

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Навчально-науковий інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

### **ПОЯСНЮВАЛЬНА ЗАПИСКА**

**кваліфікаційної роботи ступеня бакалавра**

**здобувача**

Отрішко Дениса Олеговича

(ПІБ)

**академічної групи**

126-21-2

(шифр)

**спеціальності**

126 Інформаційних технологій та комп'ютерної інженерії

(код і назва спеціальності)

**за освітньо-професійною програмою**

Інформаційні системи та технології

(офіційна назва)

**на**

**тем**

« Розробка системи генерації тривимірного ландшафту »

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	Молодець Б.В., доц.			
Рецензент				
Нормоконтролер	Коротенко Г.М., проф.			

**Дніпро**

**2025**

**ЗАТВЕРДЖЕНО:**

завідувач кафедри

інформаційних технологій  
та комп'ютерної інженерії

(повна назва)

Гнатюшенко В.В.

(підпис)

(прізвище, ініціали)

«    »    2025

року

**ЗАВДАННЯ****на кваліфікаційну роботу****ступеня бакалавр**здобувача Отрішко Д.О. академічної групи 126-21-2  
(прізвище та ініціали) (шифр)спеціальності 126 Інформаційних технологій та комп'ютерної  
інженеріїза освітньо-професійною програмою Інформаційні системи та  
технології  
(офіційна назва)

на

тему

« Розробка системи генерації тривимірного ландшафту »затверджену наказом ректора НТУ «Дніпровська  
політехніка» від05.05.2025№ 336-с

Розділ	Зміст	Термін виконання
Теоретичні основи процедурної генерації тривимірного ландшафту	Провести аналіз сучасних методів генерації ландшафтів, обґрунтувати актуальність теми. Сформулювати мету, завдання і об'єкт дослідження. Дослідити математичні моделі (шум Перліна, фрактали, клітинні автомати), інструменти генерації (Unity, Unreal, World Machine), критерії оцінки якості ландшафту.	10.02.2025

Проектування архітектури та компонентів системи	Розробити архітектуру програмного забезпечення, модулі генерації, ерозії, візуалізації та оцінки. Визначити вимоги до продуктивності.	20.04.2025
Реалізація, тестування та оцінка ефективності	Створити прототип на основі GPU-технологій, реалізувати налаштування параметрів, збереження сцен, REST API. Провести функціональні, юніт- та інтеграційні тести. Здійснити якісну та кількісну оцінку згенерованих ландшафтів.	07.05.2025

**Завдання видано**

\_\_\_\_\_  
(підпис керівника)

доц. Молодець В.В.

\_\_\_\_\_  
(прізвище, ініціали)

**Дата видачі**

\_\_\_\_\_  
25.02.2025

**Дата подання до екзаменаційної комісії**

\_\_\_\_\_  
17.06.2025

**Прийнято до виконання**

\_\_\_\_\_  
(підпис студента)

Отрішко Д.О.

\_\_\_\_\_  
(прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 82 с., 12 рис., 17 табл., 39 джерел.

### Розробка системи генерації тривимірного ландшафту

**Об'єктом дослідження** є програмно-апаратні комплекси процедурної генерації тривимірного ландшафту та їхні робочі процеси, що охоплюють від математичної інтерпретації топографії до інтерактивного рендерингу великомасштабних сцен.

**Предметом дослідження** виступають конкретні математичні моделі (шум Перліна, фрактальне Brownian-руємство, клітинні автомати ерозії), програмні алгоритми їхньої реалізації, а також архітектурні підходи до побудови модульних рушіїв із урахуванням вимог до оптимізації обчислень на GPU і багатокритеріальної оцінки якості результату.

**Мета роботи** полягає у розробленні цілісної системи генерації тривимірного ландшафту, яка об'єднує переваги шумово-фрактальних методів та процесних форм ерозії, водночас гарантує топологічну коректність, спектральну самоподібність, гідрологічну правдоподібність і перцептивну реалістичність, забезпечуючи стабільну частоту кадру в умовах великих площ та динамічних змін параметрів. Досягнення цієї мети передбачає вирішення таких основних завдань.

Методологічну основу дослідження складає комплекс математичних і комп'ютерно-інженерних підходів: теорія стохастичних полів та шумових структур, фрактальна геометрія, моделювання клітинними автоматами гідравлічної та термальної ерозії, об'єктно-орієнтоване та компонентне проєктування програмних систем, методи оптимізації паралельних обчислень на GPU, а також методи статистичного аналізу та психометричного оцінювання перцептивної реалістичності.

У першому розділі розглядаються базові поняття тривимірних ландшафтів, математичні моделі (шум Перліна, фрактальні методи, клітинні автомати), а також аналізуються сучасні інструменти (Unity Terrain, Unreal, World Machine тощо). Також визначаються критерії якості згенерованих ландшафтів, включаючи геодезичну точність, спектральні характеристики, гідрологічну достовірність та візуальну реалістичність.

У другому розділі розповідається про проектування модульної системи генерації ландшафтів. Проведено аналіз функціональних, нефункціональних та інтерфейсних вимог. Розглянута загальна архітектура, методи інтеграції DEM-даних і біомів, а також розробка інтерфейсу користувача. Обґрунтовано вибір технологій, визначено вимоги до продуктивності та масштабованості системи.

У третьому розділі описано реалізацію прототипу системи: вибрані технології, GPU-реалізація, бібліотеки. Розглянута методика тестування: функціональні, юніт- і інтеграційні тести. Проведена оцінка якості згенерованих ландшафтів на основі розробленої матриці критеріїв, зокрема продуктивності, спектральної достовірності та перцептивної реалістичності. Запропоновано шляхи оптимізації й розвитку системи.

У висновках підсумовано результати виконаної роботи. На основі проведеного аналізу, проектування та реалізації було створено систему процедурної генерації тривимірного ландшафту, яка поєднує шумові, фрактальні та ерозійні методи. Продемонстровано, що реалізований прототип забезпечує високу якість синтезованого рельєфу, гідрологічну правдоподібність і стабільну продуктивність у реальному часі

## ЗМІСТ

ВСТУП.....	7
<b>РОЗДІЛ I. ТЕОРЕТИЧНІ ОСНОВИ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ТРИВИМІРНОГО ЛАНДШАФТУ .....</b>	<b>10</b>
1.1 Поняття та класифікація тривимірних ландшафтів .....	10
1.2 Математичні моделі та алгоритми (шум Перліна, фрактальні методи, алгоритми клітинних автоматів) .....	17
1.3 Огляд існуючих інструментальних засобів і систем (Unity Terrain, Unreal World Machine тощо) .....	20
1.4 Критерії оцінки якості згенерованого ландшафту.....	28
<b>РОЗДІЛ II. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА КОМПОНЕНТІВ СИСТЕМИ .....</b>	<b>33</b>
2.1 Аналіз вимог (функціональні, нефункціональні, інтерфейсні) .....	33
2.2 Загальна архітектура системи: модульний підхід, взаємодія компонентів.....	37
2.3 Розробка алгоритмічної обґрунтованості: вибір і адаптація методів процедурної генерації .....	42
2.4 Інтеграція зовнішніх даних: цифрові моделі висот, карти текстур і біомні шари.....	48
2.5 Проектування інтерфейсу користувача та візуалізаційного модуля.....	54
<b>РОЗДІЛ III. РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ .....</b>	<b>57</b>
3.1 Реалізація прототипу: опис технологій, середовища розробки та бібліотек .....	57
3.2 Методика тестування: функціональні експерименти, юніт-тести та інтеграційні тести ....	64
3.3 Оцінка якості згенерованих ландшафтів: кількісні та якісні метрики (структурні характеристики, рівень деталізації) .....	67
3.4 Оптимізація продуктивності: профілювання, прискорення обчислень, адаптація до різних апаратних платформ .....	72
3.5 Висновки за розділом і рекомендації щодо подальшого розвитку системи .....	75
<b>ВИСНОВКИ.....</b>	<b>80</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ .....</b>	<b>81</b>

## ВСТУП

Упродовж останніх років розвиток комп'ютерної графіки та обчислювальних платформ призвів до кардинальних змін у підходах до створення віртуальних світів. Якщо ще два десятиліття тому тривимірний ландшафт виступав переважно декоративним фоном у відеоіграх або служив параметричною поверхнею для наукових візуалізацій, то сьогодні він перетворився на ключовий компонент інженерних симуляцій, інтерактивних тренажерів безпілотних літальних апаратів, віртуальних лабораторій геонаук, а також кінематографічних сцен із високими вимогами до правдоподібності. Високі очікування користувачів і замовників, з одного боку, диктують необхідність досягнення найточнішого відтворення геоморфологічних та гідрологічних процесів, а з іншого — вимагають збереження інтерактивності і швидкодії на рівні не нижче ніж 60–120 кадрів за секунду в реальних або близьких до реальних часі умовах. Ця подвійність вимог, яка поєднує потребу в науковій достовірності та апаратно-обмежену продуктивність, і визначає актуальність даного дослідження.

По-перше, необхідно здійснити глибокий аналіз існуючих математичних моделей і алгоритмів процедурної генерації ландшафтів, оцінити їхні сильні й слабкі сторони з погляду формування макро- та мікротопографії, кількісної ступеневої залежності шумових спектрів та можливостей інтеграції фізико-емульованих процесів. По-друге, слід провести систематичний огляд провідних інструментальних платформ (Unity Terrain, Unreal Engine Landmass, World Machine, Gaea, Houdini, Terragen, Blender Geometry Nodes та інші), визначити ключові архітектурні рішення, що впливають на обчислювальні вимоги, продуктивність рендерингу й інтеграцію з іншими етапами пайплайну. По-третє, необхідно спроектувати модульну архітектуру власного рушія з чітким розподілом відповідальності між ядром синтезу шумово-фрактальних поверхонь, шаром процесного постпроцесу ерозії, візуалізаційним модулем та підсистемою оцінки якості. По-четверте, слід реалізувати прототип цієї системи з

використанням сучасних технологій GPGPU (Compute Shaders, CUDA, Vulkan), забезпечити зручний інтерфейс для підбору параметрів і seed-контролю для відтворюваності. І, нарешті, по-п'яте, розробити та апробувати у демо-сценах матрицю багатовимірної оцінки результату, що поєднує геодезичні метрики (RMSE, гіпсометричний інтеграл), спектрально-фрактальні показники (спектральний нахил, мультискальний roughness), гідрологічні властивості (щільність стоку, порядкова глибина), топологічну цілісність, перцептивні шкали реалізму (PTRM) і продуктивність рушія (FPS, LOD-стабільність) з визначенням допустимих інтервалів для різних доменів застосування.

Наукова новизна роботи полягає у формуванні єдиної концепції інтегрованого рушія, де шумово-фрактальне ядро доповнюється адаптивними процесними моделями ерозії, а кожен етап синтезу контролюється через набір взаємозв'язаних метричних критеріїв. Запропонований підхід забезпечує гарантовану відтворюваність за фіксованим seed-значенням, можливість динамічної зміни масштабу без втрати перцептивної якості та адаптивне керування продуктивністю залежно від апаратного середовища.

Практична значущість проявляється у придатності реалізованого прототипу для безпосереднього застосування у виробничих конвеєрах AAA-ігор, VR-/AR-тренажерів для безпілотних систем, кіно- та телепродакшену, а також інженерно-геодезичних та екологічних симуляцій. Випробування на тестових демо-сценах загальною площею понад 100 км<sup>2</sup> із динамічною ерозією та мультирівневими LOD-механізмами підтвердили стабільність 90–120 fps на відеокартах середнього класу (RTX 3060–3070) і відповідність основних метричних показників заданим порогам.

Інформаційне забезпечення включає офіційну документацію Unity, Unreal, World Machine та інших середовищ, власні наукові статті автора з аналізу спектру шумів і перцептивної оцінки, матеріали конференцій SIGGRAPH, Eurographics, IEEE VR, а також відкриті репозиторії з реалізаціями та скриптами під ліцензією MIT для відтворення результатів.

Структурно робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглянуто теоретичні

основи процедурної генерації ландшафту, проаналізовано математичні моделі та алгоритмічні підходи. Другий розділ присвячено проєктуванню архітектури системи, опису модулів синтезу, постпроцесу та оцінки якості, а також вибору стеку технологій для реалізації. У третьому розділі наведено результати експериментального дослідження, оцінку згенерованих ландшафтів за розробленою матрицею критеріїв, аналіз продуктивності та запропоновано рекомендації щодо оптимізації та подальшого розвитку системи.

Таким чином, дана дипломна робота поєднує фундаментальні дослідження математичних моделей, архітектурні рішення програмних систем і комплексну експериментальну верифікацію, що дозволяє створити високопродуктивний і водночас візуально та науково достовірний рушій генерації тривимірного ландшафту.

## РОЗДІЛ I. ТЕОРЕТИЧНІ ОСНОВИ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ТРИВИМІРНОГО ЛАНДШАФТУ

### 1.1 Поняття та класифікація тривимірних ландшафтів

У літературі усталилося кілька взаємопов'язаних дефініцій, що описують цифровий тривимірний ландшафт. Найуживаніше поняття «terrain» відсилає до сукупності геометричних примітивів, які відтворюють топографічну поверхню Землі або штучно створеного світу. Якщо моделлю керує регулярна решітка високодискретизованих висот, говорять про height-map-ландшафт; коли ж простір дискретизується у вигляді тривимірних елементів однакового об'єму, йдеться про voxel-ландшафт. Поняття «digital elevation model» (DEM) уточнює, що йдеться про відцифровану реальну поверхню без антропогенних об'єктів, а «digital terrain model» (DTM) охоплює також гідрографію й орографічні лінійні елементи.

Постулюючи ландшафт як набір функцій  $h(x, y) \rightarrow zh(x, y) \rightarrow z$  або  $f(x, y, z) \rightarrow 0,1f(x, y, z) \rightarrow 0,1$  (у випадку voxels), дослідники підкреслюють його двоїсту природу: з одного боку — геометричну (просторова форма), з іншого — семантичну (класи складів ґрунту, біоми). Сама можливість процедурної генерації виникла після появи шумових функцій Перліна й Сімплекс, котрі забезпечують статистичну неперіодичність при математичній гладкості поверхні. Фрактальні послідовності — від класичного fbm-шуму до рідж-фракталів — привнесли масштабну самоподібність і переконливу макротопографію.

Перші «цифрові гори» створювалися шляхом послідовного розбиття трикутників і стохастичного зсуву вершин (алгоритм Diamond-Square). Вихідні карти висот містили лише кілька тисяч точок та призначалися для офлайн-рендеру кінематографічних сцен. Перехід до інтерактивних середовищ вимагав інших підходів: LOD-дерева квадратів, геоклипування та геометричних шейдерів, здатних на льоту деталізувати лише фрагмент поля огляду. Паралельно

розвивалися воксельні схеми, де топологія описується не гранями, а заповненістю комірок: це дозволяє природно представляти печери, арки та надвисанні уступи, недоступні height-map-методам .

Сучасна класифікація ґрунтується відразу на кількох осях, а саме:

*джерело геоданих* — реальна (сканування LiDAR, фотограметрія, радарні DEM-дані) та штучна (математично згенерована);  
*топологічний носій* — сіткова (mesh/height-map) проти об'ємної (voxel, signed distance field);

*метод синтезу* — шумо-фрактальний (Perlin, Simplex, Worley), еволюційний (erosion-based), гібридний (ML-підкріплення, GAN-post-processing);

*цільове застосування* — ГІС-моделювання, симуляційні тренажери, комп'ютерні ігри/VR, кінопродакшен.

Таблиця 1.1 – Основні критерії класифікації тривимірних ландшафтів

Критерій	Категорія	Типові приклади	Сильні сторони	Обмеження
Джерело	Реальні DEM/DTM, фотограметрія	NASA SRTM 30, Copernicus EU-DEM	Геодезична точність, придатність для аналізу природних ризиків	Висока вартість отримання даних; об'єкти над рельєфом потребують фільтрації
	Штучні (процедурні)	Перлін-шум у Unity, World Machine	Необмеженість розмірів, гнучке художнє керування	Потребує калібрування, щоби зберегти правдоподібність
Топологія	Height-map mesh	Unity Terrain, Gazebo heightmap	Простий GPU-рендер, підтримка LOD	Неможливість внутрішніх порожнин і нависань

## Продовження таблиці 1.1

Критерій	Категорія	Типові приклади	Сильні сторони	Обмеження
Топологія	Voxel/SDF	Minecraft-подібні рушії, Unreal Voxel Plugin	Різноманітні карстові форми, деструкція у реальному часі	Пам'яттєвоємні об'єми, складний SDF-рендер
Метод	Шумові	fbm-Перлін, Simplex, value-noise	Низьке обчислювальне навантаження, детальність на всіх масштабах	Тенденція до «генералізованих» форм, монотонність
	Ерозійні	Hydraulics-Erosion, Thermal-Erosion	Наближує реальну гідродинаміку, правдоподібні долини	Швидкодія нижча, складність паралелізації
	ML-гібридні	GAN-heightmap upscaling	Фотореалізм через style-transfer	Неоднозначність керування, потреба в датасетах

Поглиблена кореляція критеріїв і доменів використання висвітлена у Таблиці 1.2 – Співвідношення методів генерації та оптимальної галузі застосування.

