

УДК 004.4:[004.946.5:793.7

DOI: 10.31673/2412-9070.2025.018721

М. С. БОЙКО, студент;

ORCID: 0009-0002-0318-8494

С. В. ШУМИК, аспірант;

ORCID: 0009-0004-2477-5780

О. В. ПРОНЬКІН, аспірант,

ORCID: 0009-0004-8529-6919

Державний університет інформаційно-комунікаційних технологій, Київ

ЗАСТОСУВАННЯ МЕТОДІВ ГЕНЕРАЦІЇ ІГРОВИХ СТРУКТУР ДЛЯ ГРИ З ВІДКРИТИМ ВСЕСВІТОМ У ЖАНРІ RPG

Стаття розглядає важливі аспекти використання методів генерації ігрових структур для створення унікального ігрового процесу. У зв'язку з поширенням технологій генерації 3D світу у сучасних іграх, виникає необхідність вдосконалення процесів налаштувань та контролю за генерацією світу. Автори досліджують вплив використання алгоритмів генерації ігрових структур на покращення якості ігрового процесу, а також на оптимізацію гри. Стаття розглядає можливості застосування алгоритмів генерації ігрових структур для автоматизації процесів створення світу. Це містить в собі підбір оптимальних параметрів генерації, таких як швидкість, навантаження процесора та інші. Автори розглядають можливість створення моделей прогнозування, які дозволяють передбачати оптимальні умови генерації ігрових структур для конкретних комп'ютерів та користувачів. Основна увага приділяється усуненню дефектів генерації за допомогою алгоритмів створення об'єктів та аналізу великого обсягу даних, зібраних під час процесу відкриття гри для того, щоб було вибрано оптимальний алгоритм. Застосування методів генерації ігрових структур дозволяє виявити можливі неполадки під час генерації та знайти оптимальні шляхи для їх виправлення. Це не тільки підвищує якість генерації ігрових структур, зменшує відсоток навантаження на операційну систему, але й забезпечить стабільність цих процесів. Загальний висновок статті полягає в тому, що застосування методів генерації ігрових структур до створення світу в 3D грі має великий потенціал для покращення ефективності гри та оптимізації процесів її створення. Подальше дослідження в цьому напрямку може відкрити нові можливості для індустрії створення ігор та їх дизайну, сприяючи розвитку цифрового виробництва.

Ключові слова: 3D ігри, 3D моделі, ігрові структури, алгоритм, NPC, RPG, Open world, інформаційні системи, інформаційні технології, моделювання програмного забезпечення.

Вступ

З розвитком сучасних технологій та зростанням популярності відеоігор, особливо жанру RPG (рольових ігор), все більше уваги приділяється створенню ігрових світів з відкритим всесвітом. Такі ігри надають гравцям свободу дій, дозволяючи досліджувати величезні віртуальні простори та взаємодіяти з численними елементами навколишнього середовища. Проте, створення таких ігрових структур є надзвичайно складним та трудомістким процесом, що вимагає значних ресурсів і часу. Тому, постає питання автоматизації цього процесу за допомогою методів генерації ігрових структур.

Основна задача полягає в розробці методів, що дозволять автоматично створювати реалістичні та захоплюючі ігрові світи з відкритим всесвітом. Це включає генерацію ландшафтів, будівель, предметів, NPC (неігрових персонажів) та різноманітних подій, що відбуваються у грі. Такі методи мають враховувати багато параметрів: ігрову механіку, сюжетну лінію, естетику, а також забезпечувати належний рівень складності та інтерактивності для гравців.

© Бойко М. С., Шумик С. В., Пронькін О. В., 2025

Аналіз літературних даних і постановка проблеми

Існують різноманітні дослідження та розробки, що стосуються процедурної генерації в іграх з використанням алгоритмів або машинного навчання. Деякі з них зосереджені на уникненні помилок, шляхом довгих загрузок та генерації великої кількості об'єктів відразу, коли інші дивляться на динамічну генерацію, ризикуючи отримати помилки, хоч і позбавляються довгих навантажень на процесор. Деякі алгоритми намагаються знайти золоту середину, об'єднавши відразу два цих підходи.

