

**Сергій Анатолійович ГРІНЕНКО<sup>1</sup>**,

кандидат технічних наук, доцент, кафедра інформаційних технологій,

ORCID: <https://orcid.org/0000-0001-5544-2605>

**Максим Андрійович КОНОНЕНКО<sup>1</sup>**,

магістрант кафедри інформаційних технологій,

ORCID: <https://orcid.org/0009-0008-0551-3274>

<sup>1</sup> Національний транспортний університет

Прийняття: 19/06/2026

Рецензія: 27/06/2026

Публікація: 09/07/2026

DOI: <https://doi.org/10.53920/ITS-2026-1-8>

## АНАЛІЗ ПІДХОДІВ ПРОЦЕДУРНО ГЕНЕРОВАНИХ ІГРОВИХ РІВНІВ У ВЕБОРІЄНТОВАНИХ СИСТЕМАХ

удк

004.4; 004.6; 004.9



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Гріненко С.А.,  
Кононенко М.А.,  
2026

*Розглядається актуальна задача аналізу підходів до інтелектуального аналізу процедурно згенерованих ігрових рівнів у вебсистемах. Сучасні методи генерації процедурного контенту дозволяють автоматизовано створювати велику кількість ігрових карт. Однак забезпечення їхньої здатності до проведення гри, балансу та відповідності вимогам ігрового дизайну залишається актуальним завданням. Запропоновано підхід, який поєднує алгоритми генерації просторових структур з інструментами для автоматизованого аналізу маршрутів та оцінки якості рівнів. Особливий акцент робиться на використанні алгоритмів пошуку шляхів, створенні теплових карт руху та моделюванні динамічних змін середовища. Розглянуто можливість використання алгоритмів пошуку в ширину (BFS) для визначення найкоротших маршрутів, а також методів просторової рандомізації для визначення альтернативних шляхів руху. Досліджено, що поєднання механізмів процедурної генерації та аналізу покращує якість згенерованих рівнів та скорочує час, необхідний для ручного тестування. Запропонований підхід може служити інструментом прийняття рішень для розробників комп'ютерних ігор, дослідників штучного інтелекту та розробників інтерактивних систем.*

*Практичне значення цієї роботи полягає у встановленні методологічних основ для автоматичної оцінки структурних особливостей ігрових середовищ. Результати дослідження демонструють потенціал інтеграції алгоритмів*

ISSN 2786-7226 (print)  
ISSN 2786-7234 (online)

**мів аналізу графів, процедурної генерації контенту та вебтехнологій для розробки адаптивних цифрових середовищ наступного покоління. Крім того, були досліджені можливості використання клієнтських вебтехнологій для реалізації інтерактивних систем аналізу без серверної інфраструктури. Це зменшує обчислювальні зусилля та забезпечує високу швидкість взаємодії з користувачем. Представлений підхід сприяє автоматизації процесів проектування цифрового середовища та створює передумови для впровадження інтелектуальних механізмів адаптації контенту до поведінки користувача.**

**Результати застосовні не лише до індустрії відеоігор, але й до таких галузей, як моделювання складних просторових систем, робототехніка, навчальні симулятори та віртуальні середовища. Перспективним підходом для майбутніх досліджень є інтеграція методів машинного навчання для автоматичного налаштування параметрів генерації, прогнозування поведінки користувачів та проектування персоналізованих цифрових просторів з урахуванням індивідуальних сценаріїв взаємодії.**

**Ключові слова:** процедурна генерація контенту, веб технології, ігрові рівні, алгоритм BFS, теплова карта, аналіз маршрутів, алгоритм пошуку шляху.

