

УДК 004.421:519.17

DOI [https://doi.org/10.15589/znp2025.4\(502\).37](https://doi.org/10.15589/znp2025.4(502).37)

CLASSIFICATION OF MAZE GENERATION ALGORITHMS IN GAME DEVELOPMENT

КЛАСИФІКАЦІЯ АЛГОРИТМІВ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ЛАБІРИНТІВ
ДЛЯ ЗАСТОСУВАННЯ В РОЗРОБЦІ ІГОР**Olena G. Trofymenko¹**

trofymenko@onua.edu.ua

ORCID: 0000-0001-7626-0886

Olexander V. Zadereyko¹

zadereyko@onua.edu.ua

ORCID: 0000-0003-0497-9861

Ihor L. Mykhelev²

igor.michelev@nuos.edu.ua

ORCID: 0000-0001-9579-6547

Mykhailo V. Vorona²

mykhailo.vorona@nuos.edu.ua

ORCID: 0000-0003-4288-0096

Alona U. Pavlenko²

alona.pavlenko@nuos.edu.ua

ORCID: 0000-0002-3776-8377

О. Г. Трофименко¹,

канд. техн. наук, доцент

О. В. Задерейко¹,

канд. техн. наук, доцент

І. Л. Михелєв²,

канд. техн. наук, доцент

М. В. Ворона²,

доктор філософії

А. Ю. Павленко²,

сташий викладач

¹ National University "Odesa Law Academy", Odesa² Admiral Makarov National University of Shipbuilding, Mykolaiv¹ Національний університет «Одеська юридична академія», м. Одеса² Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв

Abstract. The article considers the problem of classifying algorithms for procedural generation of labyrinths in the context of modern video game development. Labyrinths, as a component of the game environment, perform not only a navigation function, but also form aesthetic, cognitive and gameplay dynamics. Given the growing popularity of procedural generation, the issue of systematizing algorithms according to their topological, morphological and navigational characteristics is relevant, which allows ensuring adaptability, replayability and genre correspondence of game levels. The *purpose* of the study is to form a systematic classification of algorithms for procedural generation of labyrinths according to their structural properties, to compare their productivity, difficulty of passage and navigational efficiency, as well as to develop practical recommendations for their application in different genres of video games. The research *methodology* is based on the use of general scientific methods: system analysis, comparison, synthesis, abduction, idealization, analogy, as well as logical and historical approaches. The *results* of the study offer a systematic analysis and classification of maze generation algorithms into four groups: ideal, non-ideal, cave-like, and hybrid. For each group, characteristic topological features, morphological features, and operational advantages are identified. The *scientific novelty* lies in the first proposed systematic classification of maze generation algorithms by topological criteria, which considers not only structural properties, but also navigational and cognitive aspects. The authors also outlined the areas of application of the algorithms depending on genre requirements, technical limitations, and user expectations, which allows integrating these algorithms into game design based on substantiated criteria. The *practical importance* of the study lies in the possibility of using the results to develop adaptive game levels, educational platforms, simulation systems, and educational programming tools. The proposed recommendations contribute to optimizing maze generation, increasing replayability, improving user experience, and standardizing approaches to procedural generation in the video game industry.

Key words: algorithms; optimality of algorithms; graphs; maze generation; game development; maze.

Анотація. У статті розглянуто проблему класифікації алгоритмів процедурної генерації лабіринтів у контексті сучасної розробки відеоігор. Лабіринти, як складова ігрового середовища, виконують не лише навігаційну функцію, а й формують естетичну, когнітивну та геймплейну динаміку. З огляду на ріст популярності процедурної генерації, актуальним є питання систематизації алгоритмів за їхніми топологічними, морфологічними та навігаційними характеристиками, що дозволяє забезпечити адаптивність, реіграбельність і жанрову відповідність ігрових рівнів. *Мета дослідження* полягає у формуванні системної класифікації алгоритмів процедурної генерації лабіринтів за їхніми структурними властивостями, порівнянні їхньої продуктивності, складності проходження та навігаційної ефективності, а також у розробці практичних рекомендацій щодо їх застосування в різних жанрах відеоігор. *Методика* дослідження базується на використанні загальнонаукових методів: системного аналізу, порівняння, синтезу, абдукції, ідеалізації, аналогії, а також логічного та історичного підходів. *Результати* дослідження пропонують системний аналіз та класифікацію алгоритмів генерації лабіринтів на чотири групи: ідеальні, неідеальні, печероподібні та гібридні. Для кожної групи визначено характерні топологічні ознаки, морфологічні особливості та експлуатаційні переваги. *Наукова новизна* полягає у вперше запропонованій системній класифікації алгоритмів генерації лабіринтів за топологічними критеріями, що враховує не лише структурні властивості, а й навігаційні та когнітивні аспекти. Автори також окреслили зони застосування алгоритмів залежно від жанрових вимог, технічних обмежень та очікувань користувача, що дозволяє інтегрувати ці алгоритми в геймдизайн на основі обґрунтованих критеріїв. *Практична значимість* дослідження полягає в можливості використання результатів для розробки адаптивних ігрових рівнів, освітніх платформ, симуляційних систем та інструментів навчального програмування. Запропоновані рекомендації сприяють оптимізації генерації лабіринтів, підвищенню реіграбельності, покращенню користувацького досвіду та стандартизації підходів до процедурної генерації в індустрії відеоігор.

Ключові слова: алгоритми; оптимальність алгоритмів; графи; генерація лабіринтів; розробка ігор; лабіринт.