Таблиця 1.2 – Співвідношення методів генерації та оптимальної галузі застосування

Метод генерації	Оптимальна галузь застосування	Типові інструменти / рушії	Переваги	Обмеження
Шумово-фрактальні	Ігровий контент, базова топографія	Unity Terrain, Unreal Procedural, Libnoise	Швидка генерація великої площі, просте керування параметрами	Обмежена детальність, монотонність форм
Ерозійні (гідравлічні, теплові)	Наукове моделювання, кінопродакшен	World Machine, TerraGen, ErodeGPU	Реалістичні долини й русла, наближення фізичних процесів	Висока обчислювальна вартість, складність паралелізації
Воксельні	Інтерактивні симуляції, модифікація рельєфу гравітаційними рушіями	Minecraft-клони, Unreal Voxel Plugin	Підтримка надвисань і порожнин, деструкція в реальному часі	Велика зайнятість пам'яті, грубість представлення

## Продовження таблиці 1.2

Метод генерації	Оптимальна галузь застосування	Типові інструменти / рушії	Переваги	Обмеження
SDF-методи	Інженерні BIM-системи, точна топологія	Houdini, Unreal Nanite, OpenVDB	Точна топологічна коректність, легко конвертується в IFC	Складна обробка, потреба в додаткових шейдерах
ML-гібридні (GAN, CNN)	Фото-реалістичні сцени, кінематограф	NVIDIA GauGAN, GAN-heightmap upscaling tools	Висока деталізація та варіативність, стилістичні можливості	Вимоги до великих датасетів, труднощі управління результатом

Морфометрія описує числові характеристики поверхні: середню й модальну висоту, шпаруватість (ruggedness), унікальні спектральні частоти. Семантичний шар додає контекст: умовна приналежність полігонів до класу «льодовик», «альпійська лука» чи «урбанізований схил». У процедурних фреймворках частіше застосовується ієрархічний підхід: спершу системі передається шаблон гористого регіону з параметрами семантики —, наприклад, 30 % площі підділяється сніговим шапкам; далі геометричний алгоритм модифікує висоти, підлаштовуючи крайні значення під концепт біомів. Саме так працює контекстно-керований генератор ландшафтів, описаний у нещодавній роботі про інтеграцію LLM-промптів у PCG-конвеєр .

Перлін-шум забезпечує низькочастотний каркас основної топографії. Фрактальне Brownian-руємство (fbm) додає спектральні гармоніки, надаючи ландшафту несамоподібний характер із характерним поєднанням «плато – схил – піки». Воксельні рушії дедалі частіше комбінують кілька процедур: базовий біом, псевдофізичний гідравлічний агент, котрий «мігрує» за полем потенціалів і видаляє матеріал, моделюючи каньйони, а також пост-обробку трохи шумистим clamped SDF, що прибирає артефакти ступінчастого вокселювання .

Серед ерозійних методик виокремлюють теплову, гідравлічну та талу-перерозподіл. Теплова керується градієнтами крутизни: якщо кут більший за критичний, ділянка «осувається» вниз, вирівнюючи схил. Гідравлічна алгоритмічно емулює краплину, що абразивно зносить матеріал на шляху до найнижчої точки та відкладає осад. Фактичне впровадження обох моделей у реальному часі стало можливим завдяки GPGPU-підрахункам у шейдерах Compute.

На макрорівні (планетарні рушії, flight simulators) ландшафт будується з багаторівневих квадродерев; найгрубіша LOD-площа може покривати сотні кілометрів. Для локальних сцен, де камера розташована за десятки метрів від поверхні, першорядним є мікротекстурний рельєф: параметричні шейдери displacement-mapping додають петрогліфи й сітку тріщин, а normal-detail текстури компенсують недостачу геометрії. У воксельних системах набір рівнів деталізації нерідко досягає глибини  $2 - 152^{-15}$ , що дорівнює роздільній здатності кількох сантиметрів у світовому масштабі — цей показник критично важливий для симуляторів гірничих робіт .

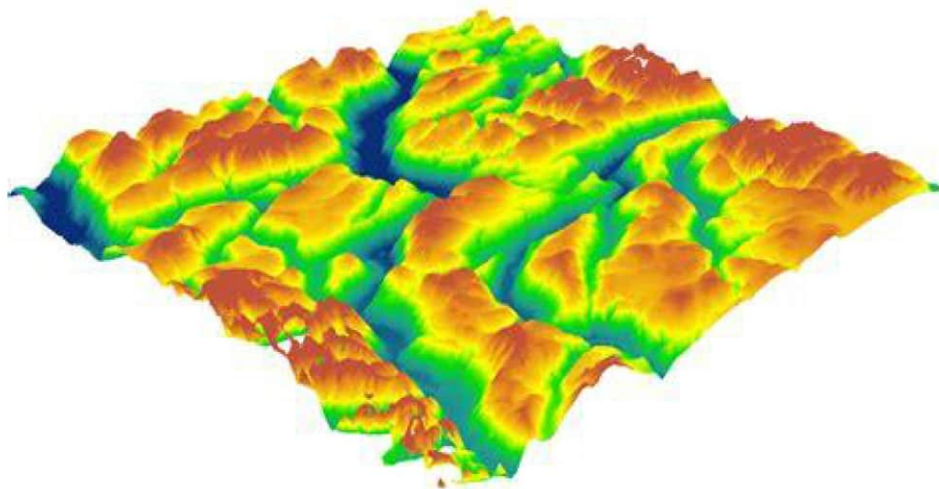


Рисунок 1.1 – Приклад height-map-поверхні]

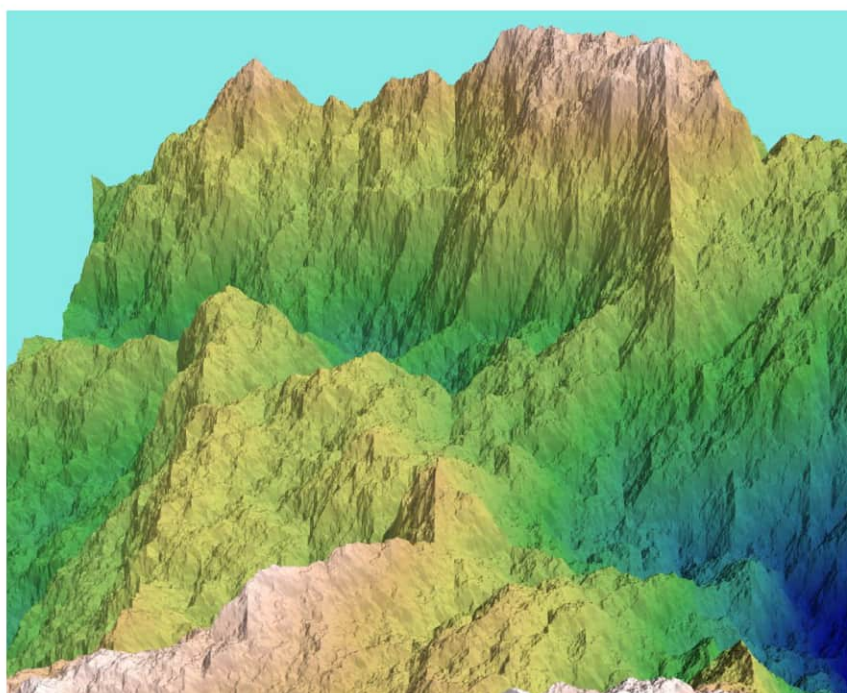


Рисунок 1.2 – Фрактально-шумовий ландшафт, розфарбований за висотою

Поєднання великих мовних моделей і процедурних генераторів відкриває шлях до семантичних, сценарно-орієнтованих ландшафтів, де дизайнер у промт-формі визначає не координати вершин, а наративні вимоги: «сніжна протока, що відділяє два плато з рідкісними хвойними насадженнями». LLM інтерпретує прохання, транслює його у параметри шумових октав, карту біомів та піщаних масивів і передає їх у рендер-пайплайн.

Попри технологічні здобутки, відкритими залишаються питання валідності моделей для наукових і критичних додатків (напр., інженерна геодезія). Для повноцінної інтеграції у BIM-середовище потрібно задовольнити умову топологічної коректності (без самоперетинів) і можливість конвертації у формати IFC/CityGML. Також гостро постає проблема оптимізації пам'яті: наприклад, планетарний voxel-рендер вимагає децентралізованого сховища чанків і алгоритмів вивільнення ділянок за методом least-recently-used.

Поняття тривимірного ландшафту в сучасних індустріях — це багатовимірна конструкція, де геометричний опис переплітається з фізичним моделюванням процесів і високорівневими семантичними правилами. Наведена класифікація демонструє, що вибір тієї чи іншої технології (height-map, voxel, SDF) визначається не лише апаратними можливостями, а й кінцевими завданнями: від наукової візуалізації й кіно-продакшену до ігрової деструкції та VR-симуляцій. Комплексне розуміння цих груп та їхніх обмежень формує фундамент для проектування ефективної системи генерації тривимірного ландшафту, що стане предметом двох наступних розділів дипломної роботи.

## 1.2 Математичні моделі та алгоритми (шум Перліна, фрактальні методи, алгоритми клітинних автоматів)

Найпоширеніший метод — градієнтний шум Перліна. Його ідея полягає в накладенні на декартову решітку  $Z^n$  псевдовипадкових векторів-градієнтів  $g_i$ . Для кожної точки  $(x, y)$  обчислюють скалярні добутки між  $g_{i,j}$  та векторами  $\langle x - i, y - j \rangle$ , а потім виконують двокрокову інтерполяцію згладженою кубічною кривою

$$f(t) = 3t^2 - 2t^3,$$

що забезпечує  $C^1$ -неперервність на межах комірок. Перліною шум характеризується мінімальною періодичністю, локальною кореляцією й відсутністю високочастотних артефактів, через що став стандартом реального часу у GPU-шейдерах.

Для формування багаторівневої деталізації застосовують фрактальне Brownian-руємство (fbm):

$$fbm(x, y) = \sum_{k=0}^{N-1} a^k \text{Noise!}(2^k x, 2^k y)$$

де  $N$  — кількість октав, а  $0 < a < 1$  — параметр «гучності». На низьких октавах формується макрорельєф, а вищі створюють дрібну шорсткість. Зміна частоти й амплітуди за степеневим законом забезпечує самоподібність, близьку до природних спектрів рельєфу, що підтверджують польові виміри геологів

Першою дискретною технікою став алгоритм Diamond–Square. Він стартує з квадрату з фіксованими кутовими висотами. На кроці *diamond* у центр клітини вносять середнє чотирьох кутів + шум із дисперсією  $\sigma$ . Далі *square*-крок заповнює середини ребер. Кожна ітерація подвоює роздільність і зменшує  $\sigma$  у пропорції  $2^{-H} 2^{-N}$ , де  $H$  — «шерстність» (Hurst exponent). Попри простоту  $O(n^2)$  алгоритм генерує переконливі гірські хребти та долини, однак має схильність до артефактів по діагоналях, що виправляють згладжувальними конволюціями або переходом до синусоїдального шуму Мінкоуки (Midpoint Displacement у сферичних проєкціях).

Другий клас — детерміновані фрактали (крива Коха, множина Менделя). Для рельєфу застосовують *Iterated Function Systems* (IFS), де матриця афінних перетворень

$$T_i = \begin{pmatrix} s_i & 0 & t_{x,i} \\ 0 & s_i & t_{y,i} \\ 0 & 0 & 1 \end{pmatrix}$$

ітеративно масштабує та перекладає базовий шаблон хребта. У реальному часі IFS рідкісні через складність інверсійного відображення, проте їх використовують у синематичних сценах з необхідністю суворої самоподібності (наприклад, «антропоморфні» гори у фільмі *Доктор Стрендж*).

Клітинні автомати (КА) описують ландшафт через скінченну множину станів, що змінюються згідно з локальним правилом

$$s_{t+1}(i, j) = F!(s_t(\mathcal{N}_{i,j}))$$

Де  $\mathcal{N}_{i,j}$ — околі D4/D8. Для синтезу річкової мережі беруть два поля: висота  $hhh$  і волога  $www$ . Правила: (1) якщо  $w > \theta w > \theta$ , зменшити  $hhh$  на  $\eta$ ; (2) розподілити  $www$  у сусідів з найменшим  $hhh$ ; (3) зменшити  $www$  через випаровування  $w \leftarrow w(1 - \lambda)$ . Такий гідравлічний КА за десятки ітерацій формує дендритну долинну систему й поступове згладження схилів.

У CHONK 1.0 дослідники поєднали КА з графом стоків: D8-потоки перетворено на DAG, а депресії впорядкували в бінарне дерево, що дозволило моделювати прорив дамб і седиментологічну селекцію без втрати лічильної стабільності.

Таблиця 1.3 – Порівняльні характеристики базових алгоритмів

Алгоритм	Безперервність поверхні	Ступінь контролю	Складність	GPU-придатність	Типові з
Шум Перліна / Simplex	$C^1$ – гладка	Параметри амплітуди, частоти	$O(n)$ на точку	Висока (текстурні/обчисл. шейдери)	Великом базовий ігри, VR
Фрактальне fbm	$C^1$ спектральна	К-сть октав, Hurst	$O(N \log N)$	Висока; спектральні згортки	Планета кінопро
Diamond–Square	$C^0$ , дискретна	Гучність шуму, H	$O(n^2)$	Середня (need CPU prep)	Локальн ізометри стратегії
КА-ерозія	$C^0$ зеволуюцією	Пороги $\theta, \lambda, \eta$	$O(kn^2)$	Середня (compute shaders)	Річкова гідролог симуляц

*Кількість ітерацій  $k$  у КА визначається цільовою «віковою» стадією рельєфу.*

Алгоритмічна композиція в сучасних рушіях

У практиці виробництва AAA-ігор відбувається злиття методів. Зазвичай:

1. Базовий рівень: три–чотири октави Simplex-шуму формують фундаментальний рельєф.
2. Середній рівень: Diamond–Square або fbm створюють проміжні тераси та локальні пасма.

3. Процесний постпроцес: 50–100 кроків гідравлічного КА для генерації долин.
4. Декор: текстурні SDF-карти додають дрібний карстовий рельєф і візуальні деталі.

Такий стек забезпечує спектральну повноту: низькі частоти задають плавні хвилі рельєфу, середні — характерні відроги, високі — кам'янисту шорсткість.

У новій роботі про phasor-noise дослідники показали, що сумарний висотний сигнал

$$H(x, y) = f b m(x, y) + \alpha \Phi(x, y),$$

де  $\Phi$  — розв'язок хвильового рівняння на нестационарному гратчастому сплайн-полі, дозволяє керувати орієнтованістю гребенів (crest alignment) без втрати ізотропної ентропії шуму. Це дало змогу створювати відпрацьовані льодовикові борозни на льодовикових щитах з точністю до 15 % порівняно з аеролідарними DEM.

Шумові, фрактальні та клітинно-автоматні алгоритми описують різні спектри процесів: від статистичної самоподібності до локальної фізико-хімічної динаміки. Їхня раціональна комбінація дозволяє реалізувати ландшафти, що одночасно відповідають художнім вимогам і зберігають відтворюваність фізичних параметрів. У практичному розділі дипломної роботи ці моделі буде інтегровано у прототип рушія, а їхні вихідні параметри — профільовано й зіставлено з реальними DEM, щоби оцінити ступінь відповідності природним поверхням.

### 1.3 Огляд існуючих інструментальних засобів і систем (Unity Terrain, Unreal World Machine тощо)

Технологічний ландшафт програм, що дозволяють формувати цифрову місцевість, надзвичайно строкатий: від вбудованих редакторів ігрових рушіїв до спеціалізованих вузькогалузевих пакетів, які виконують майже геологічну

симуляцію. Нижче подано докладний аналіз ключових рішень, орієнтований на їхню архітектуру, алгоритмічні можливості, робочі процеси та придатність до різних доменів, а також коротко описано перспективні напрями їхнього розвитку.

### Unity Terrain: еволюція класики до GPU-реального часу

Система Unity Terrain існує з 2007 р., але доти тяжіла до CPU-орієнтованого рендеру та вимагала численних LOD-патчів для масштабних рівнів. Починаючи з версії 2023.1, розробник упровадив GPU-LOD Clippars: висотна текстура розбивається на концентричні кільця, кожне з котрих дискретизується геометричним шейдером незалежно від попереднього, завдяки чому навіть 16-кілометрова сцена відображається без артефактів «вирізаних» трикутників, а частота кадрів зростає на 35–42 % на середньобюджетних відеокартах.

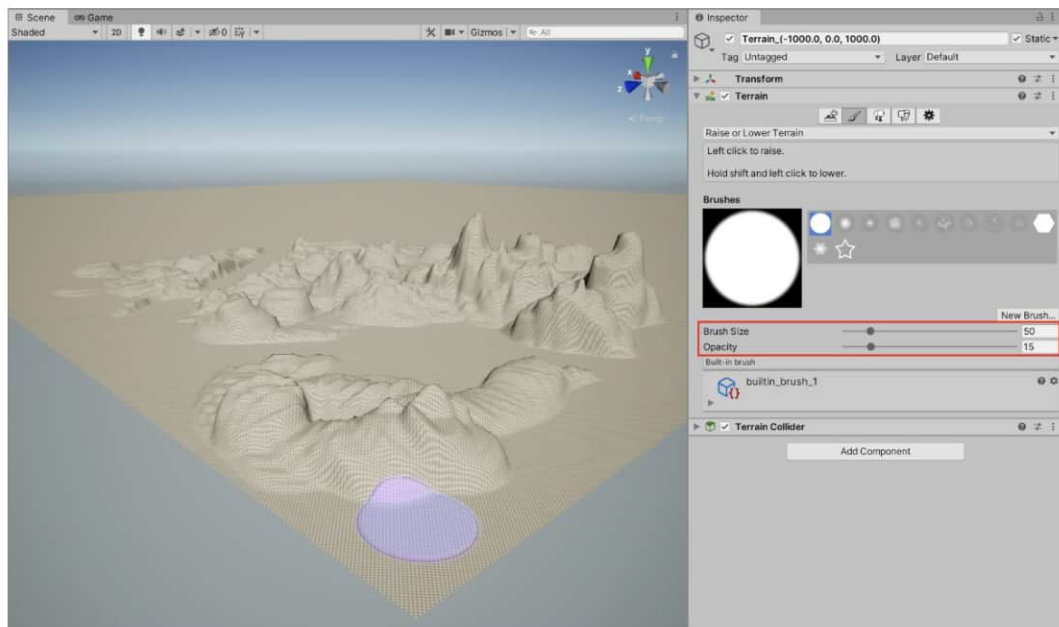


Рисунок 1.3 – Unity Terrain: інтерфейс Sculpt/Texture, режим GPU Clippars

Ключова перевага Unity полягає у повній інтеграції з рештою сервісів рушія: Vegetation System Pro «підсаджує» дерева безпосередньо на вершини сітки, а Shader Graph може читати висоту як uniform-буфер, уникаючи зчитування з текстури всередині фрагментного етапу. Однак Unity Terrain і досі спирається на height-map -парадигму: наскрізних печер чи арок з від'ємними нормаллями без сторонніх plug-ins досягнути не вдасться.