---

Sergiy GRINENKO, Maksym KONONENKO

## **ANALYSIS OF APPROACHES TO PROCEDURALLY GENERATED GAME LEVELS IN WEB-ORIENTED SYSTEMS**

**The current task of analyzing approaches to the intellectual analysis of procedurally generated game levels in web systems is considered. Modern methods of generating procedural content allow for the automated creation of a large number of game maps. However, ensuring their playability, balance, and compliance with game design requirements remains a pressing task. An approach is proposed that combines algorithms for generating spatial structures with tools for automated route analysis and level quality assessment. Special emphasis is placed on the use of path-finding algorithms, creating heat maps of movement, and modeling dynamic changes in the environment. The possibility of using breadth-first search (BFS) algorithms to determine the shortest routes, as well as spatial randomization methods to determine alternative movement paths, is considered. It is studied that the combination of procedural generation and analysis mechanisms improves the quality of generated levels and reduces the time required for manual testing. The proposed approach can serve as a decision-making tool for computer game developers, artificial intelligence researchers, and developers of interactive systems.**

***The practical significance of this work lies in establishing methodological foundations for automatic assessment of structural features of game environments. The research results demonstrate the potential of integrating graph analysis algorithms, procedural content generation, and web technologies for developing adaptive digital environments of the next generation. In addition, the possibilities of using client-side web technologies for implementing interactive analysis systems without server infrastructure were investigated. This reduces computational effort and ensures high speed of interaction with the user. The presented approach contributes to the automation of digital environment design processes and creates prerequisites for the implementation of intelligent mechanisms for adapting content to user behavior.***

***The results are applicable not only to the video game industry, but also to such industries as modeling of complex spatial systems, robotics, educational simulators, and virtual environments. A promising approach for future research is the integration of machine learning methods for automatic adjustment of generation parameters, prediction of user behavior, and design of personalized digital spaces taking into account individual interaction scenarios.***

**Keywords:** procedural content generation, web technologies, game levels, BFS algorithm, heat map, route analysis, path finding algorithm.

**Постановка проблеми.** Розвиток індустрії комп'ютерних ігор супроводжується зростанням вимог до обсягу та різноманітності цифрового контенту. Створення великих ігрових світів вручну є трудомістким та дорогим. Тому процедурні методи генерації контенту набули широкого поширення. Однак автоматично згенеровані рівні не завжди відповідають вимогам до ігрової зручності, балансу та логіки гри.

Існує потреба в інструментах для автоматичного аналізу якості рівнів та оцінки їхньої зручності використання.

Ця проблема є особливо актуальною з огляду на зростаючу складність сучасних ігрових середовищ, де, окрім технічної точності згенерованих карт, має бути гарантований також комфортний ігровий досвід. Важливим завданням є розробка методів, що дозволяють автоматично визначати ігрову зручність рівнів, оцінювати ефективність навігації, виявляти потенційні проблеми в просторовій структурі та формувати рекомендації щодо їх вирішення ще на етапі проектування.

**Аналіз останніх досліджень і публікацій.** Варто виділити кілька ключових архітектурних методів генерації. Найвідомішим прикладом макро-генерації є використання функцій неперервного шуму (наприклад, Perlin Noise або Value Noise) для створення природних ландшафтів. Безперечним лідером цього підходу є гра Minecraft.

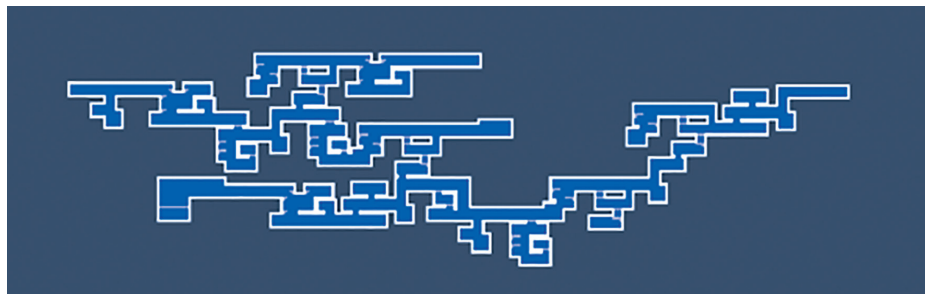
Цей проект генерує практично нескінченний тривимірний піксельний світ, застосовуючи багатозарові математичні шуми для формування височин, гір, океанів, систем печер та логічного розподілу кліматичних біомів. Кожен згенерований світ визначається унікальним початковим числом — «сідом» (seed), що дозволяє гравцям ділитися генераціями та відтворювати абсолютно ідентичні світи на різних пристроях.

