Складанюк Олексій, аспірант кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, Україна, ORCID: 0009-0003-3180-0326

Skladaniuk Oleksii, Postgraduate Student of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, Ukraine, ORCID: 0009-0003-3180-0326

ІНТЕЛЕКТУАЛЬНІ МЕТОДИ РЕДАГУВАННЯ ІГРОВИХ СЦЕН У СЕРЕДОВИЩАХ РОЗРОБКИ ВІДЕОІГОР

INTELLIGENT METHODS FOR EDITING GAME SCENES IN VIDEO GAME DEVELOPMENT ENVIRONMENTS

Анотація

Мета роботи. Метою статті є дослідження сучасних інтелектуальних методів редагування ігрових сцен у середовищах розробки відеоігор та визначення їх ролі в автоматизації процесів створення, модифікації й оптимізації ігрового контенту. Особливу увагу приділено підходам, що дозволяють підвищити ефективність роботи з великою кількістю об’єктів сцени, покращити продуктивність інтерактивних застосунків і зменшити обсяг ручного налаштування елементів віртуального середовища.

Методологія. У дослідженні використано методи теоретичного аналізу, узагальнення, систематизації та порівняння сучасних підходів до редагування ігрових сцен. Проаналізовано можливості застосування ієрархічних структур об’єктів, систем рівнів деталізації (LOD), процедурної генерації контенту, інтерактивних редакторів сцен, а також алгоритмів штучного інтелекту, машинного навчання, навчання з підкріпленням і генеративних моделей. Окремо розглянуто функціональні можливості

сучасних середовищ розробки, зокрема Unity та Unreal Engine, а також використання структурованих форматів даних, зокрема JSON, для збереження, передачі й обробки інформації про об’єкти сцени.

Наукова новизна. Наукова новизна роботи полягає у систематизації інтелектуальних підходів до редагування ігрових сцен як окремого напряму в межах сучасних GameDev-технологій. У статті узагальнено взаємозв’язок між алгоритмами штучного інтелекту, засобами процедурної генерації, методами автоматизованого налаштування параметрів сцени та інструментами адаптивної модифікації ігрового середовища. На відміну від підходів, що зосереджуються лише на генерації контенту, у роботі акцентовано увагу саме на інтелектуальному редагуванні вже сформованих сцен, їх оптимізації та підтримці дизайнерських рішень у процесі розробки відеоігор.

Висновки. Встановлено, що використання інтелектуальних методів редагування ігрових сцен дозволяє підвищити ефективність розробки, скоротити витрати часу на ручне налаштування, оптимізувати використання обчислювальних ресурсів і покращити якість кінцевого продукту. Застосування алгоритмів штучного інтелекту та машинного навчання відкриває можливості для автоматизованого розміщення об’єктів, адаптації поведінки елементів у реальному часі та створення більш варіативних і реалістичних ігрових середовищ. Практичне значення отриманих результатів полягає в можливості їх використання під час розробки сучасних відеоігор, зокрема у середовищах, орієнтованих на масштабоване та продуктивне редагування сцен.

Ключові слова: інтелектуальні методи, редагування ігрових сцен, штучний інтелект, машинне навчання, процедурна генерація контенту, Unity, Unreal Engine, інтерактивні середовища.

Abstract

Purpose. The purpose of the article is to study modern intelligent methods for editing game scenes in video game development environments and to determine their role in automating the processes of creating, modifying, and optimizing game content. Particular attention is paid to approaches that improve the efficiency of working with a large number of scene objects, enhance the performance of interactive applications, and reduce the amount of manual configuration required for virtual environment elements.

Methodology. The study is based on the methods of theoretical analysis, generalization, systematization, and comparison of modern approaches to game scene editing. The possibilities of applying hierarchical object structures, level-of-detail (LOD) systems, procedural content generation, interactive scene editors, as well as artificial intelligence algorithms, machine learning, reinforcement learning, and generative models are analyzed. Special attention is also paid to the capabilities of modern development environments, in particular Unity and Unreal Engine, and to the use of structured data formats, especially JSON, for storing, transmitting, and processing information about scene objects.

Scientific novelty. The scientific novelty of the study lies in the systematization of intelligent approaches to game scene editing as a separate direction within modern GameDev technologies. The article generalizes the relationship between artificial intelligence algorithms, procedural generation tools, automated scene parameter adjustment methods, and adaptive environment modification instruments. Unlike approaches focused only on content generation, the paper emphasizes the intelligent editing of already created scenes, their optimization, and support for designer decisions during the video game development process.