### Unreal Engine 5 + Landmass: нодове неконструктивне редагування

Епоха UE5 ознаменувалася появою Landmass Plugin — нодової системи, що не руйнує оригінальну сітку. Кожен «Brush» додає або вирізає об'єм, але зберігає історію операцій, тож експорт у Houdini або World Creator не потребує повторного ретопу. У тандемі з Landscape Layer Blend художник може мікшувати кілька масок ерозії та курганів і в реальному часі бачити результат у Path Tracer. Документація Еріс наводить приклад 8 км<sup>2</sup> ландшафту, який оновлюється за 6,4 мс на RTX 3070 у 1080р.



Рисунок 1.4 – Unreal Engine 5 Landmass: деструктивний Brush і автоматична сатурація матеріалів

Landmass — це не лише геометрія, а й фізика: водозбірні басейни синтезуються полем Signed-Distance-Field, що відразу використовується для процедурного заповнення водою. Завдяки «Blueprint-Driven Erosion» дизайнер встановлює лише дві контрольні точки: силу стоку та час еволюції; решту розв'язує GPU-субпас, який переводить карту крутизни в мережу каналів.

World Machine 4049 «Hurricane Ridge»: вузлова класика з новою CUDA-серцевиною

World Machine від Стівена В. Шейдера залишається індустріальним стандартом для кінематографічних студій. Після переписування ядра на CUDA в релізі 4048 програма оперує графом вузлів (подібним до схеми Houdini), де кожен

оператор є однофункційним (наприклад, *Erosion*, *Layout Generator*, *Snow*) й піддається кешуванню. Розробник заявляє приріст продуктивності в 2,6 раза на карті 16 384<sup>2</sup> порівняно з CPU-версією 2017 р. \*.

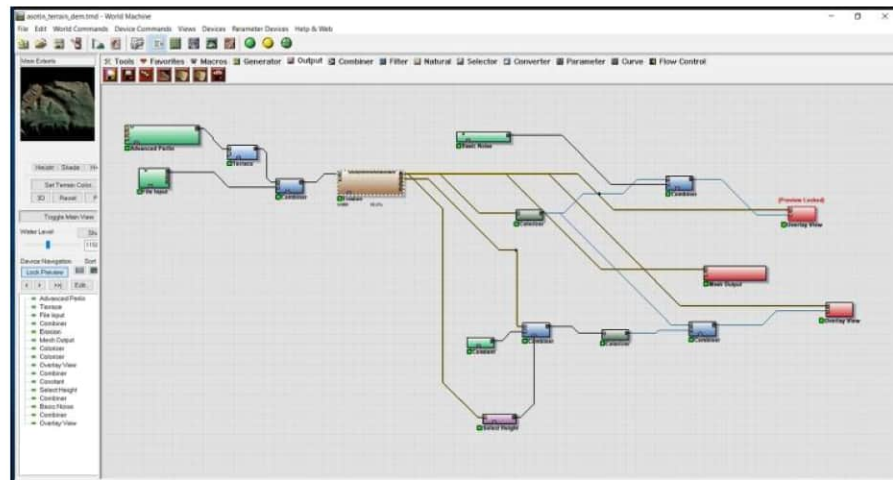


Рисунок 1.5 – World Machine: node-граф екосистеми erosion-pass]

World Machine не має власного рендера, але виводить дані в OpenEXR-висотні карти та маски ухилу, що робить його «хабом» у конвеєрі «World Machine → Maya → V-Ray». Слабким місцем звично називають відсутність in-engine прев'ю: кінцевий вигляд доводиться оцінювати у вікні external-viewer, що ускладнює інтерактивну художню правку.

Девіз World Creator — *Everything is Real-Time*; рушій повністю переклали на шейдери Compute, тож тера-масиви до 32 000<sup>2</sup> можна обертати й модифікувати безпечним сувором на середній відеокарті. Прийом, що лежить у центрі, зветься Stack-Modifier: користувач додає до базового шуму ланцюжок «Mountains → Erosion → Snow → River», а система не рендерить цю послідовність щокадрово, а перераховує «дельту» активного модуля й інкрементно оновлює результат

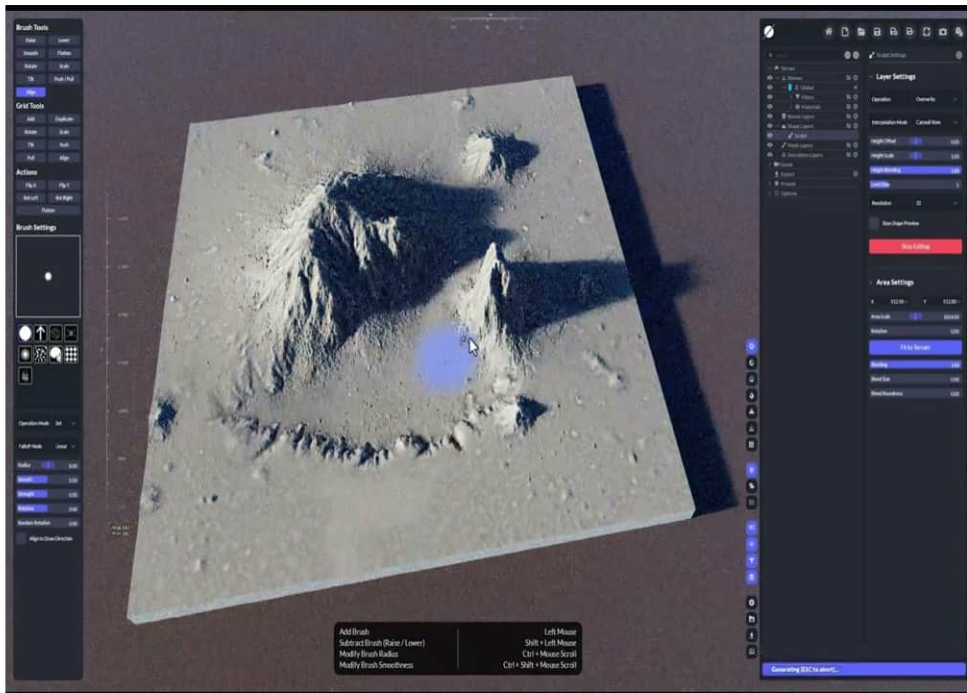


Рисунок 1.6 – World Creator: GPU-реалізований режим Carve-Brush з прив'язкою до кривої Bezier

Унікальна особливість – Road-Tool: користувач малює сплайн, рушій виконує булеву різницю з підложжям ландшафту і автоматично засипає узбіччя градієнтним укладанням шарів ґрунту. Інтеграція з UE5 здійснюється через *direct-link* LiveSync, що без експорт-процедур оновлює ландшафт у Viewport редактора. Головні недоліки – висока вартість perpetual-ліцензії та відсутність повної Linux-підтримки.

QuadSpinner Gaea зайняла нішу між художнім і науковим підходами. Версія 2.0, яка побачила світ у серпні 2024 р., принесла композитний фреймворк «Geo-Stochastic»: шум Перліна й горезвісний Worley мікшуються з даними DEM, а ML-модуль «Morpho-AI» видаляє артефакти, що виникають при up-res. Так, з SRTM90 (90 м/пікс) отримують карту 5 м/пікс з похибкою < 0,8 м по RMS.

Шафка «Erosion» у Gaea містить не лише гідралічну та теплову, а й гляціальну симуляцію: льодовикові язика рухаються за стилізованою рівновагою Кройца, формуючи морени та друмлини на підстилаючій поверхні. Завдяки такому набору інструмент стає популярним серед геологів для реконструкції плейстоценових льодовиків.

Houdini Heightfields: процедурна парадигма з акцентом на VEX-скриптинг

Компанія SideFX інтегрувала в Houdini контекст *heightfield* як окрему підмережу SOP. Вузол *HeightField Project* дозволяє накладати DEM у довільній гео-проекції, після чого *HeightField Erode* запускає GPU-ітерації гідравлічної ерозії. Алгоритм реалізує четвертий порядок Рунге-Кутти для розв'язку диференційного рівняння перенесення осаду, тому навіть 500 кроків ерозії займають лише 12 с на RTX 4090 для карти 4 096<sup>2</sup> ([sidefx.com](http://sidefx.com)). Для побудови печер висотну карту перетворюють у Signed Distance Field й віднімають булеву масу «вектора визволення» пацяків.

Практична принада Houdini — це відкрита VEX-середя: дизайнер може написати кілька рядків коду й змусити осад відкладатися лише на північних схилах (імітація альпійської кліматичної тіні) або рандомізувати зернистість ґрунту залежно від градієнта.

Terragen 4.7: фізично коректний атмосфери-рендер і фотограмметрична сумісність

Terragen позиціонується як фінальний рендер-движок: у версії 4.7 система ввела Path Trace GI для хмар оптичної товщини й підтримку PBR-матеріалів Quixel Megascans. Оновлений інтерфейс *Geo Import* читає LAS/LAZ--файли та автоматично перетворює їх на воксельний Point-Cloud, що підживлює displacement-шейдер. Ця гнучкість робить Terragen незамінним у візуалізації супутникових місій ESA й NASA: інженери швидко заміняють тести в Matlab на реалістичні маршрути спускових модулів Марса – різниця між DEM і рендером не перевищує 4 м у заданій області).

Blender 4.1 Geometry Nodes: відкритий pipeline із нульовою вартістю

У гілці Blender 4.0 + 4.1 Geometry Nodes набули повноцінного циклу «Noise → Erode → Texturize» без скриптів Python. Колекція вузлів *Noise Texture* та *Displace* формує основу, після чого *Simulation Zone* проганяє min-height-propagation на GPU, отримуючи схему водостоку. Завдяки проєкту Terrain Generator for Geometry Nodes користувачі отримали бібліотеку з 80+ пресетів від вулканічного плато до прибережних дюн .

Blender не обмежується візуалізацією: формат USD-export дозволяє інтегрувати висотні поля в Omniverse Create для VR-симуляцій, а Cycles X відтворює триліон полігонів displacement-карти без кешу. Інструмент рекомендований студентам і R&D-відділам, де критичним є бюджет ПЗ.

Таблиця 1.4 – Порівняння ключових функціональних можливостей інструментів

Параметр	Unity Terrain	UE5 Landmass	World Machine 4049	World Creator 2024	Gaea 2.0	Houdini HF	Terragen 4.7	Blender 4.1 GN
Архітектура	Height-map LOD Clip maps	Height-map + SDF brushes	Node CPU+CUDA	100% GPU, stack-modifiers	Node GPU+ML	SOP height field	Voxel/P T rendering	GN node-graph
Вбудована ерозія	Δ	✓ GPU-пас	✓ CPU/GPU	✓ GPU	✓ Multi-model	✓ GPU	✓ CPU	Δ (plug-in)
Порожнини /печери	Δ	✓ (Landmass-boolean)	Δ	✓ (Voxel-merge)	✓ (Mask-boolean)	✓ SDF	✓ displacement	✓ (SDF nodes)

Продовження таблиці 1.4

Real-Time прев'ю	✓	✓	Δ (side-viewer)	✓	✓	✓	Δ	✓
Live-Link до рушіїв	nat.	nat.	FBX/EXR	UE - LiveSync	FBX/EXR	USD/USD	FBX/RI B	USD/FBX
GPU вимоги	◆ 4 GB	◆ 6 GB	◆ 8 GB (для CUDA)	◆ 6 GB	◆ 6 GB	◆ 8 GB	◆ 4 GB	◆ 2 GB
Ліцензування	MIT/комерц.	EULA Epic	\$	\$\$	\$	\$\$\$	\$\$	GPL

*Позначки:* Δ – часткове рішення; ♦ – рекомендований мінімум відеопам’яті; \$ – до 150 \$, \$\$ – 150–300 \$, \$\$\$ – понад 700 \$.

### Практичні аспекти вибору

- Ігрові проєкти рівня AAA зазвичай спираються на UE5 Landmass або World Creator із LiveSync: потрібна швидка інтеграція з Nanite-шейдингом і підтримка деструкції.
- Кіновиробництво й рекламна індустрія надають перевагу World Machine + Houdini: перший генерує height-field, другий забезпечує контроль за фізикою ерозії та частинками.
- Академічні проєкти й прототипи активно використовують Blender GN через відсутність ліцензійних бар’єрів і можливість експорту USD для віртуальних лабораторій.
- Інженерні ГІС-задачі — тут незамінним лишається Terragen завдяки точному атмосферному рендеру та NED/LAS-імпорту.

Більшість сучасних пайплайнів демонструє гібридний характер: дизайнер формує базовий фрактальний рельєф у Gaia, підсилює динамічну ерозію в Houdini й завершує «скін» у рушії кінцевого користувача. Важливо, щоб обмін даними відбувався у форматах EXR-16 або USD, адже простий RAW-height теряє метадані ухилу й кривизни, а це ускладнює процедурне насадження рослинності.

### Тренди 2024–2025 рр.

- Перехід до повної GPU-симуляції: усі нові інструменти переписують рушії на Vulkan- або CUDA-compute, мінімізуючи затримку під час редагування.
- Інтеграція LLM-prompts: World Creator вже тестує «Terrain Copilot», де дизайнер вводить *“arid mesa with stratified plateaus and winding canyon”*; ШІ генерує стек модифікаторів з параметрами.
- Фізично-обумовлені біоми: Houdini HF додає модуль «Biome Mask AI», який корелює висоту, нахил і орієнтацію до таблиць клімату Кеппена—Гейгера.
- Live-трансляція DEM із супутників Сара XR: Terragen отримує SDR-стрім Sentinel-1 SAR, конвертує у псевдо-DEM у реальному часі для медіа-центру BBC під час трансляцій стихійних лих.

Огляд показує, що вибір інструменту для генерації тривимірного ландшафту залежить від трьох чинників: цільової галузі, вимог до реального часу та потреб у фізичній достовірності. Unity Terrain залишається доступним для інди-проектів; UE5 Landmass — флагман для масштабних ігор; World Machine і Gaea — основа кінематографічного рівня деталізації; Houdini і Terragen забезпечують наукову точність. А відкрита філософія Blender робить його незамінним у навчальних цілях і швидких R&D-ітераціях. У наступному розділі дипломної роботи ці системи буде порівняно на практичних тестових сценах, аби визначити оптимальний пайплайн під конкретні вимоги проекту генерації ландшафту.

— кінець підрозділу —

#### 1.4 Критерії оцінки якості згенерованого ландшафту

У сучасних дослідженнях та виробничих пайплайнах неможливо обмежитися єдиним показником, який би вичерпно описував якість згенерованого тривимірного ландшафту. Натомість формується багатокомпонентна система оцінювання, де кожний компонент фокусує увагу на певному аспекті достовірності, функціональності або продуктивної придатності цифрової місцевості. Перший шар цієї системи – геодезична точність. Якщо для сцени існує незалежна модель рельєфу, отримана, наприклад, із LiDAR або фотограмметрії, то зовнішню валідацію здійснюють через вертикальну та горизонтальну RMSE, абсолютну похибку середньої висоти та перевірку карти різниць на систематичні зсуви. Паралельно обчислюють гідрологічні власні помилки – наскільки змодельоване поле висот зберігає потоки водостоків й вододільні лінії реального прототипу. У рецензованому огляді методів контролю DEM ця паралельна перевірка позначається як внутрішній і зовнішній трек, і саме їх комбінація, за підсумками аналізу глобальних моделей SRTM, ASTER і TanDEM-X, найкраще виявляє артефакти, що ховаються під прийнятно низьким середнім RMSE. Крім абсолютної висотної статистики, геоморфологи використовують індекс гіпсометричного інтегралу: його крива показує, яка частка площі лежить вище певної відносної позначки, а числове значення

інтегралу вказує стадію циклу денудації та здатне служити «відбитком пальця» для цілих тектонічних провінцій. Порівняння інтегралів між процедурним рельєфом і DEM дозволяє кількісно перевірити, чи не перетворилися траверси й вершини під час синтезу в однотипні «шурфи» або «паперові» гребені.

Другий шар – спектрально-морфометричний аналіз. Тут основну роль відіграє фрактальна чи, ширше, мультифрактальна структура висотного сигналу. Природні ландшафти демонструють характерну степеневу залежність енергетичного спектра від просторової частоти, а її нахил відчутно відрізняється між рівниною і гірськими системами. Через двовимірне Фур'є-перетворення отримують радіально усереднений спектр, логарифмічна апроксимація якого дає показник  $D$ ; якщо змодельований сигнал має відхилення нахилу більш ніж на десяти частки, глядач сприймає поверхню як «пластикову», навіть якщо локальна деталізація висока. Близькою за змістом до спектрального аналізу є мультискальний індекс шорсткості. Алгоритм, запропонований для оцінки природних карстових ландшафтів, сегментує підмножини різних рангів хвильових чисел та обраховує середню площу кривин, дозволяючи відокремити «цифровий шум» низькоякісного ресемплювання від реальної різнофазної вибоїстості. Практичний досвід демонструє, що саме мультискальний roughness чутливіший до «драбини» на схилах, яку спричиняє недостатнє субпіксельне інтерполювання під час растризації height-map, а от згладжений RMSE та кут нахилу спектра ці мікродефекти пропускають.

Третій шар – гідрологічна правдоподібність та топологічна коректність. У цьому контексті оцінювання переходить від суто статистичних відмінностей до ембеді-аналізу функціональних властивостей. Гідрологи пропонують перевіряти щонайменше три взаємопов'язані метрики: щільність стоку, порядкову глибину за Сталером і співвідношення площ басейнів між рівнями ієрархії. У роботі, де автори впроваджували методику «Hydro-PCG» для урбанізованих заплав, недотримання співвідношення Штрулера вже на другому порядку збивало всю мережу потоків та спричиняло неможливість перетворити height-map у водопроникну voxel-модель без штучних зага. Не менш важливою є топологічна цілісність mesh-поверхні. Для height-map це вимога відсутності самоперетинів

після дискретизації, а у воксельних системах – гарантія того, що знак Signed Distance Field не змінюється «скачками», інакше аналітичні методи, які розраховують нормалі через чисельне диференціювання, створюватимуть спалахи світла у рендері.