Інший, структурований підхід — це гібридна генерація на основі фрагментів або кімнат (Room-based / Chunk-based generation). У цьому випадку алгоритм не створює кожен блок з нуля. Замість цього він працює як конструктор, збираючи рівень із сотень невеликих «кімнат», які були заздалегідь спроектовані та збалансовані левел-дизайнерами вручну.

Відмінними прикладами такої системи є хітові інді-проекти Spelunky та Dead Cells [5]. Програма випадковим чином обирає кімнати з бази даних, перевіряє їх на топологічну сумісність (щоб виходити з однієї кімнати логічно і фізично з'єднувалися з входами іншої) і вибудовує з них складний, багаторівневий лабіринт.

Аналогічний метод просторового розбиття використовується у мобільному хіті Soul Knight, де процедурний алгоритм не лише формує загальну топологію поверху, складаючи його з прямокутних кімнат, але й алгоритмічно визначає їхнє призначення: розраховує дистанцію до кімнати з босом, генерує безпечні зони з торговцями та кімнати зі скарбами.

Саме цей гібридний метод генерації забезпечує ідеальний баланс між непередбачуваністю випадкових алгоритмів та гарантованим контролем якості ручного левел-дизайну. На рисунку нижче наведено типовий приклад такої структурної генерації у сучасному платформері, де чітко простежується розбиття ігрового простору на окремі взаємопов'язані зони з різним рівнем архітектурної складності та вертикальності.



**Рис. 1. Приклад PCG в Dead Cells**

Джерело: [2]

Як видно з наведеної топологічної структури, алгоритм генерації не просто хаотично розставляє платформи та об'єкти, а слідує чіткому графу математичної прохідності (Critical Path). Це гарантує, що гравець завжди матиме гарантовану фізичну можливість дістатися від стартової локації до фінішної, навіть якщо оптимальний або найкоротший шлях буде прихований за додатковими перешкодами. Більше того, внутрішня система просторового аналізу додатково налаштовує криву складності (Difficulty Curve), розміщуючи точки інтересу, ворогів та пастки виключно в тактично важливих вузлах цього графа.

Окрім класичних платформерів та шутерів, алгоритми процедурної генерації активно інтегруються у жанри з глибокою тактичною та стратегічною складовою. До таких можна віднести сучасні автобатлери та RPG-проекти.

Наприклад, у концепціях ігор, що базуються на механіках синтезу персонажів та постійному подоланні поверхів, процедурна генерація застосовується одразу на кількох рівнях абстракції. Алгоритм не лише створює симетричні або асиметричні бойові арени з різним тактичним рельєфом, але й процедурно підбирає хвилі супротивників, їхні синергії та нагороди [6].

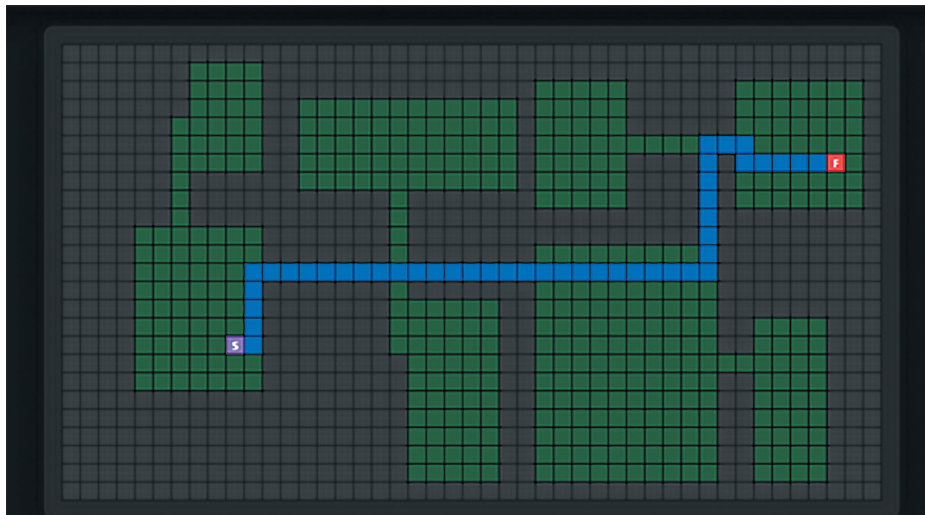
У таких випадках PCG виступає в ролі «віртуального ігрового режисера» (AI Director), який постійно аналізує математичний стан гри та адаптує згенерований контент під поточний прогрес користувача.