Conclusions. It has been established that the use of intelligent methods for editing game scenes increases development efficiency, reduces the time required

for manual configuration, optimizes the use of computational resources, and improves the quality of the final product. The application of artificial intelligence and machine learning algorithms opens up opportunities for automated object placement, real-time adaptation of game element behavior, and the creation of more diverse and realistic game environments. The practical significance of the obtained results lies in their applicability to the development of modern video games, especially in environments focused on scalable and high-performance scene editing.

Key words: intelligent methods, game scene editing, artificial intelligence, machine learning, procedural content generation, Unity, Unreal Engine, interactive environments.

Постановка проблеми. Сучасна розробка відеоігор пов’язана з постійним ускладненням ігрових світів, збільшенням кількості інтерактивних об’єктів, зростанням вимог до візуальної якості та необхідністю забезпечення стабільної продуктивності в режимі реального часу. За таких умов ручне редагування сцен стає дедалі менш ефективним, оскільки потребує значних часових витрат, великої кількості повторюваних операцій та ускладнює швидке внесення змін до структури ігрового середовища (Shaker, Togelius, Nelson, 2016; Zhang, 2024).

Ігрова сцена є багатокомпонентною структурою, яка поєднує геометрію простору, освітлення, матеріали, фізичні властивості об’єктів, поведінкові модулі, маршрути навігації, анімаційні елементи та логіку взаємодії з гравцем. Через це навіть локальні зміни в композиції сцени можуть впливати на прохідність рівня, баланс, навігацію, продуктивність рендерингу та загальне сприйняття ігрового простору. Саме тому зростає потреба в методах, які дозволяють не лише редагувати сцену, а й аналізувати

її стан, виявляти проблеми та пропонувати доцільні варіанти оптимізації (Summerville et al., 2018; Green et al., 2022, с. 638–645).

Інтенсивний розвиток штучного інтелекту, процедурної генерації, генеративних моделей та ко-креативних інструментів сприяв переходу від простого автоматичного створення окремих елементів до інтелектуальної підтримки повного процесу редагування середовища. У сучасних умовах інтелектуальні методи доцільно розглядати як важливий складник GameDev-процесів, що дає змогу скорочувати обсяг рутинної роботи, підвищувати якість прототипування та зберігати узгодженість складних сцен упродовж усього циклу розробки (Maleki, Zhao, Shaker, 2024; Ткаченко, Мамаєв, 2024, с. 97–109). *Аналіз останніх досліджень і публікацій.* Проблема автоматизації створення ігрового контенту найповніше розвинулася в межах procedural content generation, де контент розглядається як результат алгоритмічного формування рівнів, просторів, маршрутів, об’єктів і сценарних структур. Такий підхід став теоретичною основою для подальшого переходу до інструментів, що не лише генерують нові елементи, а й підтримують редагування вже створених сцен, зокрема шляхом перебудови простору, варіювання його наповнення та контролю узгодженості структури (Shaker, Togelius, Nelson, 2016).

Подальший розвиток цього напряму пов’язаний із використанням машинного навчання, яке дозволяє системам виявляти патерни в уже створених рівнях і застосовувати їх для генерації або модифікації нового контенту. Це дає можливість будувати рішення не лише на формальних правилах, а й на досвіді попереднього дизайну, що особливо важливо для інтелектуального редагування сцен, де необхідно враховувати жанрові закономірності, композиційні принципи та логіку просторової організації (Summerville et al., 2018; Liu et al., 2021).

Важливе значення для розвитку редакторів нового покоління мають підходи, у яких людина і система взаємодіють у спільному циклі проєктування. У межах таких рішень алгоритм не обмежується технічним виконанням команди, а аналізує стан середовища, реагує на дії користувача та пропонує варіанти перебудови окремих фрагментів сцени. Це дозволяє зберігати творчий контроль за дизайнером і водночас скорочувати обсяг ручної роботи, пов'язаної з перевіркою проходів, логіки розміщення або просторового балансу (Smith, Whitehead, Mateas, 2010; Smith, Whitehead, Mateas, 2011).

Значну увагу в новітніх дослідженнях приділено також поєднанню процедурної генерації з великими мовними моделями та іншими генеративними інструментами. Така інтеграція розширює межі автоматизації, оскільки дозволяє формувати не лише структуру сцени, а й правила її побудови, описи логіки середовища, рекомендації для дизайнера та сценарії адаптивної зміни простору. Унаслідок цього інтелектуальне редагування починає розглядатися не як окремий модуль, а як частина ширшої системи підтримки розробки ігрового контенту (Maleki, Zhao, Shaker, 2024).