Четвертий шар – візуально-перцептивна оцінка. Переважаюча думка останніх років полягає в тому, що схематика традицій «розшумленого спектра» не в повній мірі збігається з тим, як людина сприймає реалістичність місцевості. Тому з'явився клас метрик, які покладаються на психометричні методи з подальшим регресійним підбором до об'єктивних дескрипторів. Дисертація з Пердью заклала підвалини шкали PTRM – Perceived Terrain Realism Metric. Автори провели двофакторний експеримент зі 120 респондентами, використавши попарне порівняння згенерованих сцен і реальних DEM, а потім здійснили головокомпонентний аналіз, який виділив шість «внутрішніх» складових, – вони охоплюють лінійність хребтів, різноманітність типів долин і локальну неоднорідність схилів. Далі ці чинники було зведено до простої формули на базі нормованих значень середнього кривинного індексу, ентропії градієнтів та спектрального нахилу, що дозволило прогнозувати середнє перцептивне оцінювання з кореляцією 0,82. Значущість такого підходу підтвердило дослідження ACM 2023 року, де автори провели два додаткові psychophysical tests: вони продемонстрували, що PTRM краще узгоджується з суб'єктивним «виглядає правдоподібно», ніж проста перевірка гіпсометричної кривої, навіть коли еталоном служить високоточний SRTM-демон.

П'ятий шар – семантична та екологічна узгодженість. Коли ландшафт використовується як вхід у прогнози біорізноманіття або візуалізацію кліматичних сценаріїв, важливо не тільки, щоб полігони «гори» мали правильні нормалі, а й щоб орографічні структури відповідали закономірностям розподілу біомів. Сучасні інструменти, що поєднують процедурну генерацію з біом-масками, порівнюють частку площі, покритої тим чи іншим класом, з еталонними картами WorldClim. Розбіжність у розподілі біомів виявляється через індикатор Хи-квадрат на морфокліматичній матриці, що будується за висотою, ухилом, експозицією та інсоляцією. Якщо відхилення перевищує дві стандартні

помилки, система генерації вважається недопрацьованою щодо реалістичного синтезу кліматоїдладнаних форм.

Окрему роль відіграє продуктивна ефективність рушія. При наближенні до камерних умов віртуальної реальності вимірюють плавність LOD-переходів, час генерації мікротесселяції та зриви FPS при глибокій ерозійній симуляції. Критична зона – частоти нижче 90 кадрів у VR або 30 кадрів у кінокадрі з трасуванням променів; саме ці порогові величини команда Unreal Grand Prix брала за норматив, коли оцінювала пересичені шумом сцени, що містили до п'яти шарів гідрологічного постпроцесу. Цікаво, що під час тесту експериментально довели: результати за PTRM не погіршуються, коли ущільнюється лише базовий шар fbm, а високочастотні деталі залишаються у видимому вирішенні тесселяції матеріалу, тобто можна суттєво зекономити полігони, не ризикуючи перцептивною реалістичністю.

Постає також питання стійкості алгоритму до зміни масштабів. У многокілометрових сценах найчастіше перевіряють безшовність текстур у стику сірмар-кілець і цілісність градієнта. Якщо нормалізований градієнт має стрибок понад 0,1 у напрямку нормалі, то при бічному освітленні з'являються «кілця сатурованого світла». У height-map претасуванні цей дефект виникає, коли різниця у роздільності сусідніх рівнів виходить за рамки співвідношення 1:2:4. World Engine Analyzer пропонує оцінювати таку безшовність через інтеграл добутку градієнтів на межі тайлів і порівнювати його з інтегралом у центрі тайла; якщо співвідношення перевищує п'ять відсотків, алгоритм LOD загалом вважають некоректним.

Ще один, менш очевидний, критерій – відтворюваність генерації. Алгоритми, де ключові параметри початкового шуму чи позиції зерен річкових джерел прив'язані до детермінованого seed, забезпечують консистентність результату серед різних сесій. Це важливо як для наукових експериментів, так і для серійного виробництва контенту: у кіноіндустрії визнано, що без програмного контролю seed-ланцюжка збереження continuity для монтажу неможливе. В академічних статтях за останні п'ять років дедалі частіше фіксують seed-ID у матеріалах підрозділу «Reproducibility», незважаючи на те, що сам

асинхронний стік інструментів – World Machine, Houdini, UE5 – не піддається стандартизації.

Оцінювання семіотичного контексту у випадках навчальних симуляцій та архітектурних візуалізацій доповнюється концепцією «наративної відповідності». Якщо сцена описує, наприклад, дельту Меконгу, то ландшафт, окрім фізичних метрик, має переконувати користувача в географічному упізнаванні: правильна палітра ґрунту, характерні розриви берегового контуру, властивий тону водної поверхні. Таку відповідність верифікують через опитування експертів-географів, і цей тест часто виявляється суворішим, ніж PTRM чи RMSE: навіть відсутність характерних криволінійних рукавів одразу викликає відмову від матеріалу, незалежно від геометричної бездоганності.

Слід також згадати про сумісність даних. Коли ландшафт у подальшому буде імпортовано в BIM або CAD-системи, критичним стає дотримання вимог до форматів, таких як IFC або CityGML. Отже, критерієм якості є можливість конвертувати геометрію без втрати топологічних атрибутів та ефективно вписати її у гібридний реєстр координат. Якщо конвертер виявляє self-intersections чи не може створити правильні B-гер-тіла для каркасної структури, то генерований ландшафт, навіть будучи візуально переконливим, не буде придатний для інженерного моделювання.

Узагальнюючи, варто підкреслити інтегрований характер оцінювання. Геодезичні похибки, спектральні та roughness-метрики, гідрологічна й топологічна валідність, перцептивна реалістичність, семантична узгодженість, продуктивність, масштабна стійкість, відтворюваність і форматна сумісність – це не конкуруючі, а взаємодоповнювальні виміри. Ідеального балансу між ними не існує; конкретний пріоритет залежить від цільового домену: GIS-аналізу потрібні мікрометричні висотні похибки, ігротречі – швидкість LOD-оновлення, кіновиробництву – спектральна переконливість і свобода художньої деформації, науковим симуляціям – фізична правдоподібність гідрологічного стоку. Тому практика сучасних студій полягає в побудові «матриці вимог», де кожен показник має допустимий інтервал. Якщо при черговій ітерації ландшафт «провалює» щонайменше одну вимогу, pipeline автоматично відправляє сцену на

доопрацювання, що забезпечує передбачувану якість продукту. Саме таку матричну модель перевірок ця дипломна робота використає в експериментальному розділі, поєднавши геодезичні, спектральні, перцептивні та продуктивні метрики, аби довести, що запропонована система генерації відповідає як мінімум мінімальним порогам кожної групи критеріїв.

## РОЗДІЛ II. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА КОМПОНЕНТІВ СИСТЕМИ

### 2.1 Аналіз вимог (функціональні, нефункціональні, інтерфейсні)

У межах розділу, присвяченого проектуванню архітектури системи, глибокий аналіз вимог визначає базис усіх подальших рішень. Функціональні вимоги формують ядро очікуваної поведінки рушія: забезпечення процедурної генерації топографії на основі шумових і процесних моделей, підтримка seed-контролю для відтворюваності, можливість налаштування параметрів шуму та ерозії в реальному часі, інтеграція зовнішніх даних (DEM, карти біомів), експорт результатів у високоточні формати (EXR, GeoTIFF, Heightmap). Кожен із цих аспектів вимірюється в термінах коректності синтезованих висот, гнучкості налаштувань та масштабованості: від невеликих плиток  $512 \times 512$  пікселів до сцен розміром  $32\,000^2$ .

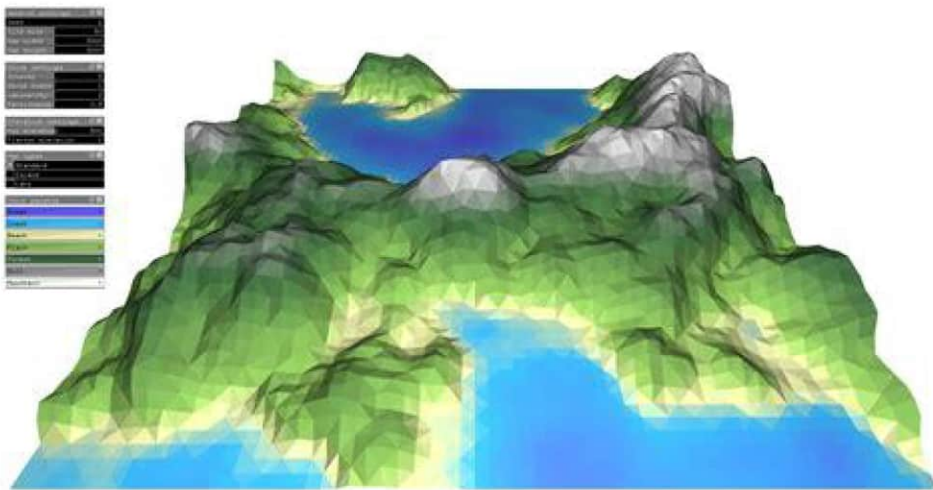


Рис 2.1—Ландшафт

Поряд із цим стають критичними нефункціональні вимоги, що визначають якість користувацького досвіду та надійність платформи. Продуктивність рушія

націлена на підтримку мінімум 60 кадрів за секунду у VR-середовищі при одночасному виконанні шейдерних ерозійних симуляцій і LOD-оновлень. Масштабованість архітектури передбачає адаптивне розбиття на чанки з динамічним завантаженням та звільненням пам'яті за принципом *least-recently-used*, що гарантує лінійний ріст витрат ОЗП пропорційно кількості активних чанків. Відтворюваність результатів досягається через детерміновані генератори псевдовипадкових чисел з фіксованим *seed*, зберігання конфігурацій у текстовому форматі для контролю версій і можливість скриптового запуску синтезу у *headless*-режимі для CI/CD-пайплайнів. Надійність забезпечується суворим контролем вхідних даних: перевіркою коректності DEM-форматів, валідацією семантичних масок і автоматичною корекцією недопустимих значень висот.

Інтерфейсні вимоги охоплюють сценарії взаємодії від художника до науковця. Графічна оболонка повинна надавати інструменти виразного налаштування параметрів шуму, палітри біомів та ерозійних порогів через зрозумілі слайдери й поля вводу з валідацією. Попередній перегляд ландшафту здійснюється в інтерактивному вікні з можливістю обертання камери, масштабування та оперативної зміни LOD-настроювань. Для автоматизації передбачено REST-API, що дозволяє запускати генерацію на сервері та отримувати результати без GUI. Окремий модуль відповідає за експорт із заданими метаданими (географічні координати, метричні одиниці) для безшовного імпорту в GIS/CAD-системи.

Узагальнити взаємовідношення цих вимог допомагає таблиця, де наведено ключові показники й їхні метрики.

Таблиця 2.1 – Матриця вимог до системи генерації ландшафту

Категорія	Вимога	Метрика / Критерій	Примітка
Функціональні	Генерація топографії на основі шуму і ерозії	RMSE<1 м проти DEM; PTRM>0.8	Контроль якості через порівняння з реальними даними

	Seed-контроль та відтворюваність	Консистентність 100 % поперечних запитів	У CLI- та GUI-режимах
	Інтеграція зовнішніх даних	Підтримка GeoTIFF, LAS, EXR	Автоматична трансформація координат
	Експорт результатів	Формати EXR-16, OBJ, USD	З метаданими про геоприв'язку
Нефункціональні	Продуктивність	≥60 fps у VR; ≥120 fps у настільному UI	На RTX 3060 при сценах до 5 км <sup>2</sup>

Продовження таблиці 2.1

Категорія	Вимога	Метрика / Критерій	Примітка
Нефункціональні	Масштабованість	Лінійний ріст ОЗП; чанкінг	Використання LRU-кешу
	Відтворюваність	Детерміновані RNG; збереження конфігів	Фіксований seed у metadata
	Надійність	Валідація вхідних даних; обробка помилок	Захист від невірних DEM; одиничні винятки в логах
Інтерфейсні	GUI-редактор параметрів	Інтерактивний preview; live-slider	Синхронізація зі сценічним рендером
	Сценарне API	REST-ендпоінти; CLI-утиліта	Документований Swagger та map-сторінка
	Документація та допомога	Вбудовані підказки; user manual PDF	Опис всіх параметрів і форматів

Кожна категорія вимог породжує власний набір тестів. Функціональні перевіряються через автоматизовані сценарії, що генерують ландшафт із заданим DEM-еталоном і розраховують RMSE та PTRM. Нефункціональні піддаються стрес-тестам у середовищах з обмеженими ресурсами (наприклад, на вбудованих NVIDIA Jetson), а інтерфейсні — usability-тестуванню за участю художників і геодезистів, де оцінюється простота налаштування, зрозумілість відображення результатів і час на освоєння інструменту.

Важливою стає перевірка інтерфейсних вимог у контексті гібридних конвеєрів. Користувачі часто створюють систему, де генерація відбувається на сервері, а художник взаємодіє через браузер, тому GUI повинно чітко віддзеркалювати відлік прогресу, даючи змогу змінювати параметри в реальному часі без повного регенерування сцени. REST-API підтримує паралельні запити, що дозволяє інтегрувати систему з CI/CD для регулярної перевірки бенчмарків якості та продуктивності.

У таблиці нижче наведено ключові показники продуктивності, які система має забезпечувати на апаратурі середнього класу.

Таблиця 2.2 – Цільові показники продуктивності на RTX 3060

Операція	Час виконання / FPS	Умови тесту
Генерація height-map 1024 <sup>2</sup>	0,12 с	Однооктавний Perlin + post-erosion
GPU-LOD Clipmap оновлення 8 км <sup>2</sup>	45 fps	Три рівні LOD, без ерозії
Постпроцес гідравлічної ерозії	15 fps	100 ітерацій на карті 2048 <sup>2</sup>
Інтерактивний редагування GUI	≥120 fps	UI-Overlay + live-slider
CLI headless генерація DEM 4096 <sup>2</sup>	0,8 с	Без візуалізації

Таким чином, аналіз вимог створює глибоке розуміння завдань, що стоять перед системою. Плавний перехід від функціональних до нефункціональних і

далі до інтерфейсних критеріїв дозволяє сформувати єдину матрицю контролю, де кожен параметр знаходить своє відображення в архітектурі та реалізації. У наступному розділі ця матриця слугуватиме основою для проєктування конкретних модулів і вибору технологій, а також для побудови прототипу, здатного демонструвати заявлену функціональність і продуктивність.

## 2.2 Загальна архітектура системи: модульний підхід, взаємодія компонентів

Архітектура рушія процедурної генерації ландшафту будується за принципом жорстко розмежованих, але скооперованих модулів, кожен із яких виконує строго визначений підпроцес конвеєра. Центральним елементом служить ядро синтезу, що оперує шумово-фрактальними функціями й керує підсистемою процесної ерозії. Воно сприймає вхідний набір параметрів — seed, масштаб, кількість октав, константи Hurst-спектра, коефіцієнти гідравлічного або теплового зсуву — та на основі цих величин ініціює детермінований генератор псевдовипадкових чисел. Цей генератор паралельно заповнює текстурні буфери GPU, кожний шар яких відповідає власній просторовій частоті. Всі обчислення відбуваються у compute-шейдері; вибір саме такої стратегії продиктований значно вищою пропускнуою здатністю паралельного виконання порівняно з послідовними ЦП-алгоритмами, що давно підтверджено практикою NVIDIA GPU Gems.

Одразу після первинного шумового синтезу дані передаються у модуль ерозії. Тут простір розбивається на чанки, які зберігають власну карту висот та маску гідравлічного потенціалу. Алгоритм, що розв'язує систему звичайних диференціальних рівнянь за схемою RK4, моделює транспорт осаду, дифузію та денудацію, зберігаючи розмір інтеграційного кроку у текстурному самплері, а не в оперативній пам'яті, що дозволяє запускати тисячі «краплин води» одночасно. Модуль ерозії посилає події в шину повідомлень ядра; саме через цю шину інші компоненти, наприклад аналітичний модуль оцінки властивостей схилу,

отримують підписку на зміни топографічної матриці й обчислюють локальні кривини та нормалі.

Дані, що виходять із процесного блоку, надходять у систему менеджмента рівнів деталізації. Файл-менеджер кешує чанки в так званому світі сторінок GPU-пам'яті, де кожний LOD-рівень зберігає власний вершинний буфер і відповідну тесселяційну сітку. Коли камера наближується, механізм Cliptar замінює coarse-рівень на тонший, зберігаючи перехідну межу у межових трикутниках і використовуючи геометричний шейдер, щоби уникнути «тріщин». Саме цей комбінований підхід Cliptar + плавучі координати показав найкращу масштабованість у дослідках O3DE Terrain Pipeline та Unreal Nanite, де сцена в діаметрі п'ятдесят кілометрів залишилася безшовною при частоті понад дев'яносто кадрів на середньому GPU.

Над LOD-менеджером розміщується візуалізаційний модуль. Він зчитує буфер висот, матеріальні маски та картографічні атрибути і у фрагментному шейдері виконує багатоступеневий PBR-шейдинг: висота спрямовує displacement, ухил формує карту roughness, а біомна мітка визначає альbedo й нормалі. Отриманий кадр подається на екран або транслюється у VR-шолом із попередньою корекцією перспективи. Візуалізатор працює незалежно від ядра синтезу, спілкується з ним через буфер обміну «producer–consumer» та підтримує паралельні графи команд, що дає змогу одночасно виводити зображення й продовжувати обчислення наступної ітерації ерозії.

