957> (date of access: 02.07.2026). (date of access: 16.06.2026).

8) Vechtomova O. Introduction to information retrieval Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (Stanford University, Yahoo!

Research, and University of Stuttgart) Cambridge: cambridge university press, 2008, *Computational linguistics*. 2009. Vol. 35, no. 2. P. 307–309. URL: <https://doi.org/10.1162/coli.2009.35.2.307> (date of access: 16.06.2026).

9) Crawford K. Worlds without number. Sine nomine publishing, 2021. 397 p.

10) Wyatt J., Shwalb J.R., Cordell B.R. Dungeons & Dragons player's handbook. – Wizards of the coast, 2024. – 384 p.

11) Vaswani A. Shazeer N. Parmar N. Attention is all you need. *arXiv*. 2017. URL: <https://doi.org/10.48550/arxiv.1706.03762>. (date of access: 16.06.2026).

12) Park J. S., O'Brien J., Cai C. J., S. Bernstein. Generative agents: interactive simulacra of human behavior. *UIST '23: Proceedings of the 36th annual acm symposium on user interface software and technology*. 2023. Vol. 2. P. 1–22. URL: <https://doi.org/10.1145/3586183.3606763>. (date of access: 16.06.2026).

13) Learning to speak and act in a fantasy text adventure game / J. Urbanek et al. *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, Hong Kong, China. Stroudsburg, PA, USA, 2019. URL: <https://doi.org/10.18653/v1/d19-1062> (date of access: 16.06.2026).

14) Mearls M., Crawford J. Dungeons & Dragons dungeon master's guide. Wizards of the coast, 2014. 320 p.

15) Mearls M., Crawford J. Dungeons & Dragons monster manual. Wizards of the coast, 2014. 352 p.

16) Muennighoff N., Tazi H., Magne L. Et al. MTEB: massive text embedding benchmark. *Proceedings of the 17th conference of the european chapter of the association for computational linguistics (EACL)*. Dubrovnik, Croatia, 2023. P. 2012–2037. URL: <https://aclanthology.org/2023.eacl-main.148> (date of access: 16.06.2026).

- 17) Nader A., Shufan W., Whiting J., et. al. A dataset and evaluation platform for machine-in-the-loop story generation . *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*. 2020. URL: <https://doi.org/10.18653/v1/2020.emnlp-main.525>. (date of access: 16.06.2026).
- 18) Gao Y. Xiong Y. Gao X. Et al. Retrieval-augmented generation for large language models: a survey. *Arxiv*. 2023. URL: <https://doi.org/10.48550/arxiv.2312.10997>. (date of access: 16.06.2026).
- 19) Udovenko S. G., Chala L. E., Kushvid E. S. A method for comparing text and graphic fragments in electronic documents using a hybrid criterion. *Bionics of Intelligence*. 2019. Vol. 1, no. 92. P. 71–76. URL: [https://doi.org/10.30837/bi.2019.1\(92\).12](https://doi.org/10.30837/bi.2019.1(92).12) (date of access: 16.06.2026).
- 20) Filatov V., Zolotukhin O., Kudryavtseva M. Intellectual data analysis in relational information and analytical systems. *Innovative technologies and scientific solutions for industries*. 2025. No. 4(34). P. 101–111. URL: <https://doi.org/10.30837/2522-9818.2025.4.101> (date of access: 16.06.2026).
- 21) Bodiansky Y., Chala O. Intelligent information technology for rapid classification under conditions of overlapping classes. *Bulletin of National Technical University "KhPI". series: system analysis, control and information technologies*. 2024. No. 2 (12). P. 108–112. URL: <https://doi.org/10.20998/2079-0023.2024.02.17> (date of access: 16.06.2026).