**Мета статті.** Метою статті є аналіз можливостей інтеграції алгоритмів процедурної генерації контенту та інтелектуального аналізу просторових структур для автоматичного оцінювання якості ігрових рівнів у веборієнтованих системах.

**Виклад основного матеріалу дослідження.** Дослідження базується на концепції представлення ігрового рівня у вигляді двовимірної матриці. Кожна клітинка матриці характеризує певний тип об'єкта в середовищі: прохідну область, перешкоду або спеціальну зону. Генерація рівнів може використовувати алгоритми BSP, клітинні автомати та рекурсивні методи побудови лабіринтів.

Після генерації рівня виконується структурний аналіз. Для визначення прохідності використовується алгоритм пошуку в ширину (BFS). Цей алгоритм дозволяє швидко перевірити маршрут між початковою та кінцевою точками та визначити оптимальну довжину шляху. Крім того, оцінюються доступність окремих зон карти.

Метод теплової карти використовується для визначення потенційних маршрутів руху. Шляхом багаторазового запуску алгоритму пошуку з випадковими змінами порядку сусідніх вузлів, що перевіряються, можна визначити області з найвищою активністю користувачів.



**Рис. 2. Згенероване BSP-підземелля**

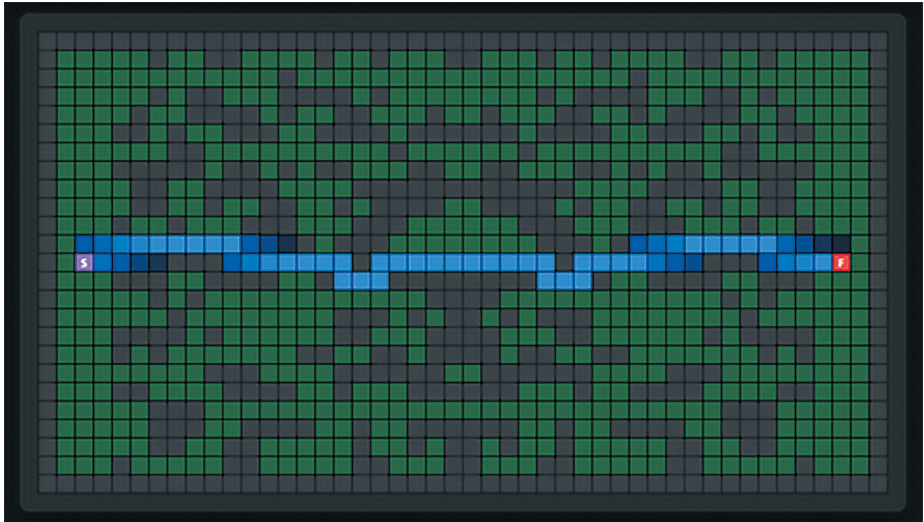
Джерело: авторська розробка

Ці дані можна використовувати для розміщення ресурсів, ворогів або контрольних точок. Один прохід стандартного BFS ідеально визначає найкоротший шлях, але з точки зору аналізу ігрового рівня цього недостатньо. У просторих печерах або на відкритих аренах може існувати безліч рівноцінних маршрутів, які мають однакову довжину. Щоб показати левел-дизайнеру всі можливі варіанти пересування гравця, було впроваджено інноваційну механіку «Теплової карти».

При активації цієї функції програма замість одного пошуку запускає алгоритм BFS 150 разів поспіль. Головним секретом цього підходу є просторова рандомізація. У класичному BFS сусіди завжди перевіряються в жорстко заданому порядку (наприклад: спочатку верх, потім низ, ліво, право). Через це алгоритм завжди обирає один і той самий маршрут.

У нашій реалізації перед перевіркою сусідів масив напрямків перемішується випадковим чином (`neighbors.sort() => Math.random() - 0.5`). Це призводить до того, що на відкритих просторах «хвиля» пошуку щоразу розтікається дещо іншим шляхом, зберігаючи при цьому загальну математичну довжину маршруту.