В українському науковому просторі також зростає інтерес до застосування штучного інтелекту в розробці ігор. Наявні дослідження підтверджують доцільність використання інтелектуальних алгоритмів для автоматизації створення контенту, адаптації поведінки елементів і оптимізації процесу розробки, однак проблема інтелектуального редагування сцени як цілісного об'єкта потребує подальшої систематизації. (Трофименко та ін., 2025, с. 109–119; Обидало, Пономарьова, 2025, с. 475–477; Ткаченко, Мамаєв, 2024, с. 97–109).

Отже, сучасні дослідження створюють достатню наукову основу для розгляду інтелектуального редагування ігрових сцен як самостійного

напряму, що поєднує procedural generation, машинне навчання, генеративні моделі, оптимізаційні підходи та ко-креативні засоби взаємодії з дизайнером. Разом із тим саме комплексне поєднання цих компонентів у межах редакторського процесу залишається недостатньо систематизованим, що й зумовлює доцільність подальшого дослідження.

Мета статті. Метою статті є дослідження інтелектуальних методів редагування ігрових сцен у середовищах розробки відеоігор, систематизація сучасних підходів до автоматизації аналізу та модифікації сцен, а також визначення переваг, обмежень і перспектив застосування алгоритмів штучного інтелекту в процесі створення ігрового контенту.

Виклад основного матеріалу дослідження. Ігрову сцену доцільно розглядати як багаторівневу цифрову структуру, що включає просторові, функціональні, візуальні та поведінкові компоненти. У сучасних середовищах розробки вона поєднує 3D-об’єкти, матеріали, освітлення, камери, навігаційні вузли, фізичні тіла, тригери, скриптові компоненти, елементи інтерфейсу та аудіовізуальні ефекти. Саме така складна структура зумовлює необхідність переходу від звичайного ручного редагування до інтелектуально підтримуваних засобів, здатних враховувати взаємозалежність усіх елементів сцени (Zhang, 2024; Shaker, Togelius, Nelson, 2016).

Традиційне редагування сцени передбачає ручне розміщення об’єктів, налаштування їх параметрів, організацію ієрархії та перевірку коректності взаємодії. У масштабних проєктах це істотно підвищує трудомісткість розробки та збільшує ризик структурних помилок, перевантаження рендерингу й порушення композиційної цілісності сцени. (Ткаченко, Мамаєв, 2024, с. 97–109).

Інтелектуальні методи редагування ігрових сцен можна визначити як сукупність алгоритмів і програмних засобів, що забезпечують аналіз

поточного стану сцени, автоматичне виявлення проблем, формування рекомендацій і часткову або повну автоматизацію змін у структурі середовища. На відміну від звичайних інструментів редактора, такі методи не обмежуються прямим виконанням команд користувача, а використовують правила, евристики, моделі машинного навчання або генеративні підходи для прийняття рішень щодо розміщення, модифікації чи оптимізації елементів сцени (Summerville et al., 2018; Green et al., 2022, с. 638–645).

У межах цього дослідження інтелектуальні методи редагування сцен доцільно поділити на аналітичні, генеративні, оптимізаційні, адаптивні та змішані. Узагальнення основних груп інтелектуальних методів редагування ігрових сцен подано у таблиці 1.

Аналітичні методи орієнтовані на оцінювання поточного стану сцени та автоматичне виявлення технічних і структурних проблем, зокрема конфліктів колізій, перевантажених зон, помилок навігації чи нераціонального розподілу об’єктів. Їх перевага полягає у швидкому контролі якості сцени та зменшенні обсягу ручної перевірки під час розробки.

Генеративні методи забезпечують створення нових елементів сцени або перебудову вже наявних фрагментів. У простішому варіанті це можуть бути rule-based системи, що працюють за заданими шаблонами або композиційними правилами. У складніших реалізаціях використовуються підходи procedural content generation, завдяки яким середовище або його частина формується на основі параметрів, умов і композиційних обмежень. Це дає змогу швидко створювати варіативні локації, заповнювати порожні зони та підтримувати різноманітність простору без повного ручного опрацювання кожного елемента (Shaker, Togelius, Nelson, 2016; Maleki, Zhao, Shaker, 2024).