ПОСТАНОВКА ЗАДАЧІ

Лабіринти у відеоіграх слугують не лише інструментом навігації, а й комплексним елементом дизайну, який поєднує технічні, естетичні та ігрові функції. Використання процедурної генерації, інтерактивних елементів та адаптивних систем дозволяє реалізувати інноваційні підходи до побудови віртуального простору. Це, у свою чергу, підвищує глибину занурення, підтримує дослідницький інтерес та відкриває нові можливості для розвитку інтерактивного ігрового контенту.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Приклади використання тих чи інших алгоритмів для процедурної генерації алгоритмів висвітлені в численних наукових публікаціях. Так, у статті [1] досліджено алгоритм полювання та знищення (Hunt and Kill), де автори дійшли висновку про його потенційно довгий час виконання при великих розмірах лабіринту. У роботі [2] розглянуто три алгоритми генерації лабіринтів (рекурсивного зворотного шляху (Recursive Backtracker), Прима, Крускала) і три алгоритми пошуку шляху (DFS, BFS, A*) в них з позиції пояснення студентам і порівняння показників продуктивності. Автори [3] розглянули ще інші три алгоритми генерації 2D-лабіринтів (DFS, рекурсивний опис (Recursive description), Прима) з позиції можливого усунення їхніх недоліків. У дослідженні [4] на прикладі ігрового мобільного Android-застосунок для дітей розглянуто відмінності генерації лабіринтів різними алгоритмами (двійкового дерева (Binary Tree Algorithm), повторюваного шаблону, випадкового вибору, Вілсона, Олдоса-Бродера, полювання та знищення, рекурсивного зворотного шляху) з метою

вивчення дітьми (вік 5+) базових понять програмування та заохочення їх до обчислювального мислення. У роботі [5] порівняли складність та середній час проходження невеликих лабіринтів розміром 5×5 , 10×10 та 15×15 , згенерованих алгоритмами Крускала, Прима, Еллера та зворотного відстеження (Backtracking). Її автори дійшли висновку про те, що лабіринти, створені алгоритмом зворотного відстеження, є складними і потребують найбільший з-посеред інших час проходження, а лабіринти, створені алгоритмом Прима, є менш складними з найменшим часом проходження. Деякі інші результати продемонстровано у дослідженні [6], в якому порівняно п'ять алгоритмів (Крускала, Олдоса-Бродера, рекурсивного зворотного шляху, Hunt and Kill, двійкового дерева) для створення лабіринтів розмірами 10×10 , 50×50 , 100×100 та 250×250 . Стаття [7] порівнює продуктивність семи алгоритмів (Прима, Крускала, DFS, Hunt and Kill, Еллера, Вілсона, Олдоса-Бродера) за двома параметрами – часом генерації лабіринту й просторовою складністю лабіринту. Найшвидшим для створення лабіринтів виявився DFS. Загалом аналіз наявних публікацій свідчить про те, що порівняння алгоритмів генерації лабіринтів має високу практичну та наукову актуальність, особливо в контексті сучасної розробки ігор, інтелектуальних систем і навчального програмування.

ВІДОКРЕМЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Попри наявність численних досліджень, тема процедурної генерації лабіринтів у відеоіграх залишається недостатньо систематизованою. Є потреба в узагальненій класифікації алгоритмів за топологічними

та навігаційними властивостями, а також у чіткому зіставленні їх з жанровими особливостями ігрових продуктів. Більшість робіт не враховують вплив алгоритмів на реіграбельність, адаптивність середовища та інтерактивність, а порівняння продуктивності часто обмежується окремими метриками без комплексного аналізу. Крім того, бракує практичних рекомендацій для розробників щодо вибору алгоритмів залежно від технічних умов і цілей геймплею. Тому актуальним є дослідження, спрямоване на класифікацію алгоритмів, порівняння їхніх морфологічних і експлуатаційних характеристик, а також формулювання прикладних рекомендацій для інтеграції в ігрові системи.

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТОДИ ДОСЛІДЖЕННЯ

Метою статті є формування системної класифікації алгоритмів процедурної генерації лабіринтів для відеоігор за топологічними та навігаційними властивостями, порівняльний аналіз їхніх морфологічних характеристик, продуктивності та прохідності й надання практичних рекомендацій щодо вибору алгоритмів під жанрові та технічні вимоги.

Методи. Дослідження проводиться з використанням загальнонаукових методів дослідження, серед яких: пошук, системний аналіз, спостереження, синтез, абстрагування та порівняння, абдукція, ідеалізація, пояснення та аналогія, історичний і логічний методи.

Об'єктом дослідження є алгоритми процедурної генерації лабіринтів та згенеровані ними просторові структури в контексті відеоігор.

Предметом дослідження є топологічні властивості згенерованих лабіринтів (наявність/відсутність циклів, зв'язність, прохідність між вузлами, ізотропність).

ОСНОВНИЙ МАТЕРІАЛ

1. Аспекти використання лабіринтів у структурі відеоігор

Лабіринти відіграють важливу роль у формуванні ігрового процесу, оскільки створюють атмосферу загадковості, напруги та виклику для користувача. Аналіз структурних і функціональних характеристик лабіринтів у відеоіграх дозволяє розробникам

розширювати можливості дизайну віртуальних світів і водночас забезпечує гравцям взаємодію з комплексними просторовими середовищами, які потребують стратегічного мислення й творчого підходу.

Одним з ключових аспектів лабіринтів є їх здатність формувати ілюзію нескінченності та непізнаності. Завдяки процедурній генерації лабіринт може набувати унікальних форм під час кожного проходження, що важливо для ігор із високим рівнем реіграбельності. Якщо раніше в класичних аркадних іграх використовувалися фіксовані лабіринтні структури, то сучасні технології дозволяють інтегрувати алгоритми генерації, які забезпечують варіативність, непередбачуваність та адаптивність ігрового середовища відповідно до поточних стандартів геймдизайну.