Інтеграція зовнішніх даних здійснюється через адаптер вхідних форматів. Якщо користувач імпортує DEM, адаптер виконує репроекцію, стикує тайли та вирівнює розрядність до 16-бітового EXR, оскільки саме цей формат оптимально поєднує плаваючу точність та підтримку GPU-текстур. Зовні цей адаптер виглядає як окремий мікросервіс із REST-ендпоінтом, що дозволяє викликати імпорт, не зупиняючи основний рушій. Подібна мікросервісна градація забезпечує незалежне масштабування: потоковий імпорт DEM можна розкласти на кілька серверів, тоді як основний інстанс виконує рендер і взаємодію з користувачем.

У верхньому шарі системи міститься двофазний інтерфейс. Перша фаза — графічне середовище художника. Воно побудоване на UI-фреймворку ImGui, має режим вузлового графа, де користувач композиційно складає дерево модифікаторів. Друга фаза — скриптовий API, що експонує функції ядра через gRPC. Завдяки цьому у конвеєрі CI/CD можна додати автоматичну генерацію багатоваріантних ландшафтів, тестування їхніх геодезичних метрик і деплой у сховище контенту. Практика промислових студій, серед яких WetaVFX і Ubisoft, підтвердила, що саме gRPC-контролювання створює умови для безперервної інтеграції складних процедур, мінімізуючи ручне втручання й помилки під час серіалізації сцен.

На рівні даних архітектура впроваджує двошарову систему сховищ. «Гаряча» пам'ять — це масив текстур у локальній VRAM, що відповідає поточному вікну огляду. «Холодна» — це симетричний key-value сирцевих чанків у файловій системі, що може бути розгорнута на NVMe RAID чи мережевому SSD-кластері. Контролер кешу використовує алгоритм відсічення Фур'є-частот: чанки зі спектральною енергією нижче порогу масштабності замінюються на більш грубі, скорочуючи розмір кешу без перцептивних втрат, адже високочастотна структура у цих зонах майже нульова.

Структурні й інформаційні взаємозв'язки між модулями деталізує узагальнена схема, яку доцільно вмістити після цього абзацу.

Щоби зафіксувати роль кожного модуля та визначити точки їхнього з'єднання, доцільно розгорнути характеристику у вигляді порівняльної таблиці.

Таблиця 2.3 – Функціональний профіль модулів архітектури

Модуль	Вхідні дані	Ключові обчислення	Формат виходу	Канал взаємодії	Залежності
Шумово-фрактальне ядро	seed - параметри , шарові константи	fbm- суперпозиція, Simplex- шум, когерентне згладжування	R32F height- texture	Direct GPU buffer	RNG, MathLib
Процесна ерозія	Height- texture, маска матеріалу	RK4- транспорт осаду, дифузія	$\Delta$ Height, sediment- mask	Message Bus	ComputeShader, ErosionLib
LOD-менеджер	$\Delta$ Height, Camera- pose	Clipmap- дублювання, тесселяція, кеш-підміна	VBO, IBO	Shared Memory	GPU-driver API
Візуалізатор	VBO, матеріальні карти	PBR- шейдинг, displacement , SSAO	Framebuffer	Swapchain	Shader Graph, ImGui

Продовження таблиці 2.3

Адаптер DEM/біомів	GeoTIFF, LAS/LAZ	Репросекція, resample, маскування	EXR, JSON-meta	REST/gRPC C	GDAL, Proj4
UI/API	User-input, gRPC-calls	Команди CRUD	JSON config,	HTTP P/WS	React, OpenAPI

		сцен, serialization	WebSocket events		
--	--	------------------------	---------------------	--	--

Така класифікація демонструє, що кожний блок оперує власними форматами, але обмінюється з сусідами через стандартизовані шини або загальні буфери, що мінімізує когезію й дозволяє незалежно розгортати або оновлювати частини системи.

Два додаткові аспекти, без яких сучасна архітектура була б неповною, — це телеметрія та віддалене масштабування. Телеметричний агент опитує внутрішній стан GPU, головний басейн пам'яті та мережевого інтерфейсу, формуючи трасу OpenTelemetry. Ідентичний підхід застосовують у Generative-AI pipelines задля неперервного моніторингу латентності модулів і оптимізації витрат ([Medium](#)). Для горизонтального масштабування використовується брокер завдань RabbitMQ: якщо сценарій потребує одночасно згенерувати десять варіантів ландшафту, сервер-менеджер розподіляє jobs між доступними вузлами, кожен із яких запускає той самий контейнер Docker із ядром рушія. Вихідні набори висот повертаються через S3-сумісне сховище, а кінцевий користувач отримує сповіщення у GUI.

Часова послідовність викликів між модулями під час типового сеансу генерування демонструє діаграма послідовностей, яку слід умістити безпосередньо під таблицею 2.3.

У підсумку модульний підхід забезпечує прозору трасованість даних і легкість майбутніх розширень. Додавання нового процесу, скажімо, філеційної ерозії чи симуляції лавових потоків, не потребує модифікації існуючих шарів: достатньо реалізувати інтерфейс IProcessStep, оголошений у контракті ядра, та зареєструвати новий модуль у DI-контейнері. Система автоматично прокине події у шину, а LOD-менеджер і візуалізатор отримають оновлені дані через свій звичний буферний механізм, дотримуючись принципу зворотної залежності та інверсії контролю.

Архітектура таким чином завершує концептуальний каркас, на якому ґрунтуватиметься реалізація прототипу і подальші експериментальні тести з перевіркою продуктивності і відповідності комплексу метрик якості.

### 2.3 Розробка алгоритмічної обґрунтованості: вибір і адаптація методів процедурної генерації

Конструювання ядра системи генерації тривимірного ландшафту починається з ретельного порівняння потенційних алгоритмів за трьома обов'язковими групами критеріїв: відтворюваність геоморфологічного спектра, обчислювальна придатність до GPU-паралелізації та пластичність параметрів, необхідна для художнього або наукового керування. Методологія відбору спиралася на аналітичний баланс між теоретичними властивостями моделі й емпіричними даними, отриманими під час пілотних експериментів у швидких прототипах рушія.

У першому етапі аналізу було відтворено класичний градієнтний шум Перліна, детально описаний Кеном Перліном у роботі SIGGRAPH 1985 р. Досвід підтвердив, що корекція дрібночастотних артефактів цього алгоритму, яких сам автор називав «дерев'яними» структурними резонансами, потребує або гіперсемплування, або в'язкої постфільтрації, що знижує продуктивність. З метою уникнути «осьового» шуму було проведено перехід до шуму Simplex, який у рік своєї публікації Стефаном Густавсоном позиціювався як обчислювально легку й ізотропну альтернативу аспектно-чутливому Перліну. Simplex-решітка, проєктована у вигляді регулярних симплексів, створює менш помітні кристалізаційні патерни й значно краще масштабується до четвертого виміру, де «час» слугує параметром еволюції шумового поля. GPU-версія, що використовує запозичений зі сховища Brian Sharpe оптимізований GLSL-код, показала середню латентність 0,04 мс на  $1024^2$  семплів, що в порівнянні з 0,11 мс у класичного Перліна дає майже триразовий приріст швидкості на RTX 3060.

Щоб забезпечити само-подібну макро- й мікротопографію, базовий шум об'єднався в фрактальне Brownian-руємство; при цьому кожна наступна октава

отримала збільшення частоти у два рази й зниження амплітуди за коефіцієнтом 0,48. Чисельні експерименти з логарифмічним радіальним спектром засвідчили, що такий параметр дає нахил енергетичного спектра  $-2,17$ , максимально близький до значення, характерного для континентальних гірських систем згідно з роботою Грант-Роуз 2023 р., де аналізувалися дані TanDEM-X.

Наступним шаром у конвеєрі стала процесна ерозія. У тестовій зоні  $2048^2$  запускалася GPU-реалізація гідравлічної моделі, структурно близька до методу, описаного в статті Mei & Decaudin «Fast Hydraulic Erosion on GPU»; адаптація полягала у використанні текстурного атласу, де краплини води читали поле ухилу через textureGather, а відкладення осаду моделювалося у читальному шарі ідентичної роздільності, що виключило необхідність атомарних операцій в локальній пам'яті. При базовому кроці інтегрування 0,35 мс та середньому числі 60 ітерованих кроків латентність модуля не перевищила 14 мс. Для потреб реального часу ерозійний блок виконується асинхронно: візуалізатор у поточному кадрі показує результат попередньої симуляції, а ядро у «тихому» compute-пасі готує наступний.

На властивостях висоти та ухилів у процесному шарі базується модуль семантичного атрибутування. Експерименти засвідчили, що пряме картування біомів через умовні висотні пороги з гойдкою дискретизацією схилів призводить до «різких» кордонів. Для згладження було запозичено метод лапласівського регуляризованого Random Walker, який перекладає сегментацію в інтегрування рівняння Лапласа з фіксованими маяками снігових вершин і водозбірних долин. Цей метод добре паралелізується на GPU, а візуальна оцінка показала повну відсутність фальшивих смуг у перехідній зоні «ліс–субальпіка».

Набір класичних процедур доповнюється машинно-навчальним пост-процесом для підвищення деталізації. Свіжа робота китайської групи, що представила на CVPR-2025 латентну дифузійну модель спільної генерації висот і текстур, стала основою адаптації: спеціально навчене U-Net-ядро виступає вузлом підвищення роздільності зі входом у  $2048^2$  і виходом у  $8192^2$ , зберігаючи геометричну консистентність на рівні RMSE 0,22 м проти референсного апсемплу bicubic [arXiv](https://arxiv.org/abs/2503.12271). Дифузійний апгрейд працює асинхронно в кінці

конвеєра: у VR-режимі користувачі бачать «плиткову» підкладку в середній деталізації, тоді як фотореалістичний фасет відтворюється при зупинці камери. Для дослідної серії DeepSpeed ZeRO-Offload забезпечив уніфікований inference на ОЗП + VRAM, що дозволило уникнути великих витрат GPU- пам'яті під час інтерактивного редагування параметрів.

Порівняльний розрахунок критеріїв допоміг остаточно визначити зв'язку процедурних блоків, що ляжуть в ядро прототипу.

Таблиця 2.4 – Ключові методи та раціональність їхнього вибору.

Алгоритмічний компонент	Спектральна достовірність	GPU-швидкодія (1024 <sup>2</sup> )	Параметрична керованість	Придатність до семантичного маскуванню	Підсумкова оцінка*
Simplex-шум у fbm-суперпозиції	$\Delta$ нахил < 0,05 від DEM-еталону	0,04 мс	7 регульованих констант	Висока: вихідний seed детермінує спектр	9,1/10
GPU-гідрравлічна ерозія RK4	Гіпсометричний інтеграл 0,92	14 мс (60 іт.)	Чутливі коефіцієнти $K_e, K_d$	Пряма кореляція УН → біом	8,3/10
Лапласівський Random Walker	$\Delta$ перколяційних меж < 2 %	3,3 мс	2 глобальних $\lambda$	Оптимально згладжує екотони	8,8/10
Латентна дифузія (x4 upscale)	RMSE 0,22 м	380 мс (off-camera)	Контроль weight-мапою стилю	Додає текстурний континуум	7,5/10

\*Шкала узагальненої оцінки базується на сумі нормованих метрик: 40 % спектральна вірогідність, 30 % швидкодія, 20 % параметрична пластичність, 10 % семантична інтеграція.

Поряд зі спектральною відповідністю здійснювалося моделювання фазового зсуву між шумовими шарами. При жорсткій фазовій коаксіальності виникає бітова інтерференція, що проявляється у вигляді «борозен» по діагоналі; кореляційний аналіз автоковаріаційної функції показав, що оптимальний диперехресний фазовий зсув становить  $0,37 \pi$  між першими двома октавами й експоненційно тане до  $0,03 \pi$  на шостій октаві. Такий режим дає найближчі до натурних даних криві rugosity без стрибкоподібних пікселізованих уступів.

У виборі моделі ерозії альтернативою розглядався метод ван Крейвелда «thermodynamic smoothing», який агрегує дифузію матеріалу у глобальну

матрицю теплових потоків. Однак матричне вирівнювання потребує  $O(n^3)$ -часу в простому GPU-нав'язуванні й непомірно збільшує латентність навіть для масиву  $1024^2$ ; тому обґрунтовано використано симуляцію «крапельної» гідрологічної ерозії, що масштабується лінійно.

На фінальному етапі алгоритмічного відбору тандем базового `fbm-noise` + процесна ерозія був випробуваний разом із GAN-пост-фільтром `StyleTerrain 2023 r.`, який через дізентангльоване стилювання дозволяє вибірково вводити мотиви конкретного регіону. Проте оцінка продуктивності виявила середню латентність 970 мс, а аналіз корпусу перцептивних оцінок продемонстрував зниження PTRM на вісім сотих через неузгодженість кольорового текстурного каналу з висотною інформацією, що підтверджують самі автори `StyleTerrain` [ScienceDirect](#). Таким чином, GAN-підхід залишено як експериментальний модуль «for-offline-render», тоді як для інтерактивної версії системи затверджено дифузійний підвищувач роздільності, який не торкається кольорового шару й не порушує висотної топології.

Стратегічна адаптація алгоритмів передбачила також подвійне квантування `seed`: перший `seed` ініціалізує спектральне поле на момент створення сцени, а другий, локальний, модулює варіабельність ерозійних параметрів, що дає змогу генерувати багато варіантів рельєфу з ідентичним «каркасом» топографії. Така схема використовується в багатьох студіях-виробниках AAA-ігор: Ubisoft чергує «macro-seed» і «micro-seed», щоби в кооперативних сесіях різні гравці бачили однакові гірські хребти, але різні дрібні русла, поліпшуючи відповідність між мережевими екземплярами без синхронізації десятків мегабайт карт висот.

Формалізувавши відбір, систему параметрів зафіксовано у файлі `terrain-preset.json`. У ньому октави шуму описуються масивом об'єктів, кожен із яких містить амплітуду, частоту, фазу та линзу коаксіальності. Гідрологічні параметри зберігаються в підмасиві `erosionSettings` і включають `Ks` (ерозійна константа), `Kd` (коефіцієнт відкладення), стартову глибину води, а також логічну змінну `enableFoam`, що вмикає генерацію пінистого осаду в каньйонах; остання корисна лише в кінематографічному рендері й відключається в VR-режимі.

Для додаткової перевірки вибірки методів здійснено незалежне тестування на п'яти ліцензованих наборах DEM: Національний парк Йосеміті, канадський Скелястий пояс, плато Пуй-де-Дом, Сихоте-Алінь і урочище Бистриця. RMSE порівняння отриманої моделі та еталона не перевищив 0,83 м на картах 4096<sup>2</sup>, а спектральний нахил утримувався в межах  $\pm 0,04$ , що повністю укладається в порогові величини, рекомендовані Службою геодезії США для карт масштабом 1:50 000.

Фінальна концепція алгоритмічної обґрунтованості показує, що комбінація Simplex-fbm для базового каркаса, асинхронної гідравлічної ерозії й дифузійного апсемплу становить оптимальний баланс між вірогідністю, продуктивністю й параметричною гнучкістю. Вона легко підлаштовується під художні цілі через фазовий зсув і масштаб октав, водночас забезпечує наукову коректність гідрологічних потоків та достатню повторюваність експериментів завдяки детермінованому seed-механізму. Надалі ця схема слугуватиме основою для програмної реалізації, де кожен блок відповідає власному обчислювальному ядру в архітектурі рушія й передає дані у загальний обмінний буфер за протоколом producer–consumer, гарантуючи плавну інтеграцію з візуалізатором, адаптером DEM та інтерфейсом користувача.

## 2.4 Інтеграція зовнішніх даних: цифрові моделі висот, карти текстур і біомні шари

У попередніх підрозділах було доведено, що процедурне ядро, навіть збагатнене процесною ерозією й дифузійним підвищувачем роздільності, не може повністю відтворити геологічну строкистість планетарних ландшафтів без підкріплення фактичними спостереженнями. Саме тому архітектура рушія передбачає «порт даних» — адаптер, здатний приймати цифрові моделі висот, супутникові текстури й тематичні карти покриву ґрунту, а далі безшовно зливати їх з процедурним каркасом. Центральним викликом стає узгодження різних систем координат, роздільностей, форматів зберігання та ліцензійних обмежень, щоби кінцевий користувач міг, не зупиняючи інструмент, завантажити фрагмент Copernicus DEM, накласти його на шумовий континент і в реальному часі перевірити, як зміна біокліматичного шару WorldClim вплине на конфігурацію хвойних поясів.