На рисунку 3 наведено приклад роботи теплової карти на базі згенерованих печер. Завдяки багаторазовому проходу маршрути зливаються, показуючи найімовірніші зони активності гравця.



**Рис. 3. Візуалізація теплової карти без використання динамічного шуму (пошук найкоротших шляхів)**

Джерело: авторська розробка

Важливим компонентом системи є моделювання динамічного шуму. Перед кожною ітерацією аналізу деякі доступні клітинки тимчасово блокуються. Це дозволяє визначити альтернативні маршрути та оцінити стійкість структури місцевості за змінних умов навколишнього середовища. Цей підхід пропонує глибший аналіз топології карти порівняно з традиційним пошуком найкоротшого шляху.

Для візуалізації результатів використовуються сучасні вебтехнології, такі як HTML5, CSS3 та JavaScript. Карти базуються на CSS Grid Layout, що дозволяє ефективно відображати двовимірні структури у браузері. Всі обчислення виконуються на стороні клієнта, що підвищує швидкість роботи системи та забезпечує інтерактивність.

Незважаючи на ефективність рандомізованої теплової карти, вона має суттєвий математичний недолік: алгоритм BFS завжди шукає найкоротший шлях. Якщо на карті є прямий широкий коридор і довга звивиста обхідна печера, тепла карта на 100% заповнить прямий коридор, повністю ігноруючи довгий маршрут, адже він не вигідний з точки зору оптимізації кроків.

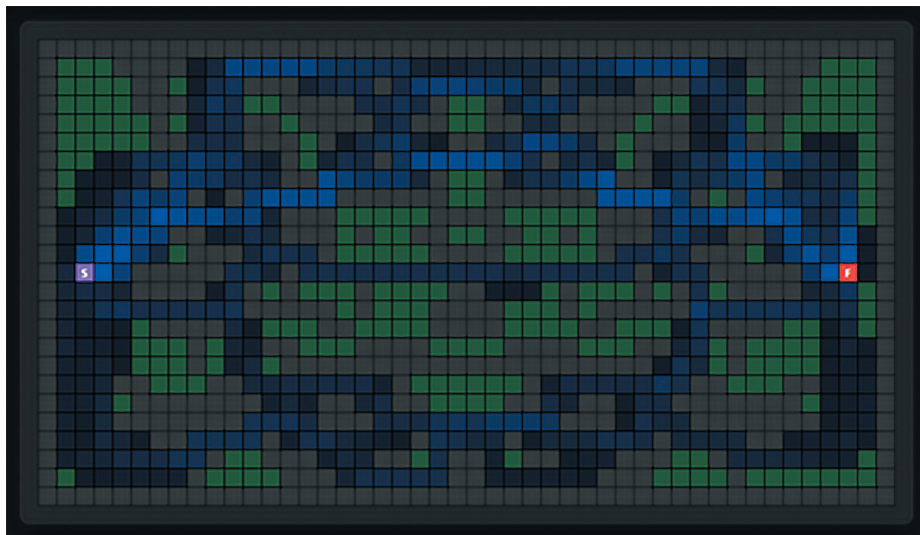
Однак у реальній грі гравець не завжди йде по прямій. Прямий шлях може бути перекритий вогнем супротивника, там може знаходитися

складний бос або тимчасова пастка. Щоб змусити систему аналізу знаходити альтернативні, довші, але тактично значущі обхідні шляхи, було розроблено концепцію Динамічного шуму.

Перед кожною з 150 ітерацій пошуку алгоритм створює тимчасову копію (зліпок) ігрового рівня (searchGrid). Далі програма проходить по всіх порожніх клітинках підлоги і з імовірністю 15% ( $\text{Math.random()} < 0.15$ ) перетворює їх на тимчасові непрохідні перешкоди (стіни). Оскільки цей «шум» генерується наново для кожної ітерації, він імітує динамічну зміну бойової ситуації: у першій ітерації прямий коридор виявляється заблокованим випадковим скупченням перешкод, і алгоритм BFS змушений шукати обхід через далеку печеру; у другій ітерації блокується обхідна печера, і маршрут прокладається іншим шляхом.