Другий важливий аспект пов'язаний із просторовою навігацією. Для оптимізації орієнтації гравця в складних лабіринтоподібних структурах використовуються різноманітні інструменти – від візуальних індикаторів (кольорова палітра, текстури, освітлення) до спеціалізованих інтерактивних карт або навігаційних знаків [8]. У табл. 1 подано порівняльну характеристику методів орієнтації, яка демонструє їхні переваги та обмеження. Ефективне поєднання таких засобів сприяє покращенню користувацького досвіду, знижуючи ймовірність фрустрації.

Третій аспект полягає в інтерактивності лабіринтів. Сучасні ігри дедалі частіше відходять від концепції статичного середовища, натомість надають лабіринтам здатність реагувати на дії гравця. Прикладами є пастки або мобільні об'єкти, поведінка яких змінюється залежно від розташування або дій користувача. Така динамічна взаємодія створює додатковий рівень виклику та занурення, а також забезпечує реалізацію нових ігрових механік, зокрема зміну конфігурації лабіринту в режимі реального часу.

Четвертий аспект стосується естетичного оформлення лабіринтів. Візуально-звукове оформлення є визначальним чинником у формуванні емоційного тону ігрового середовища. Колірна гама, освітлення, текстури та інші візуальні елементи здатні створювати атмосферу страху, тривоги або навпаки – чарівності та фантастичності. Звуковий супровід, у тому числі фонові мелодії та звукові ефекти, посилює ці

Таблиця 1. Основні методи орієнтації в лабіринтах у відеоіграх

Метод орієнтації	Опис	Переваги	Недоліки
Візуальні підказки	Використання кольорів, текстур, освітлення для акцентування локацій	Простота реалізації, швидке сприйняття, інтуїтивна навігація	Залежність від графічного стилю гри, можливе перевантаження візуальними елементами
Аудіо-індикатори	Звукові сигнали, які вказують напрямок руху або наближення до цілі	Підсилення ефекту занурення, ефективність у темряві або при обмеженій видимості	Вразливість до налаштувань звуку, складність сприйняття для гравців з порушеннями слуху
Інтерактивні об'єкти	Кarti, знаки, маркери, які надають інформацію про навігацію	Покращення орієнтації, стимулювання дослідження, зменшення фрустрації	Може знижувати рівень виклику, потребує додаткової інтеграції в геймплей

Таблиця 2. Взаємодія лабіринту з елементами гри

Тип взаємодії	Опис	Вплив на гру
Головоломки	Завдання, які потребують логічного аналізу, аналізу або маніпуляцій з об'єктами	Підвищення когнітивного навантаження, стимулювання аналітичного мислення, збагачення геймплею
Секретні проходи	Приховані або альтернативні маршрути, які ведуть до бонусних зон чи скорочень	Заохочення дослідження, підвищення реіграбельності, створення елемента несподіванки
Динамічні пастки	Елементи, які активуються залежно від дій гравця або його позиції	Підвищення напруженості, формування викликів, стимулювання реактивного мислення

враження. Комплексне поєднання візуальних та аудіоелементів забезпечує глибше сприйняття простору та емоційну залученість гравця.

П'ятий аспект – це взаємодія лабіринтів з іншими складовими ігрового процесу. Лабіринти не обмежуються функцією проходження рівня, а виконують роль активного елемента геймплею. Вони можуть містити головоломки, секретні проходи, бонусні зони та альтернативні маршрути, що стимулює дослідницьку активність користувача та сприяє відкриттю додаткового контенту. Табл. 2 систематизує форми інтеграції лабіринтів з іншими ігровими механіками.

Шостий аспект полягає у впливі лабіринтів на загальну динаміку геймплею. Лабіринти часто стають базою для реалізації різних ігрових механік: від бойових систем до стелс-проходжень або розв'язання головоломок. Здатність лабіринтів змінюватися під час проходження дозволяє створити адаптивне середовище, яке підтримує інтерес гравця та сприяє експериментуванню з новими концепціями.

2. Класифікація алгоритмів процедурної генерації лабіринтів

Системи процедурної генерації лабіринтів відіграють ключову роль у розробці динамічних та варіативних ігрових середовищ. Вони забезпечують створення унікальних рівнів, адаптивних до змінних параметрів геймплею, та сприяють підвищенню реіграбельності. Основною метою таких систем є автоматичне формування просторових структур, які одночасно відповідають вимогам дизайну гри, забезпечують ігровий баланс і задовольняють очікування користувача щодо складності й новизни.

Основні функціональні вимоги до систем генерації лабіринтів:

- *інтеграція генеративних алгоритмів*: система має реалізовувати один або декілька алгоритмів генерації лабіринтів, які створюватимуть випадкові, структуровано задані лабіринтоподібні рівні. Це забезпечуватиме розмаїття просторових конфігурацій та варіативність проходження рівнів;

- *можливість параметризації генерації*: інтерфейс системи має передбачати зміну ключових параметрів: розмірів лабіринту, щільності коридорів, рівня складності, глибини розгалужень, частоти появи пасток, головоломок або інших геймплейних об'єктів [9]. Такий підхід дозволить розробникам та користувачам тонко налаштовувати структуру рівня

відповідно до ігрових цілей;

- *унікальність результатів генерації*: кожен запуск алгоритму створення лабіринту має генерувати новий, неповторний лабіринт, що важливо для ігор з високим ступенем реіграбельності. При цьому система має дозволяти зберігати згенеровані шаблони для подальшого використання (наприклад, у PvP-режимах, speedrun-змаганнях тощо);

- *оптимізація продуктивності*: генерація повинна відбуватися з мінімальними витратами ресурсів, не спричиняючи перешкод у загальному потоці гри [10]. Для цього можуть застосовуватись стрімінгові системи рівнів, попереднє кешування структур або розділення процесу генерації на окремі етапи, наприклад, генерація на етапі завантаження ігрової сцени.