Таблиця 1

Класифікація інтелектуальних методів редагування ігрових сцен

Тип методу	Основна функція	Практичне призначення	Переваги	Обмеження
Аналітичні	Аналіз структури сцени та виявлення проблем	Пошук конфліктів колізій, перевантажених зон, помилок навігації, порушень логіки розміщення	Швидка перевірка, зменшення кількості ручних помилок	Не завжди пропонують готовий спосіб виправлення
Генеративні	Автоматичне створення або добудова елементів сцени	Заповнення простору, варіювання локацій, створення альтернативних конфігурацій	Прискорюють прототипування, підвищують варіативність	Можуть давати шаблонні результати
Методи машинного навчання	Виявлення патернів і формування рекомендацій	Підтримка дизайнер-ських рішень, адаптація сцени до жанру або стилю	Ураховують попередній досвід проєктування	Залежні від якості навчальних даних
Оптимізаційні	Пошук найкращої конфігурації за кількома критеріями	Балансування продуктивності, складності, щільності об’єктів і прохідності	Дозволяють покращувати сцену комплексно	Висока обчислювальна складність
Змішані (mixed-initiative)	Спільна робота дизайнера та інтелектуальної системи	Підтримка творчого процесу, пояснювані рекомендації, інтерактивна перебудова сцени	Зберігають творчий контроль людини	Потребують продуманого інтерфейсу та пояснюваності

Методи машинного навчання дозволяють системі вивчати закономірності розміщення об’єктів, структури проходів, логіку зонування та інші особливості вже створених ігрових сцен. Завдяки цьому редактор може не лише перевіряти структуру середовища, а й пропонувати рекомендації, наближені до реальної дизайнерської практики. Особливу цінність тут мають підходи PCGML, які поєднують генерацію й аналіз контенту на основі даних, а не лише на формальних правилах (Summerville et al., 2018; Liu et al., 2021).

Генеративні нейромережеві моделі відкривають додаткові можливості для створення альтернативних конфігурацій простору. Їх застосування дає змогу добудовувати фрагменти сцени, формувати нові варіанти розташування об’єктів або відтворювати стилістично подібні середовища на основі вже існуючих прикладів. Це особливо корисно в ситуаціях, коли розробникові потрібно швидко отримати кілька альтернативних варіантів сцени для подальшого відбору й доопрацювання (Giacomello, Lanzi, Loiacono, 2019).

Оптимізаційні методи спрямовані на пошук найкращої конфігурації сцени за множиною критеріїв. Такими критеріями можуть бути продуктивність, прохідність, складність рівня, композиційна рівновага, щільність об’єктів або зручність взаємодії. У подібних задачах часто застосовують евристичні та еволюційні алгоритми, здатні знаходити прийнятні рішення в умовах багатокритеріальності й великої кількості можливих конфігурацій. Це дозволяє розглядати редактор не лише як інструмент побудови сцени, а і як засіб її цілеспрямованого вдосконалення.

Для формалізації процесу оцінювання ігрової сцени доцільно використовувати інтегральний показник якості, який узагальнює основні критерії побудови та функціонування середовища. Такий підхід дозволяє розглядати сцену як об’єкт багатокритеріальної оптимізації:

$$Q = w_1 \text{Ц} C + w_2 N + w_3 P + w_4 \text{w} V \quad (1)$$

де Q — інтегральна оцінка якості сцени; C — композиційна узгодженість; N— навігаційна зручність; P — продуктивність; V — візуальна цілісність; w1, w2, w3, w4 — вагові коефіцієнти відповідних критеріїв.

Значення вагових коефіцієнтів w1 , w2 , w3 , w4 доцільно визначати не як фіксовані константи, а як адаптивні параметри, що залежать від жанру гри, типу сцени та пріоритетів проєкту. Для цього може бути використано метод зваженої суми критеріїв із попередньою нормалізацією показників, після чого ваги задаються експертно, емпірично або на основі сценарного профілю сцени.

$$\sum_{i=1}^{4} w_i = 1,\ w_i \geq 0 \quad (2)$$

Умова (2) забезпечує коректність багатокритеріального оцінювання та дає змогу інтерпретувати вагові коефіцієнти як відносну важливість окремих критеріїв. Адаптація вагових коефіцієнтів може виконуватися за жанрово-орієнтованим принципом. Наприклад, для ігор з відкритим світом доцільно підвищувати вагу навігаційної зв’язності та продуктивності, для сюжетно-орієнтованих сцен — композиційної узгодженості та візуальної цілісності, а для динамічних бойових арен — прохідності, щільності функціональних об’єктів і швидкості реакції системи на зміни середовища. Таким чином, вагові коефіцієнти виступають не універсальними сталими величинами, а параметрами конфігурації інтелектуального редактора, що можуть бути адаптовані до конкретного сценарію використання. Приклад адаптації вагових коефіцієнтів наведено у таблиці 2.