Наприклад, у популярній грі Minecraft були об'єднанні обидва підходи, адже при генерації світу використовувалися три карти шуму. Одна — визначення висоти рельєфу, дві інші — температури і опадів. І саме поєднання останніх двох визначало біосферу. Висока температура та низька вологість – біом пустелі. Низька температура та низька вологість – біом тундри. Завдяки такому підходу використовуючи шум Перліна ця гра створювала стартовий проєкт малюнку 2D ігрової мапи, яка достатньо довго вантажилась та створювала початку точку, від якої місцевість гравця мала відштовхуватись. Далі, для того, щоб не створювати великий та майже нескінченний всесвіт, Minecraft спочатку генерує 3D мапу біля місцезнаходження гравця, роблячи процес миттєвим та незатратним. По мірі проходження гравця по світу, підходячи до ще не згенерованих частин мапи, гра починає ці частини генерувати, відштовхувавшись від стартової 2D мапи, яка була згадана раніше.

Ця розробка має потенціал забезпечити підвищену надійність, безпеку та ефективність для генерації нескінченних ігрових світів, за допомогою готового 2D малюнку, який у подальшому у світі використовується як шаблон для створення нових ігрових структур та біомів.

Проте тема залишається досить актуальною і потребує нових досліджень та вдосконалень, особливо з використанням сучасних технологій.

Мета і задачі дослідження

Мета статті полягає у розгляді особливостей та потенціалу генерації ігрових структур у комп'ютерних іграх через застосування та поєднання різних алгоритмів. Стаття спрямована на аналіз принципів та алгоритмів генерації комп'ютерних ігрових структур, виявлення методів їх покращення, які можуть бути застосовані для оптимізації процесу створення 3D рівнів та 3D світу.

Основна мета полягатиме в ідентифікації найбільш ефективних підходів до генерації ігрових світів та структур. Це включає в себе аналіз різних алгоритмів та їх поєднання, які зазвичай використовуються в іграх жанру PRG та Open world, що може бути використано для запобігання можливих негативних сценаріїв. Окрім цього, стаття також спрямована на визначення практичних переваг таких різних алгоритмів та їхнє потенційне впровадження для поліпшення рівня надійності та функціональності ігор.

Огляд методів генерації ігрових структур для гри

Існує чимало алгоритмів для генерації оточення в іграх, кожен з яких має свій унікальний підхід до створення ігрового середовища. Усі методи процедурної генерації в тій чи іншій мірі використовують генератори випадкових чисел, але ступінь контролюваності випадковості під час створення оточення може значно варіюватися. Серед найпоширеніших алгоритмів процедурної генерації оточення для ігор можна виділити наступні.

1. Алгоритм на основі клітинних автоматів (Cellular Automata). Цей метод можна застосовувати для двовимірних або тривимірних сіток клітин. Спочатку кожна клітина має стан "шлях" або "стіна". Далі задаються правила, які змінюють стани клітин залежно від станів їхніх сусідів. Протягом певної кількості ітерацій кожна клітина оновлює свій стан відповідно до цих правил. Наприклад, зазвичай використовується правило, за яким клітина стає стіною, якщо навколо неї є певна кількість стін. Після кількох ітерацій алгоритму згенероване підземелля