Оскільки властивості лабіринтів (ідеальність, циклічність, прохідність) визначають топологію результату, важливою є класифікація алгоритмів генерації лабіринтів за властивостями згенерованого лабіринту (рис. 1).

1. *Ідеальні лабіринти* (*Perfect Mazes*) характеризуються відсутністю циклів та наявністю одного шляху між будь-якими двома точками. Ця топологічна властивість прямо відповідає побудові випадкового остовного дерева над графом клітин, тому більшість класичних алгоритмів у цій категорії фактично конструюють різновиди випадкових остовних дерев з різними статистичними упередженнями. До цієї групи належать алгоритми DFS, Прима, Крускала, Еллера, Вілсона, адже вони забезпечують передбачувану структуру та високу зв'язність.

Алгоритм обходу вглуб (DFS, recursive backtracking) генерує лабіринти з довгими коридорами та великою кількістю тупиків, що робить його придатним для головоломок і навчальних задач з контрольованим когнітивним навантаженням. Водночас він демонструє слабкий контроль над глобальною щільністю та геометрією, а також чутливість до порядку обходу сусідів, що призводить до характерних артефактів структури. *Алгоритм Прима* має кращий контроль локальної щільності та збалансовані гілки, що корисно для рівнів з помірною складністю, проте складність реалізації та залежність від структури даних можуть ускладнювати використання в реальному часі. *Алгоритм Крускала* працює на основі випадкового вибору ребер із перевіркою на відсутність циклів, що забезпечує генерацію остовного

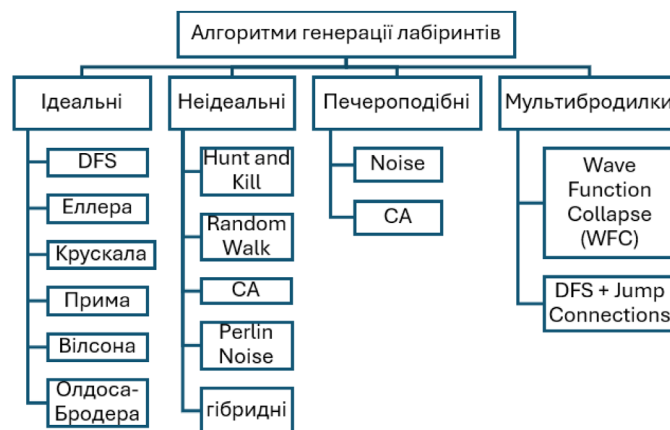


Рис. 1. Класифікація алгоритмів процедурної генерації лабіринтів за властивостями лабіринту

дерева з рівномірною глобальною щільністю та мінімальними напрямними артефактами. Це створює лабіринти зі збалансованою навігаційною складністю без домінування тупикових гілок. Основними обмеженнями є додаткові обчислювальні витрати на підтримку структур неперетинних множин та менший локальний контроль, порівняно з фронтowymi методами. *Алгоритм Вілсона* ґрунтується на багаторазових «loop-erased random walks»: починає з однієї клітини і додає нові шляхи без циклів, що значно пришвидшує генерацію при збереженні рівномірності. Він забезпечує ізотропну структуру, яка є цінною для симуляцій, досліджень та завдань зі строго випадковою топологією, проте алгоритм має вищу обчислювальну складність та вимагає ретельної реалізації з контролем видалення петель і обробки меж. *Алгоритм Олдоса-Бродера* також гарантує рівномірний розподіл, але базується на випадковому блуканні до повного покриття графа, додаючи ребра лише при першому відвідуванні. Через високу ймовірність повторних відвідин очікуваний час генерації значно зростає, що обмежує практичне використання у великих або динамічних середовищах. *Алгоритм Еллера* є ефективним у потоковій генерації з мінімальними вимогами до пам'яті. Це робить його придатним для нескінченних або довгих рівнів у side-scroller та endless runner жанрах, водночас обмежена глобальна топологічна гнучкість і характерна стрічкова морфологія звужують сферу його застосування.

Порівнюючи ці алгоритми за експлуатаційними критеріями, можна окреслити чіткі зони застосування. Алгоритм DFS вирізняється простотою й високою продуктивністю, що важливо для мобільних платформ, навчальних інструментів і головоломок із низьким навантаженням на систему. Алгоритми Прима і Крускала забезпечують більш збалансовану морфологію з кращим контролем над структурою проходів, що корисно для рівнів середньої складності, зокрема в adventure- та dungeon-жанрах. Алгоритм Вілсона генерує рівномірно випадкові остовні дерева

з ізотропною топологією, що робить його ефективним у симуляціях і дослідницьких системах. Порівняно з Олдосом-Бродером, який має аналогічні статистичні властивості, Вілсон є практичнішим завдяки значно кращій продуктивності. Алгоритм Еллера орієнтований на стрімінгову генерацію, потребує мінімальних ресурсів і оптимально підходить для нескінченних або лінійних рівнів у жанрах side-scroller та endless runner.

Морфологічно ідеальні лабіринти дають прозору метрику складності (довжина шляху, щільність тупиків, кількість розгалужень), що спрощує автоматизоване тестування [11]. Проте відсутність циклів обмежує варіативність маршруту й тактичні можливості. Це компенсується післяобробкою (додаванням циклів або кімнат) або гібридизацією з семантичними правилами, якщо потрібна більша нелінійність. Тим самим вибір алгоритму в межах ідеальної категорії визначається пріоритетами: простота й швидкодія (DFS), контроль над структурою (Прим, Крускал), статистична рівномірність (Вілсон, Олдос-Бродер) чи потокова генерація з обмеженою пам'яттю (Еллер). Така класифікація дозволяє адаптивно інтегрувати алгоритми у різні ігрові жанри та рушії, балансуючи між технічними вимогами та дизайном рівнів.

