

## ВЗАЄМОДІЯ З LLM-АСИСТЕНТАМИ ДЛЯ АВТОМАТИЗАЦІЇ ЛАНДШАФТНОЇ ГЕНЕРАЦІЇ В ІГРОВИХ РУШІЯХ

Вінницький національний технічний університет

### Анотація

*У роботі досліджено принципи проєктування людино-машинних інтерфейсів для взаємодії з великими мовними моделями (LLM) у процесі процедурної генерації ландшафтів в ігрових рушіях. Проаналізовано перехід від ручного скульптингу та нодових систем до керування середовищем через природномовні запити. Визначено ключові UI/UX патерни, що знижують когнітивне навантаження розробника та забезпечують швидке ітеративне прототипування ігрового світу.*

**Ключові слова:** процедурна генерація, ландшафти, великі мовні моделі, LLM, GameDev, штучний інтелект, людино-машинна взаємодія, ігрові рушії.

### Abstract

*The paper explores the principles of designing human-computer interfaces for interacting with large language models (LLMs) in the process of procedural landscape generation in game engines. The transition from manual sculpting and node-based systems to controlling the environment via natural language prompts is analyzed. Key UI/UX patterns that reduce developer cognitive load and enable fast iterative prototyping of the game world are identified.*

**Keywords:** procedural generation, landscapes, large language models, LLM, GameDev, artificial intelligence, human-computer interaction, game engines.

Створення просторих і високодеталізованих віртуальних середовищ є однією з ключових вимог сучасної ігрової індустрії (GameDev). Дотепер генерація ландшафтів здебільшого спиралася на ручне моделювання (скульптинг) або застосування комплексних систем на основі нодів (як-от Houdini чи World Machine). Хоча такі інструменти дають змогу точно контролювати кінцевий результат, вони відрізняються крутою кривою навчання, вимагають розуміння складних математичних алгоритмів і суттєво перевантажують розробника когнітивно. Еволюція великих мовних моделей (LLM) дала поштовх до появи в галузі людино-машинної взаємодії (HCI) інноваційного підходу — так званого «діалогового конструювання світів» (Conversational World-Building).

Впровадження LLM безпосередньо в середовище ігрових рушіїв змінює статус левел-дизайнера: відтепер він виконує роль артдиректора, делегуючи технічну рутину штучному інтелекту. Управління генерацією відбувається за допомогою команд природною мовою (промптів). Наприклад, користувач може задати запит: «Створи гірський хребет із широкою річковою долиною і густим хвойним лісом на північних схилах». У цьому процесі ШІ виконує функцію інтелектуального посередника: він аналізує неструктурований текстовий ввід, розпізнає головні семантичні елементи (масштаб, особливості рельєфу, типи біомів) та перетворює їх на конкретні числові параметри для алгоритмів процедурної генерації ігрового рушія.

Ефективна реалізація подібної взаємодії потребує розробки комплексної багаторівневої архітектури, у якій велика мовна модель виступає важливим, але не єдиним компонентом загального конвеєра (рис. 1).

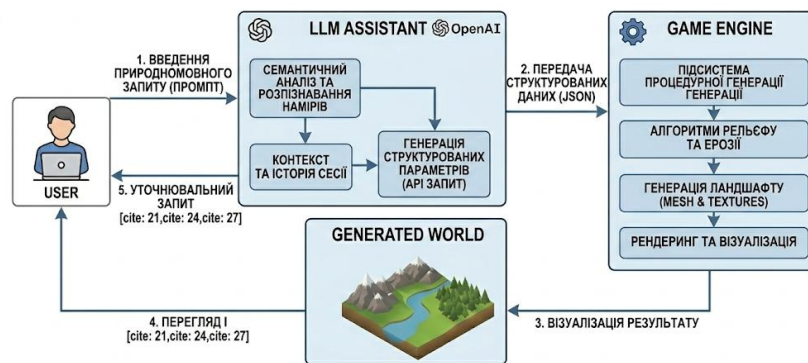


Рисунок 1 – Структурна схема взаємодії користувача з ігровим рушієм через LLM-асистента