Таблиця 2

Приклад адаптації вагових коефіцієнтів залежно від типу сцени

Тип сцени	w_1 композиція	w_2 навігація	w_3 продуктивність	w_4 візуальна цілісність
Відкритий світ	0.20	0.35	0.30	0.15
Сюжетна локація	0.30	0.20	0.15	0.35
Бойова арена	0.20	0.30	0.30	0.20

Наведені значення є прикладом сценарного налаштування ваг і можуть змінюватися залежно від жанру гри, типу сцени та цілей проєкту.

Найбільш перспективними для практичної інтеграції є змішані, або mixed-initiative, системи. У таких системах дизайнер задає загальний задум і контролює ключові рішення, а інтелектуальний модуль пропонує варіанти, реагує на дії користувача та допомагає підтримувати цілісність сцени. Перевага цього підходу полягає в поєднанні автоматизації та творчої свободи, що особливо важливо в GameDev-середовищах, де остаточна якість простору залежить не лише від технічної коректності, а й від художнього бачення та ігрового досвіду (Smith, Whitehead, Mateas, 2010; Smith, Whitehead, Mateas, 2011; Baldwin, Johnson, Wyeth, 2022).

У практиці розробки інтелектуальні методи можуть реалізовуватися як модулі аналізу, генерації, рекомендації та оптимізації. Така модульність спрощує інтеграцію інтелектуальних рішень у сучасні редактори та робить їх більш гнучкими у використанні. Узагальнену логіку взаємодії основних модулів інтелектуального редагування ігрової сцени подано на рис. 1.

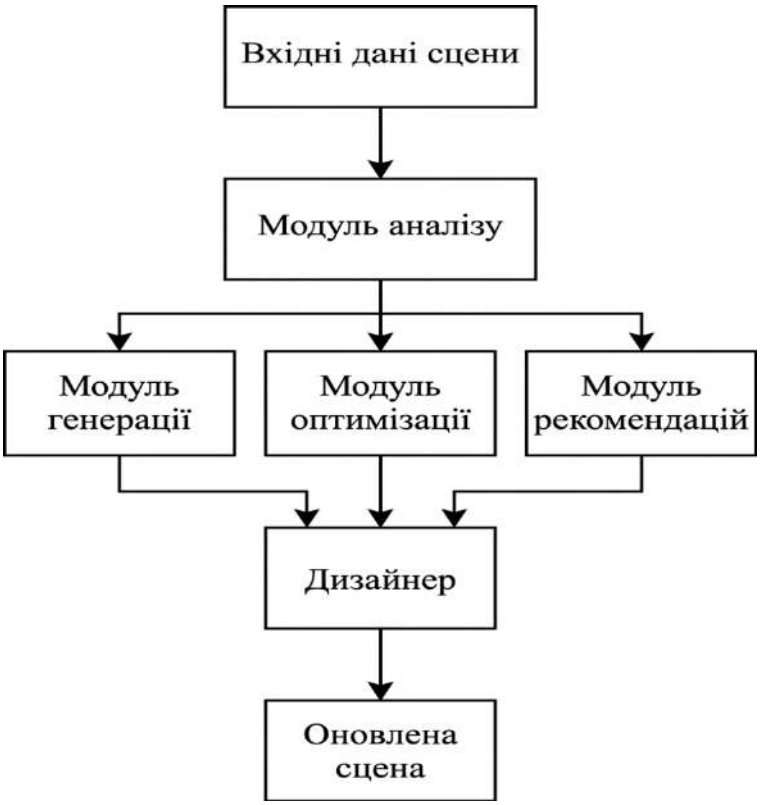


Рис. 1. Узагальнена схема інтелектуального редагування ігрової сцени

Алгоритмічна реалізація інтелектуального редагування сцени може бути подана як послідовність етапів: збір параметрів сцени, формування вектора ознак, аналіз конфліктів і структурних відхилень, генерація допустимих альтернатив конфігурації, багатокритеріальне оцінювання варіантів та вибір рішення з найвищим інтегральним показником якості. У межах модуля аналізу доцільно використовувати перевірку колізій, оцінювання щільності об’єктів, прохідності та навігаційної зв’язності, тоді як модуль рекомендацій може спиратися на rule-based правила, евристичне ранжування або навчання на попередньо розмічених прикладах.