2. Неідеальним лабіринтам (Imperfect Mazes) притаманні наявність циклів, ізольованих ділянок або кількох альтернативних маршрутів між точками. Це робить їх менш передбачуваними з точки зору топології, але водночас більш наближеними до природних або архітектурно складних структур. До таких алгоритмів належать підходи на основі випадкових блукань (Hunt and Kill, Random Walk), клітинних автоматів (Conway's Game of Life, Maze CA, Rule 4/5), шумових функцій (Perlin Noise, Simplex Noise), а також гібридні методи, які поєднують випадковість із семантичними або еволюційними правилами.

Алгоритм Hunt and Kill створює лабіринти з великою кількістю циклів і нерівномірною щільністю проходів [1]. Його перевагою є простота реалізації

та можливість отримати лабіринти з високою варіативністю, що підходить для ігор жанру roguelike або dungeon crawler, де важлива непередбачуваність. Недоліком є відносно довгий час генерації на великих сітках та відсутність контролю над глобальною топологією. *Алгоритми на основі клітинних автоматів* (Cellular Automata, CA), зокрема Conway's Game of Life або спеціалізовані варіанти Maze CA, формують лабіринти шляхом ітеративного оновлення станів клітин за локальними правилами. Такі методи дозволяють створювати печероподібні структури з органічними переходами, які нагадують природні підземелля. Їхня специфіка полягає у високій залежності від початкових умов і правил виживання, що може призводити як до надмірно хаотичних, так і до надто регулярних структур [12]. Перевагою є здатність генерувати складні морфології, які важко відтворити класичними графовими алгоритмами, однак недоліком є ризик утворення непрохідних або ізольованих ділянок, що потребує постпроцесингу. *Шумові алгоритми*, зокрема Perlin Noise, застосовуються для створення лабіринтів із плавними переходами та нерегулярними формами. Їх використання дозволяє отримати лабіринти з високим рівнем візуальної складності, які добре підходять для декоративних або естетично орієнтованих рівнів [4]. Водночас такі алгоритми не гарантують прохідності, тому їх застосування обмежене сценаріями, де лабіринт виконує радше роль художнього середовища, ніж ігрової механіки.

Порівняльний аналіз показує, що неідеальні алгоритми забезпечують більшу варіативність і природність форм, ніж ідеальні, але втрачають у передбачуваності та контрольованості. Вони особливо ефективні у жанрах, де важливе відчуття хаосу, дослідження або естетична складність, наприклад у roguelike, survival horror чи освітніх іграх для демонстрації принципів складних систем. Їхнім ключовим недоліком є відсутність гарантії єдиного шляху та ризик утворення непрохідних ділянок, що вимагає додаткових механізмів перевірки та корекції. Тому вибір алгоритму з категорії «неідеальних» має ґрунтуватися на балансі між потребою у варіативності та вимогами до ігрової навігації, а також на технічних обмеженнях рушія чи платформи.

3. Печероподібні (cave-like) алгоритми процедурної генерації лабіринтів відрізняються від класичних графових методів тим, що їхня мета полягає не у створенні строгої основної структури з єдиним маршрутом, а у формуванні органічних, нерегулярних просторів, які нагадують природні печери, підземелля чи хаотичні тунельні системи. Переважно такі алгоритми базуються на клітинних автоматах або шумових функціях, які дозволяють отримати морфологію з плавними переходами та нерівномірною щільністю стін. Одним із поширених підходів є використання CA-алгоритму *Правило 4/5 (Rule 4/5* або

«згладжування печери»), коли клітина стає стіною, якщо має чотири або більше сусідів-стін, інакше вона перетворюється на прохід. Як показано у роботі [12], кілька ітерацій такого згладжування формують структури, які нагадують печери, з характерними вузькими проходами та великими залами. Цей алгоритм активно застосовується у roguelike-іграх, де важлива природність простору та відчуття дослідження [13]. Його перевагою є простота реалізації та швидкість, проте недоліком є відсутність гарантії прохідності, що часто потребує додаткового постпроцесингу для усунення ізольованих ділянок. Іншим підходом є використання модифікованих варіантів CA-алгоритму *«Життя Конвея» (Conway's Game of Life-based CA)* [4], які дозволяють створювати складні, саморегульовані простори, що добре підходять для освітніх ігор, орієнтованих на демонстрацію принципів клітинних автоматів і складних систем. Їхня специфіка полягає у високій чутливості до початкових умов: незначна зміна стартового патерна може призвести до радикально іншої морфології. Це робить їх цікавими для експериментів, але менш передбачуваними у практичному геймдизайні. *Шумові алгоритми*, зокрема Perlin Noise, застосовуються для генерації печероподібних лабіринтів із плавними, «природними» контурами [14]. Вони широко використовуються у створенні карт для survival- та sandbox-ігор, де важлива естетика та відчуття реалістичності середовища. Шумові функції дозволяють отримати складні морфології з високим рівнем візуальної варіативності, проте вони не гарантують зв'язності простору, що обмежує їх застосування у випадках, де потрібна чітка навігаційна структура [15]. Перевагою є можливість масштабування та контроль параметрів «гладкості» чи «шорсткості» лабіринту, недоліком – потреба у додаткових алгоритмах для забезпечення прохідності.