Відповідно до наведеної схеми, текстовий запит розробника спершу піддається семантичному аналізу на стороні LLM. При цьому обов'язково враховується контекст попередніх повідомлень поточної сесії. Після обробки мовна модель формує структурований масив даних (зокрема, у форматі JSON), який за допомогою API спрямовується до модуля процедурної генерації всередині ігрового рушія. Як тільки рушій візуалізує створений ландшафт, користувач має змогу надіслати новий запит для уточнення деталей, таким чином підтримуючи безперервний цикл ітеративного дизайну.

Основна складність у розробці таких інтерфейсів полягає у збереженні контролю над творчим процесом та його безперервності. Оскільки ідеальний результат майже ніколи не досягається з першої спроби, UI/UX рішення для LLM-асистента повинні спиратися на збереження історії контексту та інструменти локального коригування. Наприклад, на запит «зробити гори більш пологими» система має адекватно відреагувати, змінивши виключно кут нахилу схилів у поточній конфігурації, не зачіпаючи вже сформовані русла річок чи розміщені ігрові об'єкти. Реалізувати таку вибірковість дозволяє метод генерації ландшафту у вигляді незалежних параметричних шарів.

Ще одним вагомим аспектом є мінімізація затримок (latency), що необхідно для підтримання стану психологічного «потoku» (flow) розробника. Оскільки створення високополігональних поверхонь із розрахунком водної та термічної ерозії потребує значних обчислювальних потужностей, інтерфейс має використовувати підходи прогресивного рендерингу для уникнення тривалих зависань. Практично це означає миттєве виведення низькополігонального (low-poly) чорнового варіанта сцени, щоб користувач міг одразу перевірити, чи правильно ШІ зрозумів промпт. Лише після схвалення ескізу система запускає фонові асинхронні процеси для додавання дрібних деталей, текстурування та ресурсоемних фізичних симуляцій.

Підсумовуючи, можна стверджувати, що залучення LLM до процесів процедурної генерації значно спрощує вхід у професію Level Design та прискорює створення первинних прототипів локацій. Успішність такого підходу визначається не лише технічними характеристиками мовної моделі, а й якістю ергономічних рішень інтерфейсу. Правильно спроектована система має гарантувати високу гнучкість, надійне збереження контексту та плавну інтеграцію між текстовим керуванням, візуалізацією в реальному часі та класичним ручним редагуванням згенерованого світу.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Буров О. Ю. Людино-машинна взаємодія: ергономічні та когнітивні аспекти. Київ: Знання, 2021. 312 с.
2. Shaker N., Togelius J., Nelson M. J. Procedural Content Generation in Games. Springer, 2016. 336 p.
3. Smith J., Doe A. Conversational Interfaces for Procedural Generation in Modern Game Engines. *Journal of Interactive Media and HCI*, 2024, Vol. 18(1), pp. 75-92.
4. Ван Дер Бург А. Оптимізація рендерингу та процедурної генерації ландшафтів у реальному часі. *Сучасні інформаційні технології*, 2023, № 4, С. 45-53.

***Коваль Олександр Дмитрович*** - студент групи 2ПІ-25Б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, email:**oleksandr.kov.dm@gmail.com**

Науковий керівник: ***Ткаченко Олександр Миколайович***— кандидат технічних наук, доцент, Вінницький національний технічний університет, місто Вінниця, email:**ant@vntu.edu.ua**

**Koval Oleksandr Dmytrovych** – student of group 2PI-25B, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email:**oleksandr.kov.dm@gmail.com**

Scientific supervisor: **Oleksandr Mykolaiovych Tkachenko** — Candidate of Technical Sciences, Associate Professor, Vinnytsia National Technical University, Vinnytsia, email:**ant@vntu.edu.ua**