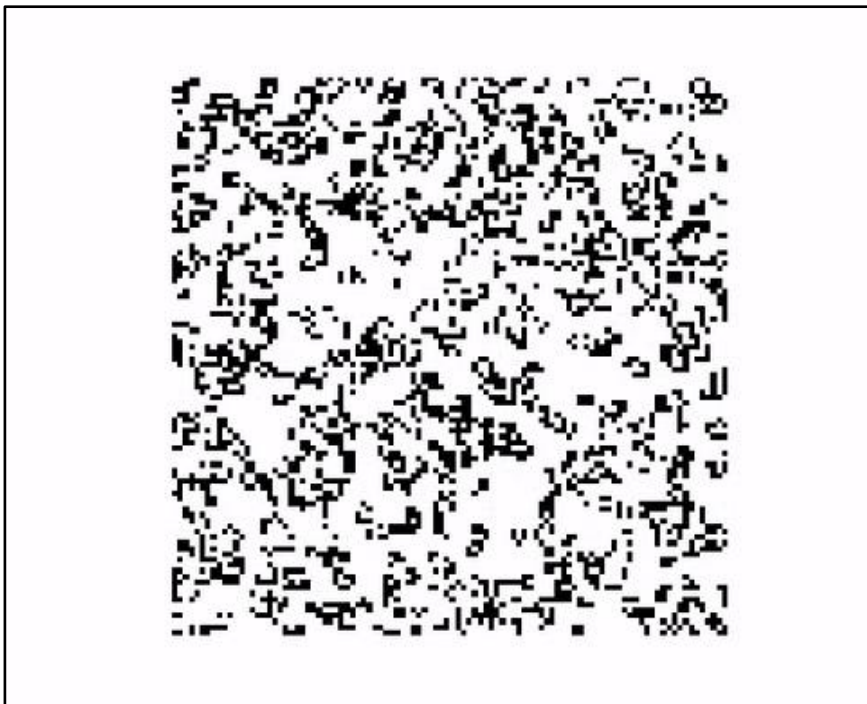


Рис. 1. Карта згенерована з використанням кліткового автомату

всесвітів. Ці алгоритми складаються з кількох основних етапів, що дозволяють створити цілісні ігрові світи з різноманітними локаціями.

- Генерація вузлів: Спочатку створюються вузли, які представляють різні важливі об'єкти у всесвіті, такі як міста, села, підземелля, ліси або гори.
- З'єднання вузлів: Вузли з'єднуються ребрами, що представляють дороги, стежки або переходи між різними локаціями. Це допомагає створити мережу зв'язків, яка забезпечує пересування гравця по світу.
- Оптимізація графа: Граф оптимізується для забезпечення певних властивостей, таких як зв'язність усіх локацій і відсутність перекриттів. Це означає, що кожна локація повинна бути доступною з будь-якої іншої локації, і при цьому вони не повинні накладатися одна на одну.



Рис. 2. Карта згенерована на основі сітки

можна перевірити на наявність кімнат і коридорів, а також додатково з'єднати незв'язані частини оточення.

Перевага цього алгоритму полягає в можливості створювати підземелля з різною структурою залежно від обраних правил, що робить його досить гнучким.

Недоліком є складність контролю над процесом генерації, оскільки зміна правил часто може призводити до несподіваних результатів.

2. Алгоритми генерації на основі графів (graph-based algorithms) застосовуються для створення ігрових 2D

світів. Перевага цього алгоритму полягає в можливості створювати зв'язний і добре структурований ігровий світ, де всі частини взаємопов'язані, що забезпечує логічність ігрового процесу. Крім того, рівень складності ігрового світу може бути налаштований, що дозволяє створювати як прості, так і складні світи.

• Перетворення графа на ігровий світ: Граф перетворюється на ігровий світ, де вузли замінюються на конкретні локації, а ребра – на дороги або інші шляхи пересування.

Крім того, рівень складності ігрового світу може бути налаштований, що дозволяє створювати як прості, так і складні світи.

Недоліком є те, що структура згенерованих локацій може бути менш органічною та менш випадковою порівняно з іншими методами генерації, що може призвести до більш передбачуваного і менш унікального ігрового світу.

3. Алгоритм BSP (Binary Space Partitioning) - метод поділу простору для створення ігрових світів, який базується на розділенні простору за допомогою площин. Цей підхід дозволяє ефективно створювати складні та цікаві локації для ігор. Ось як він працює:

- Початковий простір: Починається з початкового простору, який потрібно розбити на менші частини.
- Вибір площини: Обирається площина, яка розділить простір на дві частини. Це може бути будь-яка площина, але зазвичай обирається така, яка пройде через середину та розділить простір на більш-менш рівні частини.
- Рекурсивний процес: Кожна з отриманих частин простору поділяється знову за допомогою площини. Цей процес триває, поки кожна область не стане достатньо малою або не буде відповідати певним умовам зупинки.
- Розміщення об'єктів: Після розділення простору об'єкти (такі як стіни, об'єкти оточення тощо) розміщуються у кожній області відповідно до їх положення у просторі. Це забезпечує, що об'єкти правильно відображаються в ігровому світі.
- Відображення: Під час відображення ігрового світу відсутність об'єктів, які знаходяться за площиною розділення, дозволяє оптимізувати обробку графіки, оскільки ці об'єкти не впливають на зображення на екрані.

Алгоритм BSP допомагає зменшити обсяг обчислень, необхідних для відображення ігрового світу, оскільки дозволяє виключити області, які не видимі з поточної точки гравця. Це особливо корисно для ігор у 3D, де обчислення освітлення та відображення може бути дуже важким завданням.

Плюси алгоритму BSP включають в себе ефективне відображення ігрового світу шляхом розділення простору та простоту імплементації, що робить його привабливим для розробників з різним досвідом.