Порівняно з класичними «ідеальними» алгоритмами печероподібні методи забезпечують значно більшу варіативність і природність форм, що робить їх особливо цінними для жанрів roguelike, survival чи dungeon crawler, де важливе відчуття хаосу та дослідження. Водночас їхнім ключовим недоліком є відсутність строгого контролю над топологією та необхідність додаткових механізмів перевірки прохідності, що обмежує використання у випадках, де потрібна гарантована навігаційна структура.

4. Multi-walkers – це алгоритми, в яких лабіринт формується не одним «бродячим агентом» (walker), а кількома, або комбінацією підходів з constraints або шаблонами, що дозволяють одночасно або по черзі розбивати простір на шляхи / коридори / кімнати, з урахуванням локальних та глобальних правил. До цієї категорії часто відносять Wave Function Collapse (WFC), який дозволяє створювати складні багаторівневі карти з контрольованими патернами [4], та гібридні модифікації DFS із додатковими

«стрибковими» з'єднаннями між віддаленими ділянками, які дозволяють уникати надмірної лінійності та створювати багаторівневі маршрути [15].

Порівняння мультибродилок із класичними алгоритмами показує, що вони забезпечують значно більшу варіативність і природність лабіринтів, але втрачають у передбачуваності та контролюваності. Їхніми перевагами є здатність створювати багатопланові, складні та непередбачувані структури, що підвищує реіграбельність ігрових рівнів. Недоліками є ризик утворення надмірно хаотичних або непрохідних ділянок, потреба у додаткових механізмах перевірки прохідності та вищі обчислювальні витрати. Для великих або нескінченних карт WFC без модифікацій не завжди працює добре і застосування Nested WFC або гібридних правил (де частини карти згенеровані як WFC [16], інші – традиційно або з агентамибродячими) дозволяє знизити ці проблеми [17]. Дизайнери часто інтегрують «шаблони модульних кімнат» перед WFC, або задають «core rooms» (центральні кімнати), а потім WFC заповнює пробіли – це забезпечує структурну опору та зменшує випадковість, яка може бути занадто хаотичною [18]. Тим самим мультибродилки доцільно застосовувати, коли пріоритетом є варіативність і динамічність ігрового процесу, тоді як для освітніх або мобільних застосунків із жорсткими обмеженнями ресурсів більш придатними залишаються класичні алгоритми.

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

У ході дослідження проаналізовано та систематизовано різновиди алгоритмів процедурної генерації лабіринтів, які застосовуються у відеоіграх. Класифікація за топологічними властивостями (ідеальні, неідеальні, печероподібні, мультибродилки) дозволила виявити ключові відмінності між ними. Ідеальні алгоритми (DFS, Прима, Крускала, Вілсона, Олдоса-Бродера, Еллера) забезпечують строгий контроль над зв'язністю та однозначність маршруту, проте обмежують варіативність ігрового досвіду. Неідеальні та печероподібні методи (Cellular Automata, Perlin Noise) створюють органічні та хаотичні структури, що підвищує реіграбельність, але водночас породжує ризик непрохідності. Алгоритми мультибродилок (multi-walker, WFC, DFS із додатковими з'єднаннями) демонструють високу варіативність

і складність, проте вимагають додаткових ресурсів і перевірок.

Порівняння показало, що вибір алгоритму має визначатися не лише технічними характеристиками (час генерації, складність реалізації, споживання пам'яті), а й жанровими вимогами гри. Для головоломок і навчальних застосунків доцільно вибирати ідеальні алгоритми, для roguelike та survival – печероподібні й мультибродилки, тоді як для мобільних ігор із жорсткими обмеженнями ресурсів доречними є прості алгоритми на кшталт DFS. Важливим результатом є також виявлення ролі постпроцесингу: саме він дозволяє перетворювати ймовірно прохідні лабіринти на гарантовано прохідні, зберігаючи при цьому природність морфології.

Результати дослідження підтверджують, що універсального алгоритму не існує: кожен метод має власну нішу застосування, а оптимальність визначається балансом між продуктивністю, морфологічними властивостями та ігровим контекстом.

ВИСНОВКИ

Запропонована класифікація алгоритмів процедурної генерації лабіринтів за топологічними властивостями дозволяє системно оцінювати їхні можливості та обмеження. Ідеальні алгоритми забезпечують строгий контроль над структурою та гарантовану прохідність лабіринтів, але обмежують варіативність ігрових сценаріїв. Неідеальні та печероподібні методи створюють органічні, хаотичні структури, що підвищує реіграбельність, проте потребує додаткових перевірок і корекцій для забезпечення прохідності. Алгоритми мультибродилок забезпечують високу складність і варіативність лабіринтів, але є ресурсомісткими та менш передбачуваними. Постпроцесинг (перевірка зв'язності, додавання мостів, з'єднання кімнат) є ключовим інструментом для перетворення ймовірно прохідних лабіринтів на гарантовано прохідні. Практичне застосування алгоритмів має визначатися жанровими особливостями гри: від навчальних ігор і головоломок до roguelike, survival та sandbox-проектів.

Отримані результати можуть бути використані як методологічна основа для вибору алгоритмів у розробці ігор, а також для створення адаптивних систем генерації рівнів із балансом між продуктивністю, складністю та ігровим досвідом.

