

ОСОБЛИВОСТІ СТВОРЕННЯ ЛАНДШАФТІВ У СЕРЕДОВИЩІ UNREAL ENGINE

Вінницький національний технічний університет

Анотація

Розглянуто різні підходи до створення ландшафтів у середовищі Unreal Engine та сучасних трендів у цій сфері. Проаналізовано роботу алгоритму процедурної генерації ландшафту у середовищі Unreal Engine та створено більш гнучкий алгоритм для процедурної генерації ландшафту за допомогою редактора Blueprint.

Ключові слова: ландшафт, відеоігри, процедурна генерація.

Abstract

Various approaches to landscape creation in the Unreal Engine environment and current trends in this field have been examined. The operation of the procedural landscape generation algorithm in Unreal Engine has been analyzed, and a more flexible algorithm for procedural landscape generation using the Blueprint editor has been developed.

Keywords: landscape, video games, procedural generation.

Вступ

При розробці відеоігор у середовищі Unreal Engine доволі часто необхідно створювати ландшафт місцевості. Існує кілька способів його створення: вручну у редакторі ландшафтів UE5, за допомогою Blueprint-кисті, за допомогою компоненту Procedural Mesh. Середовище Unreal Engine дозволяє розширювати функціонал процедурної генерації за допомогою Blueprint-функцій. Перед розробкою нового алгоритму варто розглянути існуючі підходи до створення тривимірних ландшафтів. В одній з робіт [1] приділено увагу еволюції ландшафтно́ї та морської екології від двовимірних до тривимірних моделей. Автори обговорюють, як новітні технології, такі як лазерна альтиметрія, акустичні сенсори та фотограмметрія, дозволяють створювати більш реалістичні 3D-моделі екосистем. Це відкриває нові можливості для аналізу складної структури наземних і морських середовищ. В іншій роботі [2] досліджується використання комп'ютерних технологій тривимірного моделювання у процесі проектування сучасних садово-ландшафтних екосистем. У ній проаналізовано, як такі технології, як CAD, сприяють підвищенню ефективності та точності ландшафтного дизайну. Інша стаття [3] досліджує використання тривимірного моделювання для створення міських ландшафтів у контексті розумних міст. Розглядаються алгоритми прискорення рендерингу, технології розпізнавання об'єктів та методи оптимізації для покращення якості й ефективності 3D-візуалізації.

Огляд алгоритму

Процурна генерація ландшафту особливо корисна для створення великих відкритих світів у відеоіграх, де ручне моделювання всіх деталей було б занадто затратним. Завдяки цьому алгоритму можна швидко генерувати різноманітні рельєфи, такі як гори, долини, пагорби чи рівнини, забезпечуючи реалістичність та унікальність кожної області. Це особливо важливо для open-world ігор, де гравець може досліджувати великі простори, а розробникам потрібно ефективно використовувати ресурси для створення динамічного середовища [4]. Окрім ігрової індустрії, цей підхід корисний у симуляціях та віртуальній реальності. Наприклад, у військових або наукових симуляторах потрібно створювати правдоподібні пейзажі для тренувань, а в архітектурних візуалізаціях — реалістичні ділянки місцевості для планування забудови. Автоматизоване генерування поверхонь дозволяє моделювати різні умови, змінюючи параметри рельєфу, такі як висота гір чи густота нерівностей, що дає змогу адаптувати ландшафт під конкретні задачі. Ще одна перевага процедурного створення ландшафту – це оптимізація продуктивності та економія пам'яті. Генерація може відбуватися динамічно під час гри, що зменшує навантаження на систему, оскільки не потрібно зберігати великі

попередньо змодельовані карти. Це корисно для мобільних ігор та VR-проектів, де важливий баланс між якістю графіки та продуктивністю. Для цього балансу у середовищі Unreal Engine існує технологія LOD (Level of Detail), яка дозволяє регулювати деталізацію об'єктів відносно відстані до них [5].

Алгоритм процедурної генерації ландшафту у Unreal Engine 5 працює шляхом створення сітки вершин, яку потім перетворюють у полігональну поверхню. Спочатку ініціалізується масив Vertices (рис. 1), де кожна вершина має координати (X, Y, Z). Значення X і Y визначають положення на площині, а Z розраховується за допомогою Perlin Noise, що створює плавні нерівності, імітуючи природний рельєф. Усі вершини розташовуються на рівних відстанях, задаючи базову структуру ландшафту. Далі алгоритм переходить до побудови трикутників, які формують поверхню. Використовуючи масив Triangles, вершини з'єднуються в групи по три, що створює полігони. Для кожної квадратної області сітки генерується два трикутники, які з'єднують чотири сусідні вершини. Це дозволяє створити суцільну поверхню, готову до текстуровання та подальшої обробки. Трикутники додаються у правильному порядку, щоб уникнути проблем з відображенням полігонів.