Важливу роль у цьому процесі відіграє структуроване подання даних про сцену. Застосування форматів на зразок JSON дозволяє зберігати конфігурації об’єктів, передавати параметри між підсистемами,

підтримувати шаблони та повторне використання елементів. У результаті інтелектуальне редагування отримує не лише алгоритмічне, а й архітектурне підґрунтя, оскільки ефективність аналітичних і генеративних модулів значною мірою залежить від того, наскільки впорядковано подано інформацію про структуру сцени.

У сучасній практиці розробки відеоігор інтелектуальні методи редагування сцен реалізуються в різних рушіях, серед яких найбільш показовими є Unity та Unreal Engine (Zhang, 2024; Epic Developer Community, 2025). У середовищі Unity такі підходи зазвичай інтегруються через скриптові компоненти, редакторські розширення, системи роботи зі сценами та структуроване подання даних, що забезпечує гнучке налаштування логіки аналізу, модифікації та повторного використання об’єктів сцени (Zhang, 2024). У Unreal Engine аналогічні завдання підтримуються, зокрема, через Procedural Content Generation Framework (PCG), який являє собою вбудований набір засобів для створення процедурного контенту різної складності — від окремих об’єктів і біомів до цілих світів (Epic Developer Community, 2025). Центральним елементом цього підходу є PCG Graph — вузлова графова модель, у якій просторові дані надходять із PCG Component, обробляються через послідовність вузлів, фільтруються, модифікуються та використовуються для розміщення об’єктів у сцені (Epic Developer Community, 2025). Таким чином, Unity демонструє переваги гнучкої скриптової та модульної інтеграції інтелектуальних засобів, тоді як Unreal Engine надає вбудовану графову інфраструктуру для процедурної побудови контенту, що робить доцільним їх порівняльний розгляд у контексті інтелектуального редагування ігрових сцен (Zhang, 2024; Epic Developer Community, 2025).

Порівняльні особливості реалізації інтелектуального редагування сцен у Unity та Unreal Engine наведено у таблиці 3.

Таблиця 3

Порівняльні особливості реалізації інтелектуального редагування сцен у Unity та Unreal Engine

Параметр	Unity	Unreal Engine
Загальний підхід	Скриптова, компонентна та редакторська інтеграція інтелектуальних інструментів	Вбудований Procedural Content Generation Framework (PCG)
Організація процедурної логіки	Кастомні скрипти, редакторські розширення, зовнішні модулі	PCG Graph як вузлова графова модель
Робота з просторовими даними	Залежить від реалізації користувача або використовуваних пакетів	Просторові дані надходять через PCG Component і обробляються системою вузлів
Характер оновлення сцени	Переважно через редакторські або програмні виклики	Модифікація та оновлення результату в межах графової системи
Сильна сторона	Гнучкість і простота інтеграції в різні типи проєктів	Вбудована інфраструктура для масштабної процедурної побудови контенту

Окрему перспективу становить використання навчання з підкріпленням для адаптивної модифікації середовища. У цьому випадку оцінюються не лише статичні характеристики сцени, а й наслідки її змін для проходження, складності, взаємодії або поведінки гравця. Такий підхід відкриває можливість розглядати сцену як адаптивну структуру, здатну змінюватися відповідно до цілей гри або сценаріїв тестування, що є особливо важливим для динамічних ігрових середовищ (Обидало, Пономарьова, 2025, с. 475–477).

Переваги інтелектуального редагування сцен полягають у скороченні тривалості створення контенту, зменшенні кількості рутинних операцій, підвищенні узгодженості середовища та покращенні якості раннього

прототипування. Водночас слід враховувати і певні обмеження: залежність від якості даних, складність формалізації естетичних критеріїв, ризик шаблонності автоматично згенерованих рішень і необхідність збереження творчої ролі людини в проєктуванні. Тому найбільш доцільним підходом є не повна заміна дизайнера, а створення ко-креативних інструментів, що посилюють його можливості в роботі зі складними сценами (Baldwin, Johnson, Wyeth, 2022; Трофименко та ін., 2025, с. 109–119).

Отже, інтелектуальні методи редагування ігрових сцен слід розглядати як комплексний інструментарій, що охоплює аналіз, генерацію, оптимізацію, адаптацію та пояснювану підтримку дизайнерських рішень. Їх інтеграція в сучасні середовища розробки відкриває шлях до створення редакторів нового покоління, здатних одночасно підвищувати ефективність роботи, покращувати якість сцени та зберігати гнучкість творчого процесу.