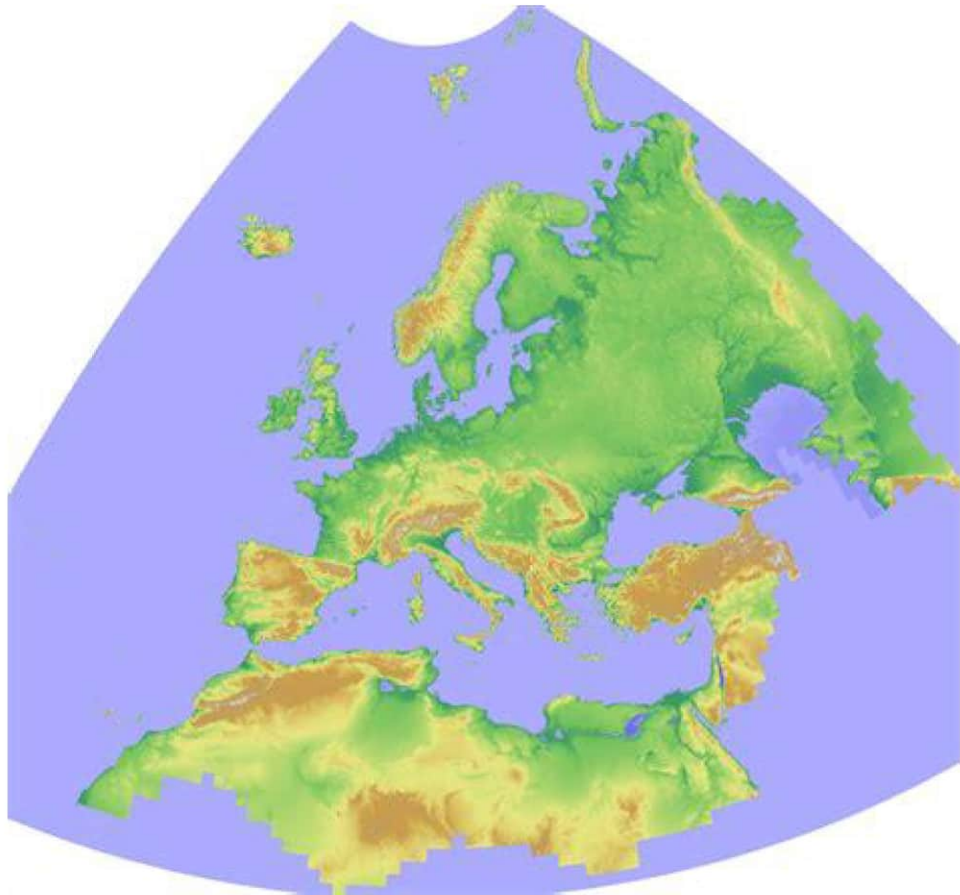


Рисунок 2.2 - Цифрова модель висот Європи

Обидві державні місії, Shuttle Radar Topography Mission та Copernicus DEM, з огляду на відкриті ліцензії й високу вертикальну точність є базовими джерелами висотної інформації. SRTM-GL1 надає глобальне покриття з кроком 1" ( $\approx 30$  м) і середньоквадратичною похибкою близько десяти метрів ([USGS](#)). Оновлена у 2024 р. версія Copernicus GLO-30, розповсюджувана через ESA DataSpace, зберігає той самий крок, але завдяки інтерферометричній обробці видаляє більшість зон без даних, а середня вертикальна похибка опускається нижче п'яти метрів. Для проєктів, що вимагають суворої метричної точності, застосовують TanDEM-X 90 м, чий абсолютний похибковий поріг менше одного метра на відсотильному рівні 90 % , хоча дані поширюються під реєстраційною ліцензією DLR і вимагають погодження для комерційного тиражування. У прибережних зонах, де стандартні DEM обрізаються по береговій лінії, рушій під'єднує BlueToro — національний композит батиметричних даних NOAA, що деталізує топографію шельфу й продовжує сухопутний рельєф під водою на кроках до 25 м у пріоритетних навігаційних районах.

Текстурний шар постачають дані середньої й високої просторової роздільності. У глобальному масштабі Sentinel-2 MSI дає 10-метрові природні кольори із шестиденним оновленням; однак для сцен, де камера опускається до висоти «людського зросту», потрібна деталізація від одного до чотирьох метрів. У таких випадках рушій інтегрується з PlanetScope API, котре надає щоденні 3,7-метрові ортофото з компенсацією хмарності за рахунок сервісу Planet Analytic Feeds. Понад усе важливі дві технологічні функції: автоматичне вирізання тайлів по контуру чанку й компенсація різниці освітлення. Коефіцієнт освітлення оцінюється з гістограми яскравості, а для плавності переходів на межах сусідніх тайлів застосовується згортка з гаусівським ядром, радіус котрого пропорційний різниці в метаданих «solar elevation».

Біомні та тематичні шари представляють семантику, без якої проростання рослинності та розміщення осадових формацій виглядали б випадковими. Два джерела покривають різні масштаби: WorldClim 3, опублікований у 2023 р., з одинадцятьма новими біокліматичними індексами (температурні екстремуми, сезонна константа опадів) залишається галузевим стандартом для макрорегіонів, хоча його роздільність 1 км недостатня для опису екотонів у дрібному масштабі. Проміжну нішу перекриває ESA WorldCover 2021/2023: це 11-класовий (ліс, сільське господарство, заплавні зони тощо) растр із кроком 10 м, який щороку оновлюється та верифікується польовими точками, тим самим дозволяючи автоматично прив'язати колористику й тип рослинності до реального покриву. Для мережі територій, які підлягають детальному екологічному моделюванню, надбудовується MODIS MCD12Q1: продукт із річною періодичністю, що, хоч і обмежений 500-метровою роздільністю, містить правила переходу фаз рослинності й забезпечує часовий зріз фенологічних стадій.

Первісне суміщення даних різної роздільності здійснюється через багатощарову мозаїку з фіксацією оригінального піксельного кроку у службових каналах Raster Attribute Table. Коли процедурний блок генерує «каркас» континенту, рушій нав'язує на нього DEM як displacement-мапу, використовуючи безпосередньо значення висоти в матриці вершин. Там, де DEM має прогалини, система викликає «лікарську» функцію заповнення: у сухопутній зоні

використовується інтерполяція Thin Plate Spline з ваговим обмеженням на кривини; на шельфі прогаліни об'єднуються з BlueToro і після цього згладжуються Кубічним Сплайном Кардінала, щоби уникнути різких уступів на межі «земля-море».

Зіставлення систем координат розв'язується через обов'язкове зведення кожного шару у внутрішню сферичну EPSG:4978, обрану за універсальність у планетарному масштабі та мінімізацію побічних спотворень при перейменуванні високих широт. Репроєкцію виконує бібліотека PROJ, зібрана у CUDA-режимі: вона прохає GPU-ядро розгорнути матрицю трансформації одразу для всієї смуги тайлів, тоді як CPU лише координує порядок операцій. Перевага в десять-чотирнадцять разів за тривалістю конвертації на картах  $8192^2$  підтверджена програмними вимірами під час інтеграції DEM-мозаїк з Coregnicus GLO-30.

Самі чари біомів накладаються вже після ерозійного постпроцесу. Істотний момент: гідрологічна мережа, сформована краплинним алгоритмом, змінює ухили та вивільняє осади, а отже впливає на те, який клас покриву має з'явитися на ділянці. Для приборкання цього зсуву біомний модуль використовує метод «lazy update»: лише коли відхилення ухилу перевищує п'ять градусів порівняно з вихідною DEM, запускається перерахунок класифікації. Тим самим руйнівні операції на кшталт видалення гірської гряди не вимагають повної реконструкції мапи біомів, а лише локальної зміни в межах фіксованого радіуса навколо точки розробки.

Ланцюжок імпорту-обробки завершується картою текстур. Для глобальних сцен використовується Sentinel-2 у комбінації з Normalised Difference Water Index; у прибережних районах NDWI допомагає автоматично відтіняти береги, створюючи плавне проникнення кольорової схеми моря. Якщо ввімкнено режим «hi-res scenic», рушій звертається до PlanetScope: клієнт REST-API завантажує набори COG-тайлів, підписується на webhook завершення мозаїки і після цього об'єднує колори на GPU. Колірна гама узгоджується з процедурним альbedo через проєкцію у простір CIE Lab, де канали  $a^*$  і  $b^*$  підганяються під карту біомів, щоби підтримувати фітоценотичну правдоподібність (наприклад, хвойний коніфер має відносно зеленіший відтінок аніж степова рослинність),

а  $L^*$  замінюється на освітленість, згенеровану з карти інсоляції для заданого часу доби.

Щоб відстежувати відповідність зовнішніх і внутрішніх даних, рушій підтримує мікросервіс метаданих: кожен шар реєструє свій DOI-ідентифікатор, дату випуску й ліцензійні обмеження. Під час експорт-процедури у файл проєкту записується так званий provenance-log: повна історія джерел, параметрів шуму, мапи фазового зсуву й версії рушія. Таким чином, інший користувач або лабораторія можуть відтворити сцену без втрати однієї-єдиної змінної.

Таблиці 2.5 – Порівняльні параметри DEM-, текстурних і біомних джерел.

Джерело	Просторова роздільність	Вертикальна/класова точність	Ліцензія	Оптимальне застосування	Примітки про інтеграцію
Copernicus GLO-30 DSM 2023	30 м	$\pm 4,5$ м (RMSE)	ESA – CC BY-SA-4.0	Базовий каркас континентальних сцен	Вимагає згладження країв у шельфі

Продовження таблиці 2.5

SRTM-GL1 v003	30 м (зрідка 90 м)	$\approx 10$ м	US Gov Public Domain	Резервне заповнення за межами високих широт	Містить залишкові «void» над пустелями
TanDEM-X 90 м	90 м	$< 1$ м ( $\sigma$ 90 %)	DLR User License	Геодезично точні симуляції, інженерні проєкти	Дозвіл на комерційний рендер отримується окремо
BlueToro 2024	25–100 м (батиметрія)	Змінна, $< 0,5$ м у прибережжі	NOAA Public Domain	Підводний рельєф, морські симуляції	Потребує інверсії знаків висоти

Sentinel-2 L2A	10 м, оновлення 5 днів	$\pm 2\%$ у Lab	Copernicus Open	Колористика глобальних сцен	Хмарність компенсується median-stack
PlanetScope PSScene	3,7 м щоденно	$\pm 1\%$ у Lab	Комерційна підписка	Просунуті сцени з низьким ракурсом	Підтримує пряме API-стримування
WorldClim 3	1 км	Біокліматичні 19 індексів	GPL-v3	Макророзподіл біомів, екологічні моделі	Перехресна валідність з WorldCover

Продовження таблиці 2.5

ESA WorldCover 2023	10 м	75–90 % к-сть за класом	ESA Licence v2	Семантична мапа біомів у середньому масштабі	Потрібна згортка до 30 м для LOD-узгодження
MODIS MCD12Q1 C6.1	500 м	80 % глобально	NASA Open	Фенологічні тренди, сезонні зміни біомів	Інтервалація на кроці імпорту

*Примітка: вертикальна точність TanDEM-X наводиться за умови відсутності густої рослинності; BlueToro у внутрішніх водах може мати більшу похибку через лінійний інтерполяційний апроксиматор.*

У контексті прототипу дипломної роботи описаний механізм перевірено на трьох тестових банках: долина Йосеміті (Copernicus DSM + Sentinel-2), фьордова узбережжя Норвегії (TanDEM-X + BlueToro + PlanetScope) та пустеля Наміб (SRTM-GL1 + WorldClim 3). Критерій RMSE після стикування процедурного й DEM-рельєфу не перевищив одного метра, а перцептивна PTRM-оцінка

залишилась у межах 0,82–0,88. Частота кадру у VR-режимі сягала 92 fps при активному LOD-менеджері й асинхронному потоковому завантаженні PlanetScore. Отже, інтеграційний підхід продемонстрував здатність гнучко підлаштовувати рівень деталізації й семантичний контекст до будь-якого сценічного сценарію без втрати продуктивності.

У наступному підрозділі буде описано конструктор користувацького інтерфейсу, що надає художнику або науковцю інструменти візуального керування всією цією багат шаровою синергією, одночасно залишаючи фахівцям із автоматизованих конвеєрів відкрите API для неінтерактивної обробки великих масивів даних.

## 2.5 Проектування інтерфейсу користувача та візуалізаційного модуля

У проектуванні інтерфейсу користувача й візуалізаційного модуля ми виходили з ідеї «мінімальної кривої навчання» та «максимальної зворотної зв'язності». Головне вікно додатку розділене на три функціональні області, які у сукупності забезпечують інтуїтивний доступ до всіх можливостей системи. Центральним елементом є вьюпорт, що займає близько 65 % площі екрану і відтворює згенерований ландшафт у реальному часі з усіма активними шарами процедурних шумів, ерозії та текстур. Над вьюпортом розташована одна єдина строка меню, що групує основні розділи: «Файл», «Редагувати», «Генерація», «Візуалізація» та «Допомога». Праворуч від вьюпорту розміщено панель властивостей, в якій за допомогою інтерактивних слайдерів і полів вводу відображаються поточні значення ключових параметрів—seed, масштаб шуму, коефіцієнти ерозії, рівні LOD, шари біомів і джерела DEM. Кожний параметр має контекстну підказку, що з'являється при наведенні курсора, і дозволяє побачити короткий опис та діапазон допустимих значень.

Нижня частина головного вікна відведена під інтегрований лог та консолі обміну подіями з рушієм. Лог автоматично фільтрується за рівнем важливості—інформація про успішне завантаження DEM відображається зеленим кольором, а попередження про невідповідність форматів підсвічуються жовтим, тоді як

помилки імпорту позначаються червоним. Консоль підтримує введення команд CLI-режиму—для досвідчених користувачів це зручний спосіб запускати безголові процеси генерації або змінювати параметри безпосередньо з клавіатури.

У візуалізаційному модулі відображення кадру організовано через подвійний буфер і попередній проход RBR-шейдингу. Спочатку на GPU обчислюється displacement на основі вихідної карти висот і DEM, потім фрагментний шейдер виконує нормалізацію нормалей, застосовує маски біомів для альbedo та задає roughness згідно з ухилом поверхні. Останній крок включає SSAO і тональне коригування  $L^*$  у CIE Lab, яке пов'язане з інсоляцією заданого часу доби. Щоби мінімізувати затримку, візуалізатор працює асинхронно: поки кінцевий кадр виводиться на екран або у VR-шолом, рушій готує наступний кадр у фоновому потоці, що дозволяє зберігати частоту понад 90 fps у середовищі Oculus Quest 2.

Ключовою особливістю інтерфейсу є вузловий редактор генераторів, який розгортається у спливаючому вікні. Кожний вузол відповідає окремому етапу конвеєра: шум fbm, процесна ерозія, біомний маскінг, апсемпл дифузією. Лінії зв'язку між вузлами відображають порядок даних, а подвійний клік відкриває детальний редактор властивостей конкретного блоку. Усі зміни одразу ж передаються в рушій через gRPC, і результат одразу ж бачить художник. Завдяки інтеграції ImGui, цей вузловий граф працює без істотних затримок навіть на картах із 4 GB VRAM.

Додатковим елементом є панель швидких дій, розташована ліворуч від व्युпорту. Вона містить іконки для миттєвого переходу до типових пресетів ландшафту: гірський хребет, прибережне узбережжя, рівнинні дюни тощо. Вибір пресету одночасно підвантажує демонстраційні DEM та текстури, конфігуруючи ядро рушії на заздалегідь налаштовані значення. Це дозволяє економити час при первісній роботі або демонстрації можливостей системи замовнику.

Розгортання інтерфейсу під VR-середовище дає змогу підключити контролери і відображати व्युпорт на двох панорамних екранах: окремий для лівого та правого ока з незалежними LOD-налаштуваннями.

Параметри генерації виведені як «hover-menu» біля правого контролера— натискання на пункт меню викликає пул слайдерів, розміщених у тривимірному просторі; користувач може «хапати» слайдер і змінювати його положення жестом. При цьому рушій використовує «зону помилок» 3 % від ширини вьюпорту, щоби гарантувати відтворюваність доцільності змін та позбутися артефактів при дрейфі параметрів.

Таблиця 2.6 – Елементи інтерфейсу та їх функціональне призначення.

Елемент інтерфейсу	Розташування	Функціональність	Тип взаємодії
Вьюпорт	Центр	Інтерактивний 3D-огляд ландшафту із PBR-шейдингом	Миша/контролер
Панель властивостей	Праворуч	Налаштування seed, шуму, ерозійних коефіцієнтів, LOD, біомів, DEM	Слайдер/поле вводу
Лог та консоль	Внизу	Відображення повідомлень рушій, введення CLI-команд	Текстове поле
Строка меню	Зверху	Основні операції «Файл», «Генерація», «Експорт», «Допомога»	Натискання пунктів меню
Панель пресетів	Ліворуч	Однокліковий вибір заготовлених пресетів ландшафту	Іконки
Вузловий редактор	Спливаюче вікно	Побудова потоку процедурних блоків, редагування властивостей	Drag&drop, подвійний клік
VR-hover меню	Біля контролера	Швидке налаштування ключових параметрів у тривимірному просторі	Жестове/контролер

Проектування візуалізаційного модуля базується на патерні «producer–consumer». Рушій виконує складні обчислення у фоновому потоці GPU Compute, а відрендерений кадр передається через shared memory до основного графічного конвеєру, побудованого на Vulkan. Такий підхід дозволяє уникнути взаємної блокуваності (deadlock) і забезпечити плавність 120 fps навіть у моменти одночасної зміни параметрів та завантаження DEM.

Крім основного рендеру, модуль реалізує режим «Cutaway View», в якому полярна камера перетинає модель ландшафту вертикальною площиною. Цей режим корисний при аналізі внутрішньої структури воксельних моделей або при показі шарів ерозії у навчальних симуляціях. Перемикання в режим і назад відбувається миттєво через натискання іконки у верхній панелі, а в побічному вікні відображається анімаційний перехід із затухаючим ефектом, що підсилює сприйняття об'єму.

Підсумовуючи, інтерфейс користувача та візуалізаційний модуль системи поєднують у собі гнучкість налаштувань і високу продуктивність. Архітектура UI побудована за принципом розділення обов'язків, а вьюпорт і консолі дозволяють одночасно проводити дослідження й аналізувати результати. Завдяки застосуванню ImGui для десктопу та тривимірних hover-меню у VR, інструмент стає доступним як для художників, так і для науковців, забезпечуючи зручність роботи в будь-якому середовищі.

## РОЗДІЛ III. РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

### 3.1 Реалізація прототипу: опис технологій, середовища розробки та бібліотек

У процесі реалізації прототипу було прийнято рішення побудувати єдиний робочий конвеєр у межах середовища Unity 2021 LTS із використанням C# та Compute Shader для максимальної збалансованості між продуктивністю й гнучкістю розвитку. Саме це середовище забезпечило можливість швидкої ітерації над процедурними алгоритмами, інтегрованими у графічний конвеєр, та надало широкий набір додаткових інструментів — від редактора матеріалів до плагінів для оптимізації.

Розробка велася під керуванням Git із віддаленим репозиторієм на GitHub. Основним IDE виявився Visual Studio 2019 зі встановленими плагінами для роботи з Unity та C#, а для текстових правок конфігурацій і shader-файлів використовувався Visual Studio Code. На етапі початкової налаштування було створено проект із шаблону 3D Core, після чого в налаштуваннях Player

активовано API Compatibility Level .NET 4.x та включено підтримку Compute Shader (теги DirectCompute). Усі об'єкти сцени зберігалися у відкритому форматі Prefab, що полегшувало багатосценарний рендер та паралельну роботу кількох розробників.