Проте, недоліки полягають у відсутності гнучкості для динамічних змін у світі і у проблемах з нерівномірним розподілом об'єктів, що може призвести до неефективного використання простору.



Рис. 3. Карта, згенерована на основі BSP алгоритму

4. Voronoi Diagrams - математичний метод, який використовується для генерації ігрових світів, заснованих на розділенні простору на рівні області за допомогою точок, які називаються "сім'ями" або "центрами". Кожен регіон простору в цій діаграмі належить одному центру і складається з точок, які найближче до цього центру, що створює набір областей, які зазвичай подібні до плям або клітин.

Генерація вузлів у випадку діаграм Вороного полягає у створенні точок, які визначають центри різноманітних областей у всесвіті. Потім проводяться лінії, що розділяють простір між сусідніми центрами, утворюючи межі між різними регіонами.

З'єднання вузлів відбувається шляхом встановлення зв'язків між сусідніми центрами, що визначають границі між різними областями. Оптимізація полягає в досягненні оптимальної структури діаграми, що забезпечує рівномірне розподілення областей та ефективне використання простору.

Потім ця діаграма може бути перетворена на ігровий світ, де кожен регіон відображається як окрема локація, а межі між ними - як шляхи чи границі. Перевагою цього методу є здатність створювати складні та цікаві ігрові світи, проте недоліком може бути менша органічність структури порівняно з іншими методами, що може призвести до менш унікальних ігрових світів.

Переваги використання діаграм Вороного для генерації ігрових світів включають природний вигляд границь, гнучкість у створенні різноманітних ландшафтів та швидкість побудови для прототипування.

Однак недоліки полягають у можливій неоднорідності границь, високих вимогах до ресурсів під час побудови та необхідності оптимізації для досягнення найкращої продуктивності та візуальної якості.

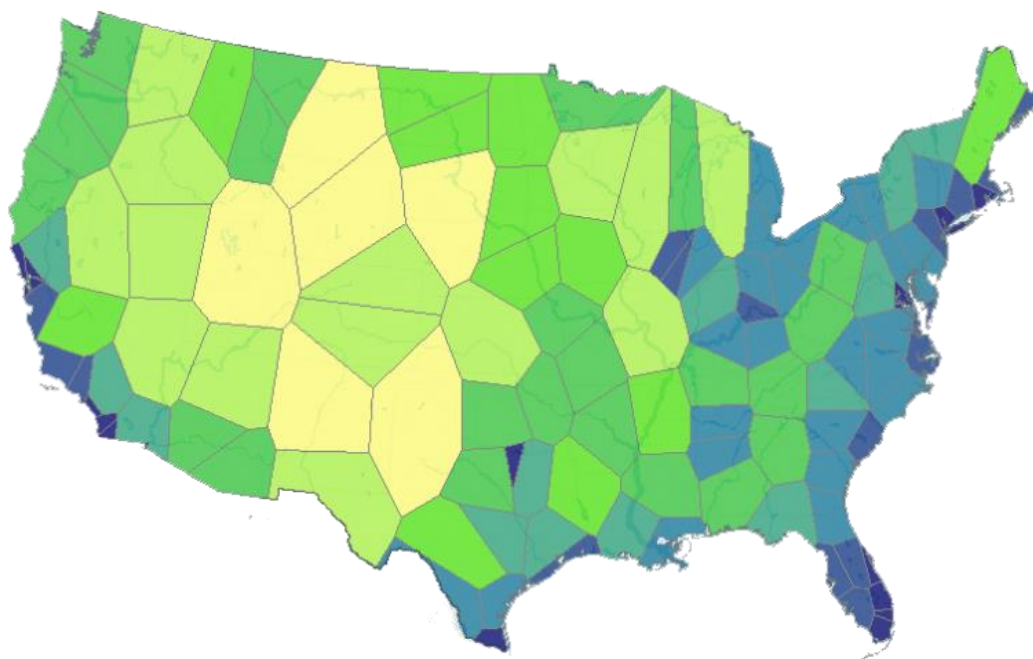


Рис. 4. Карта згенерована на основі алгоритму Voronoi Diagrams

Аналізуючи різноманітні алгоритми для створення ігрових світів, можна визначити, що кожен з них має свої переваги та обмеження. BSP відзначається простотою реалізації та можливістю створення добре структурованих світів, у той час як алгоритми на основі графів надають більшу гнучкість у створенні зв'язних ігрових світів. Проте, слід враховувати, що кожен з них має свої недоліки, такі як складність BSP при роботі з великими світами або неоднорідність границь у випадку використання діаграм Вороного. До того ж вибір конкретного алгоритму повинен враховувати потреби розробки, вимоги до продуктивності та візуальної якості, а також специфіку ігрового світу, який потрібно створити.