Висновки. У результаті дослідження встановлено, що інтелектуальні методи редагування ігрових сцен є важливим напрямом розвитку сучасних засобів створення відеоігор. Їх застосування дозволяє поєднати автоматизований аналіз, генерацію, оптимізацію та рекомендаційну підтримку в межах єдиного редакторського процесу, що особливо важливо для складних і багатокомпонентних інтерактивних середовищ (Summerville et al., 2018; Green et al., 2022, с. 638–645). Найбільш ефективними виявляються гібридні підходи, у яких поєднуються rule-based алгоритми, procedural content generation, машинне навчання, генеративні моделі та mixed-initiative взаємодія з користувачем. Саме вони забезпечують баланс між автоматизацією, продуктивністю та збереженням творчого контролю дизайнера над структурою сцени (Smith, Whitehead, Mateas, 2011; Baldwin, Johnson, Wyeth, 2022). Перспективи подальших досліджень полягають у розробленні жанрово-орієнтованих інтелектуальних редакторів сцен, створенні пояснюваних рекомендаційних модулів для сучасних рушіїв,

поглибленні інтеграції генеративних моделей і навчання з підкріпленням, а також у формалізації естетичних критеріїв якості ігрового простору. Важливим напрямом є також подальший розвиток прикладних досліджень у цій сфері з орієнтацією на практичні інструменти автоматизації GameDev-процесів (Ткаченко, Мамаєв, 2024, с. 97–109; Трофименко та ін., 2025, с. 109–119).

Література

1. Smith G., Whitehead J., Mateas M. Tanagra: A Mixed-Initiative Level Design Tool. *Proceedings of the Fifth International Conference on the Foundations of Digital Games.* New York : ACM, 2010. P. 209–216. DOI: 10.1145/1822348.1822376

2. Smith G., Whitehead J., Mateas M. Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. *IEEE Transactions on Computational Intelligence and AI in Games*. 2011. Vol. 3, No. 3. P. 201–215.

3. Shaker N., Togelius J., Nelson M. J. *Procedural Content Generation in Games*. Cham : Springer, 2016. 238 p.

4. Summerville A., Snodgrass S., Guzdial M., Holmgård C., Hoover A. K., Isaksen A., Nealen A., Togelius J. Procedural Content Generation via Machine Learning (PCGML). *IEEE Transactions on Games. 2018*. Vol. 10, No. 3. P. 257–270. DOI: 10.1109/TG.2018.2846639.

5. Liu J., Snodgrass S., Khalifa A., Risi S., Yannakakis G. N., Togelius J. Deep learning for procedural content generation. *Neural Computing and Applications. 2021.* Vol. 33. P. 19–37. DOI: 10.1007/s00521-021-06103-8.

6. Baldwin A., Johnson D., Wyeth P. Evaluating mixed-initiative procedural level design tools using a triple-blind mixed-method user study. *IEEE Transactions on Games. 2022.* Vol. 14, No. 2. P. 228–239. DOI: 10.1109/TG.2021.3086215.

7. Giacomello E., Lanzi P. L., Loiacono D. DOOM Level Generation using Generative Adversarial Networks. *2019 IEEE Conference on Games (CoG).* London, 2019. DOI: 10.1109/CIG.2019.8848021.

8. Green M. C., Khalifa A., Barros G. A. B., Neff M., Togelius J. Let's Make Games Together: Explainability in Mixed-Initiative Co-Creative Game Design. *2022 IEEE Conference on Games (CoG)*. 2022. P. 638–645. DOI: 10.1109/CoG51982.2022.9893636.

9. Maleki H., Zhao J., Shaker N. Procedural Content Generation in Games: A Survey with Insights on Emerging LLM-Based Methods. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. 2024. Vol. 20, No. 1. DOI: 10.1609/aiide.v20i1.31877.

10. Zhang J. Research on Virtual Scene Design Methods for Unity3D Games. *Proceedings of the 3rd International Conference on Art Design and Digital Technology*. 2024. DOI: 10.4108/eai.24-5-2024.2350033.

11. Procedural Content Generation Framework in Unreal Engine. Epic Developer Community. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/procedural-content-generation--framework-in-unreal-engine

12. Трофименко О. Г., Дика А. В., Лобода Ю. В., Толокнов А. М., Бондаренко М. В. Алгоритми штучного інтелекту в розробці ігор. *Кібербезпека:* освіта, наука, техніка. 2025. Т. 3, № 27. С. 109–119. DOI: 10.28925/2663-4023.2025.27.701.