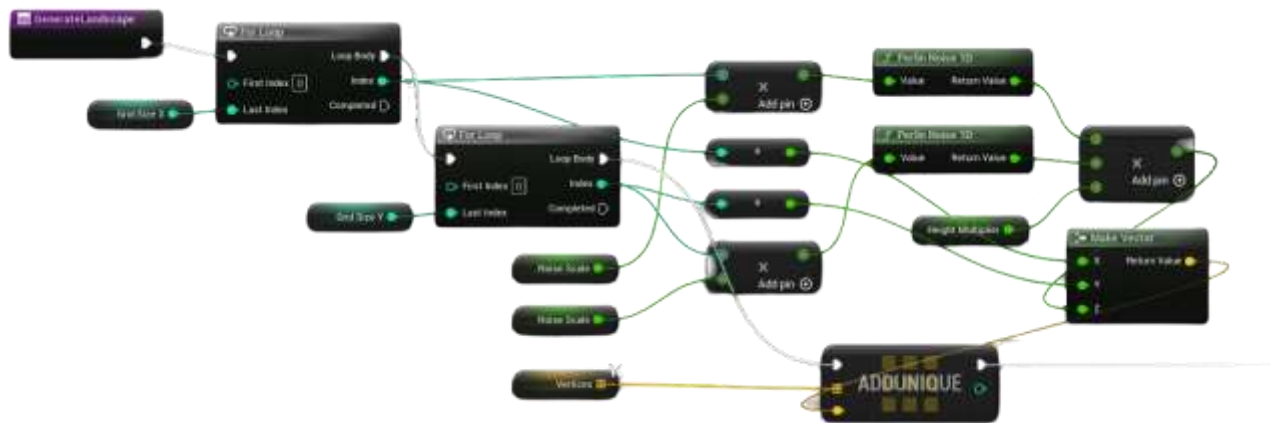


Рисунок 1 – Алгоритм для процедурної генерації ландшафту у редакторі Blueprint

Щоб ландшафт виглядав реалістичним, алгоритм додає текстурні координати та нормалі. UVs прив'язують текстуру до кожної вершини, використовуючи нормалізовані значення, що дозволяє рівномірно розподілити зображення поверхні. Normals визначають, як світло падає на ландшафт, що впливає на відображення тіней та загальну деталізацію. Додатково можна використовувати Tangents для коректного рендерингу матеріалів із картами нормалей. На завершальному етапі всі зібрані дані передаються в функцію Create Mesh Section, яка формує кінцевий об'єкт. Включення колізії дозволяє взаємодіяти з ландшафтом у грі, а можливість зміни параметрів у реальному часі дозволяє легко модифікувати світ. Такий підхід дає змогу створювати динамічні, реалістичні світи без необхідності вручну моделювати кожен елемент рельєфу.

Щоб зробити процедурно-генерований ландшафт сумісним із технологією LOD, потрібно створити кілька рівнів деталізації сітки, які динамічно перемикаються залежно від відстані камери. Для цього на етапі генерації можна розраховувати кілька версій ландшафту з різною щільністю полігонів: високополігональний варіант для близьких об'єктів та низькополігональний для далеких. Це можна зробити шляхом зменшення кількості вершин у сітці на основі квадродерева або мартенівського розбиття, де великі деталі згладжуються на великих дистанціях. LOD можна реалізувати в Blueprints через функцію Set LOD у Procedural Mesh Component або через Hierarchical LOD у Unreal Engine 5. Для цього потрібно згенерувати кілька варіантів масивів Vertices та Triangles, кожен із меншою деталізацією, і динамічно перемикати їх у Create Mesh Section. Можна також використовувати Nanite, який автоматично оптимізує рівень деталізації, але він більше підходить для статичних ландшафтів, тоді як класичний LOD гнучкіший для динамічних змін. Щоб уникнути помітних переходів між рівнями LOD потрібно використовувати morphing – поступове згладжування деталей при зміні рівня деталізації. Це досягається шляхом інтерполяції між позиціями вершин або використанням основаної на відстані теселяції, яка плавно додає або забирає полігони залежно від відстані до камери. Крім того, слід правильно налаштувати межі LOD, щоб уникнути занадто різких змін, що можуть призвести до помітного "стрибка" геометрії під час пересування камери.

Висновок

Розглянуто різні сучасні підходи до створення ландшафтів, зокрема, у середовищі Unreal Engine. Більш детально розглянуто можливості та переваги процедурної генерації ландшафтів та сумісних з нею технологій оптимізації ресурсів, доступних у середовищі Unreal Engine. Створено алгоритм для процедурної генерації ландшафту за допомогою редактора Blueprint. Цей алгоритм орієнтований на гнучке налаштування параметрів генерації у реальному часі, що дозволяє зменшити витрати зусиль на ручне корегування згенерованого ландшафту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Lepczyk C. A., Wedding L. M., Asner G. P., Pittman S. J., Goulden T., Linderman M. A., Wright R. Advancing landscape and seascape ecology from a 2D to a 3D science // BioScience. 2021. № 71(6). С. 596–608.
2. Tian Z. Application of computer 3D modeling technology in the simulation design of modern garden ecological landscape // Mathematical Problems in Engineering. 2022. № 2022(1). С. 7033261.
3. Wang L. 3D urban landscape rendering and optimization algorithm for smart city // Intelligent Decision Technologies. 2023. № 17(4). С. 943–958.
4. Landscape Technical Guide in Unreal Engine [Електронний ресурс] // Epic Games Developer Documentation. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/landscape-technical-guide-in-unreal-engine> (дата звернення: 22.03.2025).
5. Creating and Using LODs in Unreal Engine [Електронний ресурс] // Epic Games Developer Documentation. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/creating-and-using-lods-in-unreal-engine> (дата звернення: 22.03.2025).

Малініч Павло Павлович – аспірант кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: pavlo.malinich@vntu.edu.ua.

Войтко Вікторія Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: dekanfki@i.ua.

Pavlo Malinich – Postgraduate student of Software Engineering department, Vinnytsia National Technical University, Vinnytsia, e-mail: pavlo.malinich@vntu.edu.ua.

Viktoriia Voitko – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: dekanfki@i.ua.