Цей підхід дозволяє виявити абсолютно всі можливі топологічні зв'язки між стартом і фінішем, незалежно від їхньої довжини. Якщо на карті є хоча б найменша щілина, яка дозволяє обійти перешкоди, динамічний шум змусить алгоритм знайти її.

На рисунку 4 чітко видно різницю під час активації функції «Динамічний шум». Теплова карта тепер покриває значно більшу площу рівня, демонструючи левел-дизайнеру повну картину доступних переміщень.



**Рис. 4. Візуалізація теплової карти з активацією динамічного шуму**

Джерело: авторська розробка

**Висновки.** Дослідження підходів до інтелектуального аналізу процедурно згенерованих ігрових рівнів у вебсистемах показало, що хоча сучасні методи генерації процедурного контенту дозволяють ефективно автоматизувати створення цифрового середовища, вони вимагають додаткових механізмів для контролю якості та перевірки відповідності результатів вимогам ігрового дизайну. Було виявлено, що використання алгоритмів пошуку шляхів, теплових карт та динамічного шуму покращує якість автоматично згенерованого контенту та забезпечує його ігрову зручність, збалансованість та функціональну придатність для подальшого використання.

У дослідженні визначено переваги поєднання процедурної генерації з методами просторового аналізу. Зокрема, використання алгоритму пошуку в ширину (BFS) дозволяє автоматично перевіряти доступність окремих областей рівня та визначати оптимальні маршрути між ключовими точками. Використання теплових карт дозволяє візуалізувати найімовірніші шляхи руху користувача, а введення динамічного шуму допомагає виявити альтернативні шляхи та оцінити стійкість структури рівня до змін умов навколишнього середовища.

Отримані результати підтверджують доцільність інтеграції алгоритмів процедурної генерації, аналізу графів та сучасних вебтехнологій для розробки інтерактивних систем автоматизованого проектування цифрових просторів. Практична перевага запропонованого підходу полягає в можливості скоротити час тестування ігрових рівнів, зняти навантаження з розробників та підвищити загальну якість кінцевого продукту.

Перспективні напрямки досліджень включають застосування методів машинного навчання та штучного інтелекту для автоматичного налаштування параметрів генерації контенту, прогнозування поведінки користувачів, адаптації ігрових середовищ до індивідуальних сценаріїв взаємодії та розробки самонавчальних систем для оцінки якості цифрових просторів. Це дозволяє створювати більш складні, реалістичні та персоналізовані віртуальні середовища наступного покоління.

## ЛІТЕРАТУРА

1. Togelius J., Yannakakis G. N., Stanley K. O., Browne C. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*. 2011. Vol. 3, No. 3. P. 172–186.
2. Short T. X., Adams T. *Procedural Generation in Game Design*. Boca Raton : CRC Press, 2017. 334 p.

3. Hendriks M., Meijer S., Van Der Velden J., Eliens A. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*. 2013. Vol. 9, No. 1. P. 1–22.
4. Smith G. Understanding procedural content generation: a design-centric analysis of the role of PCG in games. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014. P. 917–926.
5. Yu D. Spelunky. Los Angeles : Boss Fight Books, 2016. 192 p.
6. Shaker N., Togelius J., Nelson M. J. Procedural Content Generation in Games. Cham : Springer, 2016. 208 p.

## REFERENCES

1. Togelius J., Yannakakis G. N., Stanley K. O., Browne C. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*. 2011. Vol. 3, No. 3. P. 172–186.
2. Short T. X., Adams T. Procedural Generation in Game Design. Boca Raton : CRC Press, 2017. 334 p.
3. Hendriks M., Meijer S., Van Der Velden J., Eliens A. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*. 2013. Vol. 9, No. 1. P. 1–22.
4. Smith G. Understanding procedural content generation: a design-centric analysis of the role of PCG in games. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014. P. 917–926.
5. Yu D. Spelunky. Los Angeles : Boss Fight Books, 2016. 192 p.
6. Shaker N., Togelius J., Nelson M. J. Procedural Content Generation in Games. Cham : Springer, 2016. 208 p.