13. Обидало О. С., Пономарьова С. В. Використання штучного інтелекту в ігрових застосунках. У кн.: *Радіоелектроніка та молодь у XXI столітті : матеріали 29-го Міжнародного молодіжного форуму, 16–19 квітня 2025 р.* Харків : ХНУРЕ, 2025. Т. 6. С. 475–477.

14. Ткаченко О., Мамаєв А. Використання штучного інтелекту під час формування простору комп'ютерних ігор. *Цифрова платформа:*

інформаційні технології в соціокультурній сфері. 2024. Т. 7, № 1. С. 97–109. DOI: 10.31866/2617-796X.7.1.2024.307012.

References

1. Smith, G., Whitehead, J., & Mateas, M. (2010). Tanagra: A mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (pp. 209–216). New York: ACM. https://doi.org/10.1145/1822348.1822376

2. Smith, G., Whitehead, J., & Mateas, M. (2011). Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3), 201–215.

3. Shaker, N., Togelius, J., & Nelson, M. J. (2016). *Procedural content generation in games.* Cham: Springer.

4. Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A., & Togelius, J. (2018). Procedural content generation via machine learning (PCGML). *IEEE Transactions on Games*, 10(3), 257–270. https://doi.org/10.1109/TG.2018.2846639

5. Liu, J., Snodgrass, S., Khalifa, A., Risi, S., Yannakakis, G. N., & Togelius, J. (2021). Deep learning for procedural content generation. *Neural Computing and Applications*, 33, 19–37. https://doi.org/10.1007/s00521-021-06103-8

6. Baldwin, A., Johnson, D., & Wyeth, P. (2022). Evaluating mixed-initiative procedural level design tools using a triple-blind mixed-method user study. *IEEE Transactions on Games, 14(2)*, 228–239. https://doi.org/10.1109/TG.2021.3086215

7. Giacomello, E., Lanzi, P. L., & Loiacono, D. (2019). DOOM level generation using generative adversarial networks. In 2019 *IEEE Conference on Games (CoG)*. https://doi.org/10.1109/CIG.2019.8848021

8. Green, M. C., Khalifa, A., Barros, G. A. B., Neff, M., & Togelius, J. (2022). Let's make games together: Explainability in mixed-initiative co-creative game design. In 2022 *IEEE Conference on Games (CoG)* (pp. 638–645). https://doi.org/10.1109/CoG51982.2022.9893636

9. Maleki, H., Zhao, J., & Shaker, N. (2024). Procedural content generation in games: A survey with insights on emerging LLM-based methods. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 20(1)*. https://doi.org/10.1609/aiide.v20i1.31877

10. Zhang, J. (2024). Research on virtual scene design methods for Unity3D games. *In Proceedings of the 3rd International Conference on Art Design and Digital Technology*. https://doi.org/10.4108/eai.24-5-2024.2350033

11. Epic Developer Community. (2025). Procedural Content Generation Framework in Unreal Engine. Retrieved from https://dev.epicgames.com/documentation/en-us/unreal-engine/procedural-content-generation--framework-in-unreal-engine

12. Trofymenko, O. H., Dyka, A. V., Loboda, Yu. V., Toloknov, A. M., & Bondarenko, M. V. (2025). Alhorytmy shtuchnoho intelektu v rozrobtsi ihor [Artificial intelligence algorithms in game development]. *Kiberbezpeka: osvita, nauka, tekhnika, 3(27),* 109–119. https://doi.org/10.28925/2663-4023.2025.27.701

13. Obydalo, O. S., & Ponomarova, S. V. (2025). Vykorystannia shtuchnoho intelektu v ihrovykh zastosunkakh [Use of artificial intelligence in gaming applications]. In *Radioelektronika ta molod u XXI stolitti: materialy 29-ho Mizhnarodnoho molodizhnoho forumu* (Vol. 6, pp. 475–477). Kharkiv: KhNURE

14. Tkachenko, O., & Mamaiev, A. (2024). Vykorystannia shtuchnoho intelektu pid chas formuvannia prostoru kompiuternykh ihor [The use of artificial intelligence in shaping the space of computer games]. *Tsyfrova platforma: informatsiini tekhnolohii v sotsiokulturnii sferi*, 7(1), 97–109. https://doi.org/10.31866/2617-796X.7.1.2024.307012