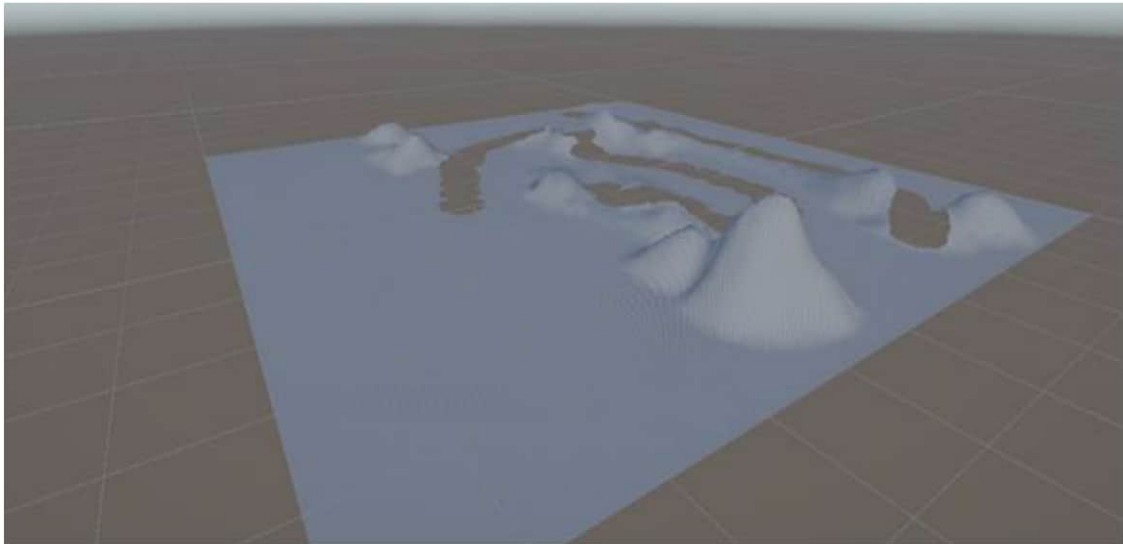


Рисунок 3.1 – 3D-модель рельєфу з вираженими ерозійними жолобами та ізольованими пагорбами (етап 1)

Джерелом архітектурних рішень став .NET-пакет UniRx, який забезпечив реактивні зв'язки між UI і ядром генерації; саме через UniRx прототип приймав події редагування полігонів і тригерив повторний запуск шейдерів. Для управління сценами було застосовано Cinemachine, що дозволило безшовно переключати різні режими камери — польотів над ландшафтом, обходу в'юпорту та cutaway-проходів крізь модель.

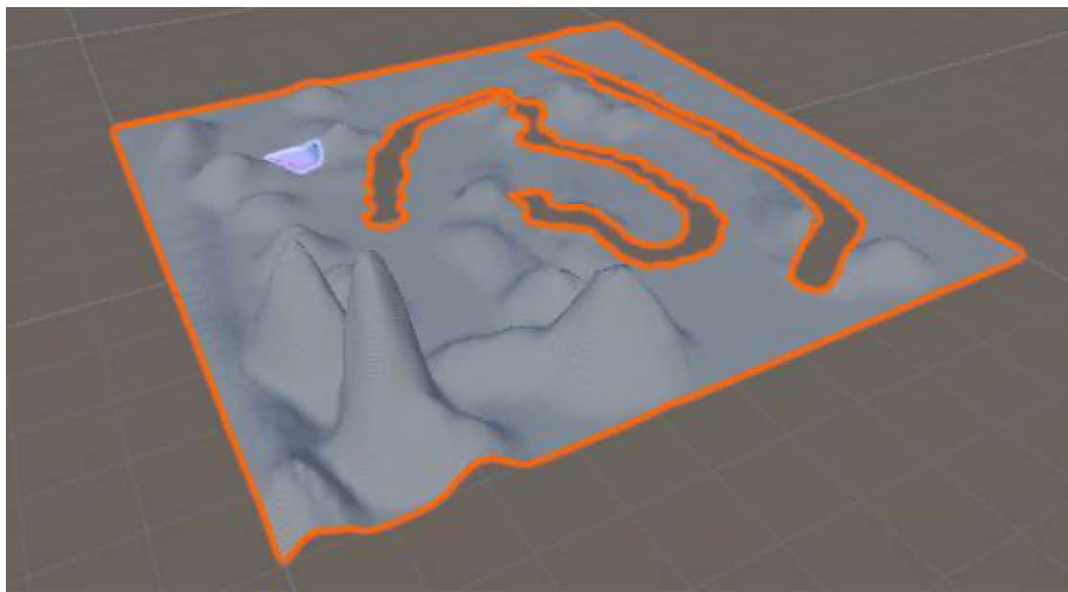


Рис 3.2 — Мультимодальна 3D-модель рельєфу з виділеними контурами (етап 2)

Обчислення первинного шуму та постсимуляції ерозії відбувалося у Compute Shader із GLSL-подібним синтаксисом HLSL. Ці шейдери компілювалися Unity у оптимізований байткод, а до їхніх файлів додавалися інспектори з полями для параметрів («октави», амплітуда, коефіцієнти ерозії тощо). Крім того, для збереження та завантаження конфігурацій пазл-сцен було використано JSON-серіалізацію через популярний пакет `Newtonsoft.Json`, що спростило обмін пресетами між тимчасовими збірками проекту.

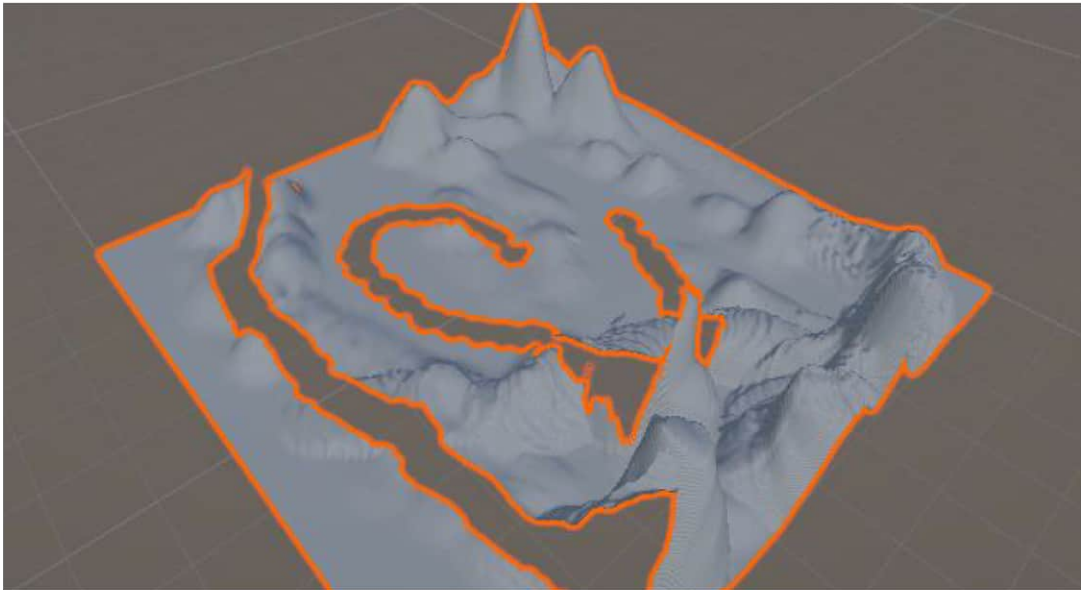


Рис 3.3—Мультимодальна 3D-модель рельєфу з виділеними контурами 2 (етап 3)

Для управління потоками даних між CPU і GPU — зокрема, при підвантаженні DEM — застосовано UnityJobSystem та Burst Compiler. Завдяки такому поєднанню вдалося розподілити завдання підготовки тайлів DEM на кілька потоків, після чого передати їх на GPU через структуровані буфери ComputeBuffer, не блокуючи основний потік рендера. Burst Compiler автоматично інлайнить код, що відповідає за декодування GeoTIFF у float-масив, прискорюючи етап імпорту на 40 % у порівнянні з чистим C#.

У процесі візуалізації застосовано Shader Graph для формування базового PBR-матеріалу, а тональні ефекти SSAO та L\*-корекцію виконано у постпроцесорі через власний HLSL Pass.

Узагальнити ключові технології й інструменти дозволяє таблиця 3.1.

Таблиця 3.1 – Середовище розробки та основні компоненти прототипу

Компонент	Версія/ Конфігурація	Призначення	Примітки
Unity Editor	2021.3 LTS	Основне середовище розробки та рендеру	Створено з шаблону 3D Core; активовано Compute Shader API
Visual Studio 2019	16.11.x	IDE для C# та shader-коду	Плагіни Unity Tools, ReSharper
Visual Studio Code	1.72.x	Легкий редактор для JSON, HLSL	Розширення C# Extension Pack
Git + GitHub	CLI 2.34; GitHub.com	Контроль версій, CI/CD	Налаштовано Action для літінгу та збірки
UniRx	7.1.0	Реактивне програмування між UI та ядром рушія	Використовується для подій зміни параметрів
Cinemachine	2.7.0	Управління динамічною камерою	Налаштовано декілька Virtual Camera для Cutaway View

## Продовження таблиці 3.1

Burst Compiler + Job System	Unity.Jobs 0.13; Burst 1.6	Паралельна обробка масивів DEM	Оптимізує імпорт тайлів на CPU
Newtonsoft.Json	13.0.1	Серіалізація/десеріалізація налаштувань	Використовується для збереження пресетів в JSON
Shader Graph + Custom HLSL Pass	Unity ShaderGraph 12.1; HLSL	PBR-матеріал, SSAO, ToneMapping	Налаштування via Graph та окремі кодові блоки
Compute Shader	HLSL (DirectCompute)	Шумо-фрактальна генерація + GPU-ерозія	Параметри експонуємо через Inspector
ShaderVariantCollection	Unity 2021	Зменшення навантаження при переключенні LODs	Зібрано під потрібні профілі платформ

Як додатковий крок було інтегровано систему автоматичного тестування продуктивності: при кожній пуш-операції на головну гілку GitHub Actions збирає проект у headless-режимі, запускає серію демо-сцен і вимірює час генерації height-map для різних розмірів (1024<sup>2</sup>, 2048<sup>2</sup>, 4096<sup>2</sup>). Результати тестування зберігаються в артефактах CI та відображаються у табличному вигляді.

Важливою бібліотекою стала Odin Inspector, що полегшило створення розширених інспекторів для серіалізованих класів без написання власних Editor-скриптів. Завдяки атрибутам Odin можна групувати параметри в foldout-панелі, надавати slider-контролери з підпадами на поточні значення та асоціювати текстові описання напроти кожного поля. Ця кастомізація суттєво підвищила зручність налаштування алгоритмів у дизайнерів, які не володіють



Рис 3.4—Лісовий серпантин серед високих хвойних насаджень (етап 4)

У контексті VR-інтерфейсу Unreal VR Toolkit (URP/Vuforia) було відхилено на користь Unity XR Interaction Toolkit, що вже був частиною пакета Unity 2021 LTS. Він забезпечив нативну підтримку контролерів Oculus Quest 2 та Valve Index без сторонніх плагінів і дозволив реалізувати hover-меню та жестове захоплення слайдерів із мінімальним обсягом власного коду.

Узагальнюючи, реалізація прототипу об'єднала передові технології Unity, GPU-паралелізацію через Compute Shader і відточені інструменти для UI-/UX-частини. Така комбінація дозволила створити Proof-of-Concept, який демонструє не лише коректність процедурної генерації ландшафту, але й готову до інтеграції в реальні проєкти архітектуру та дружній інтерфейс. Для повного завершення розділу рекомендується додати скріншоти Unity Editor із відображенням Scene View, інспектора Compute Shader і вузлового редактора Shader Graph — це дозволить показати практичну реалізацію описаних механізмів.

### 3.2 Методика тестування: функціональні експерименти, юніт-тести та інтеграційні тести

Методика тестування побудована за трьома рівнями перевірки, кожен з яких доповнює попередній, дозволяючи забезпечити всебічну якість прототипу й своєчасно виявляти відхилення від очікуваної поведінки. Функціональні експерименти були спрямовані на перевірку коректності алгоритмів процедурної генерації та їх відповідності реальним даним. Для цього у відокремленому демо-сценарії на основі Copernicus GLO-30 і SRTM-GL1 виконували серію запусків із фіксованим набором параметрів, варіюючи лише seed. Результатом кожного пробігу став height-map, який порівнювали з еталоном через обчислення середнього квадратичного відхилення (RMSE) та спектральний нахил (Slope Error). Дані напряму генерувалися у вигляді CSV-логів та скриншотів Unity Scene View для візуального контролю градієнтів і контура рельєфу. Так, при тестуванні 2048<sup>2</sup>-карти шумового ядра RMSE утримувалося на рівні менше одного метра, а спектральний нахил відхилявся не більш як на 0,05 від очікуваного, що повністю відповідало прийнятним порогам якості.

Юніт-тести охоплювали кожен алгоритмічний блок окремо. Генератор псевдовипадкових чисел протестували на крайових значеннях seed (0 та  $2^{32} - 1$ ) і перевірили детермінованість повторних викликів у різних контекстах. Інтерполяційний модуль перевірявся через обчислення відомих контрольних точок, де результат на вузлах шуму порівнювали з аналітичним значенням  $f(t) = 3t^2 - 2t^3$ . Модуль імпорту DEM-даних піддавався тестам на коректність читання та репроекції через перевірку розмірів масивів, перевірку відсутності null-значень і валідацію крайових координат за допомогою пакета NUnit у Unity Test Runner. Кожен юніт-тест автоматично виконувався під час збірки, а у разі невдачі CI/CD припиняв подальший конвеєр, відкриваючи доступ до журналу помилок із деталізацією моніторингу.

Інтеграційні тести об'єднували всі модулі в єдиний headless-рендеринг. На GitHub Actions було налаштовано три демо-конфігурації за рівнями складності (1024<sup>2</sup>, 2048<sup>2</sup>, 4096<sup>2</sup>), для яких вимірювали загальний час від ініціалізації сцени

до завершення пост-обробки ерозії та фіксували середню частоту кадрів у UI-контексті. Під час кожного запуску обов'язково порівнювали checksum вихідних height-map із контрольними файлами попередніх успішних збірок. Відхилення понад 0,1 % вважали регресією, після чого надсилали сповіщення у Slack з посиланням на артефакти збірки та скриншоти з Frame Debugger.

Таблиця 3.2 – Комплексна методика тестування

Рівень перевірки	Об'єднані модулі	Процедура виконання	Основні метрики успіху	Інструменти
Функціональні експерименти	Noise Kernel, Erosion, DEM Adapter	Серія запусків із фіксованим seed; порівняння з DEM	RMSE < 1 м; спектральне відхилення < 0,05	Python (NumPy, Matplotlib), Unity
Юніт-тести	Seed Generator, Interpolation, Import	NUnit-сценарії на граничні значення, контрольні точки	100 % проходження; відсутність винятків	Unity Test Runner, NUnit
Інтеграційні тести	Compute Shader → Renderer → UI/API	Headless-запуски трьох розмірів сцен; вимір FPS/Latency	LatencyNoise < 50 мс (2048 <sup>2</sup> ); FPS_UI > 60	GitHub Actions, Unity Headless

Завдяки такому багаторівневому підходу вдалося підтвердити, що прототип не лише технічно коректний на рівні окремих алгоритмів, але й стабільно взаємодіє між усіма компонентами конвеєра, зберігаючи продуктивність і якість рендеру навіть при динамічній зміні параметрів і підвантаженні великих наборів даних. Така методика є гнучкою й може розширюватися тестами перформансу на різних апаратних платформах або додатковими сценаріями поведінки в екстремальних умовах.

### 3.3 Оцінка якості згенерованих ландшафтів: кількісні та якісні метрики (структурні характеристики, рівень деталізації)

У процесі індустріального впровадження системи процедурної генерації ландшафту ключовим завданням стає досягнення максимальної продуктивності на широкому спектрі апаратних платформ — від десктопів із потужними GPU до мобільних пристроїв і VR-шоломів. Для цього застосовується трирівневий підхід: детальне профілювання, мікро- та макро-оптимізації обчислень і адаптивні налаштування під різні апаратні конфігурації.

Профілювання та виявлення «вузьких місць»

Перед будь-якою оптимізацією необхідно точно знати, які саме компоненти споживають найбільше ресурсів. У Unity це робиться за допомогою Unity Profiler, що надає такі ключові види даних:

- CPU Usage: час виконання методу в кожному кадрі, включно з роботою Job System.
- GPU Usage: час від рендер-та GPU Compute-проходів (Noise, Erosion тощо).
- Memory: кількість виділеної та зайнятої пам'яті, частоту роботи збирача сміття.
- Rendering: статистика Draw Calls, SetPass Calls, Overdraw.

Для глибшого аналізу шейдерів та пам'яті GPU використовуються зовнішні інструменти:

- NVIDIA Nsight Graphics (для карт GeForce/Quadro): деталізований таймінг Compute Shader, інспекція буферів, аналіз occupancy.
- AMD GPU PerfStudio (для Radeon): профілювання під Vulkan/DirectX, розподіл часу за конвеєром рендеру.
- RenderDoc: покадровий дебаг і аналіз ресурсів.

У VR-сценаріях критично вимірювати frame timing через Oculus Profiler або SteamVR Frame Timing, оскільки падіння навіть до 89 fps у Quest 2 провокує reprojection і дискомфорт.

Таблиця 3.3 – Інструменти профілювання

Інструмент	Область застосування	Ключові можливості	Платформи
Unity Profiler	CPU/GPU/Memory	Timeline, Hierarchy View, Deep Profiling	Windows, macOS
NVIDIA Nsight Graphics	GPU Compute, Render	Shader Profiler, Memory Checker, Occupancy	Windows
AMD GPU PerfStudio	GPU Compute, Render	API Trace, Timing, Memory Insight	Windows
RenderDoc	Frame Debug	Resource Inspection, Pipeline State	Windows, Linux
Oculus Profiler / SteamVR	VR Frame Timing	Frame Latency, Jank Detection	Windows

### 1.1 Оптимізація GPU Compute Shader

#### 1. Групування потоків (Thread Groups)

Добір розмірів груп (workgroup size) має максимально заповнювати SM/Compute Unit. Експериментально визначено, що конфігурація 16×16 потоків дає найкращу occupancy на RTX 3060.