REFERENCES

- [1] Marchuk, G. V. & Liubchenko, D. V. (2024). Heneratsiia labiryntiv za dopomohoiu alhorytmu Hunt and Kill [Generation of mazes using the hunt-and-kill algorithm]. *Scientific notes of Taurida National V. I. Vernadsky University. Series: Technical Sciences*, 35 (74), 3, 130–135. <https://doi.org/10.32782/2663-5941/2024.3.1/20> [in Ukrainian].
- [2] Binayak, B. (2025). Interactive Maze Generation and Pathfinding Algorithms: An Educational Visualization Platform for Algorithmic Analysis. *Zenodo*. <https://doi.org/10.5281/zenodo.15824242>
- [3] Vijaya, J., Gopu, A., Vinayak, K., Singh, J., Malhotra, K. (2024). Analysis of Maze Generation Algorithms. *5th IEEE International Conference on Innovative Trends in Information Technology (ICITIT)*, Kottayam, India, 15–16 March 2024, 1–6. <https://doi.org/10.1109/ICITIT61487.2024.10580178>

- [4] Čarapina, M., Staničić, O., Dodig, I., & Cafuta, D. (2024). A Comparative Study of Maze Generation Algorithms in a Game-Based Mobile Learning Application for Learning Basic Programming Concepts. *Algorithms*, 17(9), 404. <https://doi.org/10.3390/a17090404>
- [5] Cahyakusuma I., Istiono, W. Algorithmic diversity in maze generation: Comparative study of Backtracking, Kruskal's, Prim's, and Eller's algorithms. *International Journal of Intelligent Systems and Applications in Engineering*. 2024. Vol. 12(3). P. 1281–1287. <https://ijisae.org/index.php/IJISAE/article/view/5518>
- [6] Marchuk, G. V., Graf, M. S., Levkivskyi, V. L., & Venhlovskaya, Yu. V. (2024). *Analiz ta porivniannia isnuivayukh metodiv heneratsii labiryntiv v kompiuternykh ihrakh* [Analysis and comparison of existing maze generation methods in computer games]. *Visnyk of Kherson National Technical University*, 3(90), 228–237. <https://doi.org/10.35546/kntu2078-4481.2024.3.29> [in Ukrainian].
- [7] Mane, D., Harne, R., Pol, T., Asthagi, R., Shine, S., & Zope, B. (2023). An Extensive Comparative Analysis on Different Maze Generation Algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, 12(2), 37–47. <https://ijisae.org/index.php/IJISAE/article/view/3557>
- [8] Trofymenko, O. G., Zadereyko, O. V., Loginova, N. I., & Shevchenko, O. V. (2025). Instrumenty shuchnoho intelektu v rozrobtsi hrafichnykh elementiv, 3D-modelei ta inshykh komponentiv videoihr [Artificial intelligence tools in the development of graphic elements, 3D models and other video game components]. *Informatics and mathematical methods in simulation*, 15(2), 276–287. <https://doi.org/10.15276/imms.v15.no2.276> [in Ukrainian].
- [9] Trofymenko, O., Zadereyko O., Balandina, N., Tolocknov, A., & Huselnikov, I. (2024). Vektorna alhebra dlia rozrobky ihor [Vector algebra for gamedev]. *Electronic Professional Scientific Journal "Cybersecurity: Education, Science, Technique"*, 2(26), 71–80. <https://doi.org/10.28925/2663-4023.2024.26.626> [in Ukrainian].
- [10] Trofymenko, O., Dyka, A., Loboda, Y., Tolocknov, A., & Bondarenko, M. (2025). Shuchnyi intelekt u rozrobtsi ihor [Artificial intelligence in the gamedev]. *Electronic Professional Scientific Journal "Cybersecurity: Education, Science, Technique"*, 3(27), 109–119. <https://doi.org/10.28925/2663-4023.2025.27.701> [in Ukrainian].
- [11] Trofymenko, O., Dyka, A., Zadereyko O., & Shevchenko O. (2025). Sfery zastosuvannia shuchnoho intelektu v rozrobtsi ta testuvanni ihor [Areas of application of artificial intelligence in game development and testing]. *Electronic Professional Scientific Journal "Cybersecurity: Education, Science, Technique"*, 1(29), 41–58. <https://doi.org/10.28925/2663-4023.2025.29.832> [in Ukrainian].
- [12] Malem, F. (2023). Wall Object Design in Maze Game Using Cellular Automata Algorithm. *Journal of Computer Engineering: Progress, Application and Technology*. 2(1), 27–35. <https://doi.org/10.25124/cepat.v2i01.5781>
- [13] Wu, Z. & Liu, J. (2024). Perlin noise and its improvements: A literature review. *Applied and Computational Engineering*, 77, 278–287. <https://doi.org/10.54254/2755-2721/77/20240437>
- [14] Mountains, Cliffs, and Caves: A Comprehensive Guide to Using Perlin Noise for Procedural Generation. (2025). Retrieved from: <https://vuink.com/post/wqujvyxvaf-d-dpbz/mountains-cliffs-and-caves-a-comprehensive-guide-to-using-perlin-noise-for-procedural-generation>
- [15] Piao, J., Hu, X., & Zhou, Q. (2024). Maze and navigation algorithms in game development. *Applied and Computational Engineering*, 79, 11–19. <https://doi.org/10.54254/2755-2721/79/20241081>
- [16] Langendam, Th., & Bidarra, R. (2022). MiWFC – Designer Empowerment through mixed-initiative Wave Function Collapse. *Proceedings of the 17th International Conference on the Foundations of Digital Games (FDG '22)*. ACM, New York, NY, USA, 66, 1–8. <https://doi.org/10.1145/3555858.3563266>
- [17] Christensen R. Wave Function Collapse. (2022). <https://rmmiusr.com/wave-function-collapse/>
- [18] Yuhe, N., Shaoming, Z., Zhan, Z., & Xuan, S. (2023). Extend Wave Function Collapse to Large-Scale Content Generation. *arXiv*. <https://doi.org/10.48550/arXiv.2308.07307>