Розглянувши різні алгоритми для створення ігрових світів, можна зробити наступні висновки:

- **BSP (Binary Space Partitioning):** Цей алгоритм має простоту реалізації, що робить його привабливим для початківців. Він дозволяє створювати зв'язні та добре структуровані ігрові світи. Однак його складність зростає із збільшенням розміру ігрового світу і структура згенерованих локацій може бути менш органічною.
- **Cellular Automata:** Цей метод є відносно гнучким, оскільки може адаптуватися до різних умов і створювати різноманітні місцевості. Він дозволяє створювати складні ігрові світи зі складними взаємодіями між клітинами. Проте він вимагає більшої обробки та аналізу для досягнення бажаних результатів.
- **Graph-based algorithms:** Ці алгоритми надають більшу гнучкість і оптимізовані для створення зв'язних та структурованих ігрових світів з різноманітними локаціями. Вони можуть бути ефективними для великих і складних світів, якщо граф оптимізований належним чином.
- **Voronoi Diagrams:** Цей метод надає можливість створювати ігрові світи з різноманітними мапами, зокрема для генерації ландшафтів та розташування ресурсів. Він відносно швидкий у побудові та має природний вигляд границь. Проте його вимоги до ресурсів можуть бути високими і структура може бути менш органічною у порівнянні з іншими методами.

Для більш детального розуміння різниці алгоритмів та їх особливостей, варто ознайомитися з таблицею, в якій наведено порівняння алгоритмів для генерації ігрового світу.

Порівняльна характеристика алгоритмів генерації ігрових мап

Особливості	Binary Space Partitioning	Cellular Automata	Graph-based algorithms	Voronoi Diagrams
Ефективність	Дозволяє ефективно відобразити ігровий світ шляхом розділення простору на області	Забезпечує створення складних ігрових світів зі складними взаємодіями між клітинами	Дозволяє створювати зв'язні та структуровані ігрові світи з різноманітними локаціями	Використовується для створення різноманітних мап, зокрема для генерації ландшафтів та розташування ресурсів
Гнучкість	Менше гнучкості для динамічних змін у світі через потребу перерахунку розділення простору	Відносно гнучкий, оскільки може адаптуватися до різних умов і створювати різноманітні місцевості	Забезпечує більшу гнучкість, оскільки може враховувати зміни в структурі світу та додавати нові локації без значних перерв	Надає можливість створення різноманітних мап з рівнем деталізації, що залежить від кількості точок в діаграмі
Складність	Має простоту реалізації, що робить його привабливим для початківців	Вимагає більшої обробки та аналізу для досягнення бажаних результатів	Має середню складність, оскільки потребує встановлення зв'язків між локаціями та оптимізації графа	Вимагає складних обчислень для побудови діаграми, але може бути використаний для різноманітних застосувань у генерації світів

Ефективність обчислень	Може забезпечувати ефективне відображення, особливо для статичних ігрових світів	Вимагає значних обчислювальних ресурсів для обробки росту мапи ігрового світу	Може бути ефективним, особливо для великих і складних світів, якщо граф оптимізований належним чином	Ефективність може варіюватися в залежності від розміру діаграми та складності генерації
-------------------------------	--	---	--	---

Висновки

Отже, вибір конкретного алгоритму для генерації ігрового світу залежить від потреб розробки, рівня складності світу, вимог до швидкості та ресурсів, а також від вподобань щодо структури та природного вигляду локацій. Розробка ігрових світів є важливим і важким процесом, від якого залежить велика частина результату усєї гри, оскільки генерація ігрового світу впливає на:

- Продуктивність гри;
- Наповненість та інтерес у гравця до гри;
- Ефективне масштабування ігрового світу;
- Ефективна зміна існуючих модулів/функцій ігрового світу.

З огляду на аспекти, які залежать від генерації світу, правильний вибір ефективних підходів до генерації світів та структур дасть можливість значно покращити ефективність підтримки та розробки ігрових світів в жанрі RPG.

Подальше дослідження і особлива увага буде приділена алгоритму генерації на основі графів для ігрових структур для гри в жанрі RPG.

Список літератури

1. Buckley M. *Game Design: Level Generation Using Binary Space Partitioning* [Електронний ресурс] / Matt Buckley // Medium. – 2018. – Режим доступу до ресурсу: <https://medium.com/nice-things-ios-android-development/game-design-level-generation-using-binary-space-partitioning-e229230ab8b5>. (date of access: 01.06.2024).
2. Babiac M. *DISCOVERING GRAPH GENERATION ALGORITHMS* [Електронний ресурс] / M. Babiac, M. Karolis, R. Wattenhofer // NeSy-GeMs workshop. – 2023. – Режим доступу до ресурсу: <https://nesygems.github.io/assets/pdf/papers/discovering.pdf>. (date of access: 01.06.2024).
3. *An Intro to Graphs For Procedural World Generation* [Електронний ресурс] // Wintermutedigital. – 2020. – Режим доступу до ресурсу: <https://wintermutedigital.com/post/graphsforproceduralworlds/>. (date of access: 01.06.2024).
4. *Binary Space Partitioning* [Електронний ресурс] // Geeksforgeeks. – 2020. – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/binary-space-partitioning/>. (date of access: 01.06.2024).
5. Zhang C. *Cellular Automata DEVS: A Modeling, Simulation, and Visualization Environment* [Електронний ресурс] / C. Zhang, H. Sarjoughian // Digital library. – 2017. – Режим доступу до ресурсу: <https://dl.acm.org/doi/10.1145/3173519.3173534>. (date of access: 01.06.2024).
6. *The fascinating world of Voronoi diagrams* [Електронний ресурс] // Towards Data Science. – 2022. – Режим доступу до ресурсу: <https://towardsdatascience.com/the-fascinating-world-of-voronoi-diagrams-da8fc700fa1b>. (date of access: 01.06.2024).
7. Adams T. *Procedural Generation in Game Design* / T. Adams, T. Short., 2017. – 336 с.
8. Shekr N. *Procedural Content Generation in Games* / N. Shekr, J. Togelius, M. J. Nelson., 2018. – 237 с.
9. *Procedural Content Generation for Games: A Survey* [Електронний ресурс] / M. Hendrikx, S. Meijer, J. Van Der Velden, A. Iosup. – 2013. – Режим доступу до ресурсу: https://www.researchgate.net/publication/262327212_Procedural_Content_Generation_for_Games_A_Survey.

10. Schell J. *The Art of Game Design* / Jesse Schell., 2019. – 610 с. – (A K Peters/CRC Press).
11. *The fascinating world of Voronoi diagrams* [Електронний ресурс] // *Towards Data Science*. – 2022. – Режим доступу до ресурсу: <https://towardsdatascience.com/the-fascinating-world-of-voronoi-diagrams-da8fc700fa1b>.
12. Prusinkiewicz P. *The Algorithmic Beauty of Plants* / P. Prusinkiewicz, A. Lindenmayer., 2012. – 228 с. – (Springer New York, NY).

M. Boyko, S. Shumyk, O. Pronkin

APPLICATION OF GAME STRUCTURE GENERATION METHODS FOR AN OPEN WORLD GAME IN THE RPG GENRE

The article considers important aspects of using game structure generation methods to create a unique game process. In connection with the spread of 3D world generation technologies in modern games, there is a need to improve the processes of settings and control over world generation. The authors investigate the impact of using algorithms for generating game structures on improving the quality of the game process, as well as on game optimization.

The article considers the possibilities of using game structure generation algorithms to automate world creation processes. This includes selection of optimal generation parameters, such as speed, processor load, and others. The authors consider the possibility of creating prediction models that allow predicting the optimal conditions for the generation of game structures for specific computers and users.

The main focus is on eliminating generation defects using object creation algorithms and analyzing the large amount of data collected during the game discovery process in order to select the optimal algorithm. The use of methods for generating game structures allows you to identify possible problems during generation and find optimal ways to correct them. This not only increases the quality of the generation of game structures, but also reduces the percentage of the load on the operating system, but also ensures the stability of these processes.

The general conclusion of the article is that the application of game structure generation methods to the creation of the world in a 3D game has great potential for improving the game's efficiency and optimizing the processes of its creation. Further research in this direction could open up new opportunities for the game industry and game design, contributing to the development of digital production.

Keywords: 3D games, 3D models, game structures, algorithm, NPC, RPG, Open world, information systems, information technologies, software modeling.