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Марчук, Г. В., Любченко, Д. В. (2024). Генерація лабіринтів за допомогою алгоритму Hunt and Kill. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки*. Т. 35 (74), № 3. С. 130–135. DOI: <https://doi.org/10.32782/2663-5941/2024.3.1/20>
- [2] Binayak, B. (2025). Interactive Maze Generation and Pathfinding Algorithms: An Educational Visualization Platform for Algorithmic Analysis. *Zenodo*. DOI: <https://doi.org/10.5281/zenodo.1582422>
- [3] Vijaya, J., Gopu, A., Vinayak, K., Singh, J., Malhotra, K. (2024). Analysis of Maze Generation Algorithms. *5th IEEE International Conference on Innovative Trends in Information Technology (ICITIT'24)*. Kottayam, India, 15–16 March. P. 1–6. DOI: <https://doi.org/10.1109/ICITIT61487.2024.10580178>
- [4] Čarapina, M., Staničić, O., Dodig, I., Cafuta, D. A (2024). Comparative Study of Maze Generation Algorithms in a Game-Based Mobile Learning Application for Learning Basic Programming Concepts. *Algorithms*. Vol. 17(9). P. 404. DOI: <https://doi.org/10.3390/a17090404>
- [5] Cahyakusuma, I., Istiono, W. (2024). Algorithmic diversity in maze generation: Comparative study of Backtracking, Kruskal's, Prim's, and Eller's algorithms. *International Journal of Intelligent Systems and Applications in Engineering*. Vol. 12(3). P. 1281–1287. URL: <https://ijisae.org/index.php/IJISAE/article/view/5518>
- [6] Марчук, Г. В., Граф, М. С., Левківський, В. Л., Венгловська, Ю. В. (2024). Аналіз та порівняння існуючих методів генерації лабіринтів в комп'ютерних іграх. *Вісник Херсонського національного технічного університету*. № 3(90). С. 228–237. <https://doi.org/10.35546/kntu2078-4481.2024.3.29>

- [7] Mane, D., Harne, R., Pol, T., Asthagi, R., Shine, S., Zope, B. (2023). An Extensive Comparative Analysis on Different Maze Generation Algorithms. *International Journal of Intelligent Systems and Applications in Engineering*. Vol. 12(2). P. 37–47. URL: <https://ijisae.org/index.php/IJISAE/article/view/3557>
- [8] Трофименко, О. Г., Задерейко, О. В., Логінова, Н. І., Шевченко, О. В. (2025). Інструменти штучного інтелекту в розробці графічних елементів, 3D-моделей та інших компонентів відеоігор. *Інформатика та математичні методи в моделюванні*. Т.15, № 2. С. 276–287. <https://doi.org/10.15276/imms.v15.no2.276>
- [9] Трофименко, О. Г., Задерейко, О. В., Баландіна, Н. М., Толокнов, А. А., Гусельников, І. М. (2024). Векторна алгебра для розробки ігор. *Кібербезпека: освіта, наука, техніка*. № 2(26). С. 71–80. <https://doi.org/10.28925/2663-4023.2024.26.626>
- [10] Трофименко, О. Г., Дика, А. І., Лобода, Ю. В., Толокнов, А. А., Бондаренко, М. Є. (2025). Штучний інтелект у розробці ігор. *Кібербезпека: освіта, наука, техніка*. № 3(27). С. 109–119. <https://doi.org/10.28925/2663-4023.2025.27.701>
- [11] Трофименко, О. Г., Дика, А. І., Задерейко, О. В., Шевченко, О. В. (2025). Сфери застосування штучного інтелекту в розробці та тестуванні ігор. *Кібербезпека: освіта, наука, техніка*. № 1(29). С. 41–59. <https://doi.org/10.28925/2663-4023.2025.27.701>
- [12] Malem, F. (2023). Wall Object Design in Maze Game Using Cellular Automata Algorithm. *Journal of Computer Engineering: Progress, Application and Technology*. Vol. 2(1). P. 27–35. <https://doi.org/10.25124/cepat.v2i01.5781>
- [13] Wu, Z., Liu, J. (2024). Perlin noise and its improvements: A literature review. *Applied and Computational Engineering*. Vol. 77. P. 278–287. <https://doi.org/10.54254/2755-2721/77/20240437>
- [14] Mountains, Cliffs, and Caves: A Comprehensive Guide to Using Perlin Noise for Procedural Generation. 2025. URL: <https://vuink.com/post/wqujvyxvaf-d-dpbz/mountains-cliffs-and-caves-a-comprehensive-guide-to-using-perlin-noise-for-procedural-generation>
- [15] Piao, J., Hu, X., Zhou, Q. (2024). Maze and navigation algorithms in game development. *Applied and Computational Engineering*. Vol. 79. P. 11–19. <https://doi.org/10.54254/2755-2721/79/20241081>
- [16] Langendam, Th., Bidarra, R. (2022). MiWFC – Designer Empowerment through mixed-initiative Wave Function Collapse. *Proceedings of the 17th International Conference on the Foundations of Digital Games (FDG '22)*. ACM, New York, NY, USA. Vol. 66. P. 1–8. DOI: <https://doi.org/10.1145/3555858.3563266>
- [17] Christensen, R. (2022). Wave Function Collapse. URL: <https://rmminusr.com/wave-function-collapse/>
- [18] Yuhe, N., Shaoming, Z., Zhan, Z., Xuan, S. (2023). Extend Wave Function Collapse to Large-Scale Content Generation. *arXiv*. <https://doi.org/10.48550/arXiv.2308.07307>

© Трофименко О. Г., Задерейко О. В., Михелєв І. Л., Ворона М. В., Павленко А. Ю.
 Дата надходження статті до редакції: 28.10.2025
 Дата затвердження статті до друку: 17.11.2025
 Опубліковано: 30.12.2025

Стаття поширюється на умовах ліцензії CC BY 4.0