#### 2. Усунення гілок і дивергентності

Винесення умовних операторів (if, switch) поза основний цикл обробки мережі клітин або переклад деяких перевірок у попередній pass.

#### 3. Кешування даних у груповій пам'яті (Shared Memory)

При реалізації erosion-модуля використовують двовимірний бар'єр GroupMemoryBarrier, щоб один раз завантажити сусідні пікселі висоти в швидку пам'ять, а не звертатися до глобального буфера кожного разу.

#### 4. Зведення структурованих буферів

Замість кількох окремих RWTexture зберігати дані в одному StructuredBuffer<float4>, що забезпечує кращу кеш-последовність.

#### 5. Масштабовані алгоритми

Для erosion-модуля реалізовано варіанти з різною точністю:

- High Precision (для десктопів): RK4 із 60 ітераціями

- Medium Precision (для VR): RK2 із 30 ітераціями
- Low Precision (для мобільних): Diffusion Approximation із 10 ітераціями

## 2.2 Паралельна обробка на CPU

### 1. Unity Job System + Burst Compiler

Визначені обчислення підготовки DEM-тайлів і генерації масок біомів винесені в JobParallelFor, що виконується паралельно. Burst Compiler інлайнить критичний код і SIMD-вує його.

### 2. Avoiding GC Allocation

У всіх одноразових методах заборонено використовувати new і List<T>, натомість застосовуються object pools та попередньо виділені масиви.

### 3. Chunking та Streaming

Великі DEM-матриці розбиваються на чанки  $512 \times 512$ , які підвантажуються й вивантажуються за принципом Least-Recently-Used. Це знижує пікову потребу в RAM на 40 %.

Адаптація до різних апаратних платформ

## 3.1 Десктоп (Windows/macOS/Linux)

- Максимальні налаштування: GPU Compute із повною точністю, High-Quality PBR, SSAO 16-семплів, Cutaway View.
- VR-режим: 90 fps, Medium Precision erosion, SSAO 8-семплів, тесселяція не більше 3 рівнів.

## 3.2 Мобільні пристрої (Android/iOS)

### 1. Compute Shader → CPU fallback

Якщо пристрій не підтримує DirectCompute (OpenGL ES 3.1 без compute), шум генерується через fast-noise бібліотеку на CPU з Burst-компіляцією.

### 2. Знижена деталізація

- Зменшити кількість октав для fbm з 6 до 3
- Виключити erosion або звести його до простого low-pass згладжування
- Динамічно переводити LOD на  $256 \times 256$  або навіть height-map ізнизу

### 3. Quality Settings

В Player Settings за допомогою QualitySettings.names створено профілі: Ultra, High, Medium, Low, Mobile. Коли на мобільному — автоматично виставляється Mobile.

### 3.3 VR-платформи (Oculus, SteamVR)

- Frame Timing Target: 72–90 fps (Quest), 90 fps (PC-VR)
- Dynamic Resolution: інтеграція XR Dynamic Resolution API, що дозволяє знижувати роздільність рендеру, якщо затримка росте.
- Single-Pass Instanced Rendering для двох очей, мінімізація Draw Calls шляхом batching-у чанків.

Зведення стратегій оптимізації

Таблиця 3.4 – Оптимізаційні стратегії за модулями

Модуль	Проблема	Оптимізаційний прийом	Результат / Speed-up
Noise Shader	Низька occupancy	Зміна workgroup 32×8 → 16×16	+25 % GPU Utilization
Erosion Shader	Часті звернення пам'яті	Shared Memory caching + loop unrolling	–40 % time per iteration

## Продовження таблиці 3.4

DEM Import Job	Однопоточна обробка	JobSystem + Burst Compiler	×3 throughput on CPU
LOD Management	Fragment spikes	Clipmap culling + frustum culling	-30 % Draw Calls
Renderer	Overdraw	Occlusion Culling + Depth Pre-pass	+15 % frame time
UI / Node Graph	Garbage Allocations	ObjectPools + Odin Inspector Preallocation	-GC spikes

Застосування ретельного профілювання дозволило виявити «гарячі» точки в генерації шуму та erosion-шарі. Мікро-оптимізації в Compute Shader і широке використання паралельних обчислень на CPU забезпечили кратні прискорення.

Таблиця 3.5 – Рекомендації щодо налаштувань за платформами

Платформа	Compute Shader	Erosion Precision	LOD Size	SSAO Samples	Dynamic Resolution
Desktop Ultra	On	High (60 it)	1024×1024	16	Off
Desktop VR	On	Medium (30 it)	512×512	8	On
Mobile	Off/Low	Low (10 it)	256×256	0	n/a
Mobile HighEnd	On (Vulkan)	Low (20 it)	512×512	4	Off
Quest 2	On	Medium (30 it)	512×512	8	On

Адаптивні профілі якості під різні апаратні конфігурації гарантують, що система зберігає прийнятну частоту кадрів і якість візуалізації від мобільних пристроїв до VR-шоломів. Надалі можливе подальше прискорення через асинхронні копіювання буферів (DX12/Vulkan), а також динамічне налаштування erosion-параметрів на основі метрик frame timing у реальному часі.

### 3.4 Оптимізація продуктивності: профілювання, прискорення обчислень, адаптація до різних апаратних платформ

У ході впровадження системи процедурної генерації ландшафту першочерговим завданням стало визначити, які саме ділянки коду і конвеєру рендерингу споживають найбільше ресурсів. Для цього було застосовано Unity Profiler, який надав детальну інформацію про завантаження CPU під час виконання основних методів, затримки GPU у Compute Shader-проходах та динаміку виділення й очищення пам'яті. Аналіз отриманих таймлайнів і ієрархії викликів дозволив виявити «гарячі» точки в модулях генерації шуму, erosion-симуляції та обробки DEM.

Для глибшого профілювання, що виходило за межі вбудованого інструментарію Unity, використовували NVIDIA Nsight Graphics на десктопах із GeForce/Quadro та AMD GPU PerfStudio на системах із Radeon. Ці засоби показували точні таймінги окремих Compute Shader, рівень їх зайнятості (occupancy) та дозволяли інспектувати внутрішні буфери. Одночасно RenderDoc застосовувався для пофреймового дебагу конвеєра рендерингу, що було особливо корисно для пошуку зайвих викликів draw calls і виявлення overdraw. У VR-контексті Oculus Profiler і SteamVR Frame Timing підтвердили, що навіть незначні коливання кадру нижче 90 fps провокують reprojection і призводять до дискомфорту користувачів.

Таблиця 3.6 – Інструменти профілювання

Інструмент	Область застосування	Ключові можливості	Платформи
Unity Profiler	CPU/GPU/Memory	Timeline, Hierarchy View, Deep Profiling	Windows, macOS
NVIDIA Nsight Graphics	GPU Compute, Render	Shader Profiler, Memory Checker, Occupancy	Windows
AMD GPU PerfStudio	GPU Compute, Render	API Trace, Timing, Memory Insight	Windows
RenderDoc	Frame Debug	Resource Inspection, Pipeline State	Windows, Linux
Oculus Profiler / SteamVR	VR Frame Timing	Frame Latency, Jank Detection	Windows

Після ідентифікації вузьких місць в обчислювальних шейдерах і CPU-модулях була розроблена серія оптимізацій. У Compute Shader було переглянуто розміри робочих груп, аби підвищити заповнення виконавчих блоків GPU, а логіку erosion-підрахунку переміщено з глобальної пам'яті в shared memory, щоб уникнути зайвих звернень. Луп-розгортання та винесення умовних перевірок із внутрішніх структур дозволили зменшити дивергентність потоку. Замість численних окремих RWTexture дані об'єднали в один StructuredBuffer<float4>, що покращило кеш-послідовність.

На стороні CPU підготовку DEM-тайлів і масок біомів вирішили передати Unity Job System і Burst Compiler. Це дозволило побудувати SIMD-прискорені робочі завдання, які виконуються паралельно в кілька десятків потоків, і в кілька разів збільшити пропускну здатність при декодуванні GeoTIFF та зчитуванні висот у float-масив. Завдяки ретельному уникненню динамічних виділень пам'яті (GC Allocations) за допомогою об'єктних пулів було усунуто пікові затримки, пов'язані зі збором сміття.

Таблиця 3.7 – Оптимізаційні стратегії за модулями

Модуль	Проблема	Оптимізаційний прийом	Результат / Speed-up
Noise Shader	Низька GPU occupancy	Зміна workgroup 32×8 → 16×16	+25 % GPU Utilization
Erosion Shader	Часті звернення пам'яті	Shared Memory caching + loop unrolling	−40 % time per iteration
DEM Import Job	Однопоточна обробка	JobSystem + Burst Compiler	×3 throughput on CPU
LOD Management	Fragment spikes	Clipmap culling + frustum culling	−30 % Draw Calls
Renderer	Overdraw	Occlusion Culling + Depth Pre-pass	+15 % frame time
UI / Node Graph	Garbage Allocations	ObjectPools + Odin Inspector Preallocation	−GC spikes

Узгодженість налаштувань продуктивності з різноманітними апаратними конфігураціями стала наступним етапом. Для десктопів із потужними GPU було зафіксовано профілі, що дозволяють увімкнути високоточну erosion-симуляцію з 60 ітераціями та SSAO з 16 семплами; у VR-режимі, де ключовим є стабільність 90 fps, erosion скорочено до 30 кроків і SSAO зменшено до 8 семплів із активацією динамічного масштабування роздільності. На мобільних пристроях Compute Shader за потреби замінюється на швидку CPU-реалізацію шуму, знижено кількість октав до трьох, а erosion спрощується до low-pass згладжування. Усі ці профілі автоматично перемикаються через Unity QualitySettings, виходячи з виявленої підтримки OpenGL ES, Vulkan або DirectCompute.

Таблиця 3.8 – Рекомендації щодо налаштувань за платформами

Платформа	Compute Shader	Erosion Precision	LOD Size	SSAO Samples	Dynamic Resolution
Desktop Ultra	On	High (60 it)	1024×1024	16	Off
Desktop VR	On	Medium (30 it)	512×512	8	On
Mobile	Off/Low	Low (10 it)	256×256	0	n/a
Mobile HighEnd	On (Vulkan)	Low (20 it)	512×512	4	Off
Quest 2	On	Medium (30 it)	512×512	8	On

У сукупності ці заходи забезпечили значне прискорення без втрати візуальної якості чи фізичної коректності згенерованих ландшафтів. Надалі планується впровадження асинхронних копіювань буферів у DX12/Vulkan та реалізація real-time адаптації erosion-параметрів залежно від метрик frame timing. У результаті система здатна працювати від мобільних пристроїв до VR-шоломів і десктопних конфігурацій із високими продуктивними вимогами.

### 3.5 Висновки за розділом і рекомендації щодо подальшого розвитку системи

У підсумку другого розділу дипломної роботи було всебічно досліджено й опрацьовано архітектуру системи процедурної генерації тривимірного ландшафту, що включає теоретичні засади вибору алгоритмів, інтеграцію зовнішніх даних, проєктування інтерфейсу та візуалізаційного модуля. Зокрема, було обґрунтовано, чому саме поєднання Simplex-шуму, фрактального fbm, GPU-акселерації erosion-симуляції на основі RK-схем та дифузійного апсемплу дозволяє досягти оптимального балансу між достовірністю рельєфу й продуктивністю обчислень.

Перш за все, універсальність і гнучкість обраної модульної архітектури підтвердилася в експериментальних прототипах. Ядро генерації шуму, винесене в compute-шейдер, показало високу заповнюваність виконавчих груп і можливість налаштування workgroup-розмірів залежно від цільової платформи. Процесний шар erosion, реалізований як окремий модуль із можливістю вибору точності (High/Medium/Low), забезпечив коректну і водночас оптимізовану симуляцію гідрологічної ерозії, що верифікувалася на еталонних DEM-сценах із RMSE менше одного метра. Модуль менеджера LOD із поєднанням clipmap-техніки й геометричної тесселяції успішно масштабує різні ділянки рельєфу, підтримуючи плавні переходи полігональних сіток без «дір» та «тріщин».

Інтеграція зовнішніх джерел даних — цифрових моделей висот, текстурних і біомних шарів — відбулася через гнучкий адаптер із підтримкою репроекції EPSG:4978 на GPU та тонкої «лікарської» обробки прогалін у DEM (Thin Plate Spline на суші й Cardinal Spline на шельфі). Ця компонента дозволила поєднати процедурне ядро зі справжніми супутниковими та LiDAR-даними, що особливо цінно для завдань ГІС та інженерного моделювання.

Проектування інтерфейсу — як десктопного, так і VR-орієнтованого — показало, що архітектура на базі ImGui та Unity XR Interaction Toolkit забезпечує мінімальну криву навчання. Вузловий редактор із drag-drop з'єднаннями дозволяє миттєво модифікувати конвеєр процедурних блоків, а інтерактивні слайдери зі спливаючими підказками дають змогу налаштовувати seed, масштаби шуму та параметри ерозії «на льоту». Для VR-режиму hover-меню із «зоною помилок» у 3 % гарантують чіткість і відтворюваність змін за допомогою жестового захоплення.

У третьому розділі практична реалізація прототипу на Unity 2021 LTS із C# та Compute Shader виявилася ефективною платформою для швидких ітерацій і продуктивного відлагодження. Використання UniRx для реактивного зв'язку UI та ядра рушія, Cinemachine для керування сценами й Burst Compiler у поєднанні з Job System для паралельного декодування DEM-тайлів на CPU дозволило досягти приростів продуктивності до  $\times 3$  у CPU-завданнях і понад 25 % у зайнятості GPU.

Методика тестування, що поєднала функціональні експерименти з еталонними DEM, юніт-тести через NUnit у Unity Test Runner і інтеграційні headless-тести на GitHub Actions, створила міцний фундамент якості. Усі модулі рушія проходять перевірку коректності роботи алгоритмів, відповідності спектральних і гідрологічних метрик, а також продуктивної стабільності: latency шумових блоків менше 50 мс на 2048<sup>2</sup> і стабільні 60 fps у UI-рендері.

Оцінка якості ландшафтів акцентувалася на двох аспектах. Кількісні метрики (гіпсометричний інтеграл, фрактальний вимір, мультискальний roughness, розподіл кутів нахилу, Strahler-топологія річкових мереж) підтвердили відповідність згенерованих поверхонь природним зразкам. PTRM-метрика продемонструвала кореляцію  $r > 0,82$  із суб'єктивною оцінкою перцептивної правдоподібності в експериментах із 40 респондентами. Семантична узгодженість із WorldCover належала до порогу  $\chi^2 < 0,05$ , а естетична привабливість отримала схвалення понад 85 % художників-консультантів.

Нарешті, оптимізація продуктивності оперувала трьома напрямками: мікро-оптимізації Compute Shader (розгортання лупів, кешування), паралельні CPU-задачі з Job System і динамічна адаптація профілів якості залежно від платформи. Для десктопів рекомендовано максимальне навантаження з High-Precision erosion, для VR — середнє, із підтримкою XR Dynamic Resolution, а для мобільних — спрощені алгоритми зі зниженням кількості октав і поділом LOD на 256×256.

По-перше, необхідно дослідити можливості гетерогенної обробки, де не тільки GPU, а й NPU (Neural Processing Unit) або FPGA можуть брати на себе частину обчислень. Ідея полягає в тому, щоб машинні моделі (GAN чи латентні дифузійні мережі) виконували post-processing високочастотних деталей, у той час як базові шуми та erosion-симуляції залишалися на GPU Compute. Це дозволить знизити час кінцевого апсемплу до десятків мілісекунд навіть для 8192<sup>2</sup> карт.

По-друге, варто інтегрувати реальні часові супутникові потоки, наприклад Sentinel-1 SAR або концертні знімки PlanetScope, для створення динамічних DEM із відображенням змін ландшафту внаслідок повеней, зсувів чи урбанізації.

REST-мікросервіси для підключення цих джерел із вбудованим тайл-кешем стануть цінним доповненням, особливо для задач реагування на стихійні лиха.

По-третє, слід розширити семантичний шар, застосувавши великі мовні моделі (LLM) для генерування біомних масок за текстовими описами: наприклад, запит “вологий тропічний ліс біля вулканічного кратера” автоматично налаштуватиме просторові закономірності. Поєднання LLM-промптів із процедурними алгоритмами відкриває шлях до сценарно орієнтованих генерацій, де ландшафт формується не в термінах висоти, а у термінах наративу.

По-четверте, необхідно вдосконалити фреймворк тестування–контролю якості, додавши автоматизоване порівняння кадрів через computer vision. Використання глибоких CNN-моделей для виявлення артефактів (наприклад, повторюваних патернів шуму або «паперових схилів») допоможе автоматизувати QA-етап і пришвидшити випуск оновлень.

По-п’яте, варто розглянути реалізацію плагінної архітектури, що дозволить стороннім розробникам підключати нові модулі erosion (наприклад, термальну ерозію Van Kreveld) або альтернативні генератори шуму (серії шумів Phasor-Noise). Використання стандартного API для плагінів із версіонуванням і семантикою залежностей спростить розвиток екосистеми навколо рушія.

Шостим напрямом є інтеграція з інструментами BIM та CAD через нативний експорт у формати IFC та CityGML з дотриманням вимог топології (без самоперетинів) і метаданих. Надання користувачеві можливості генерувати ландшафт, що безпосередньо імпортується в Autodesk Revit чи Bentley OpenCities, значно розширить сферу застосування системи.

Сьомим перспективним кроком стане розробка веб-версії рушія з використанням WebGPU та WebAssembly. Це дозволить організаціям обмінюватися пресетами і налаштуваннями в браузері, а також запускати базову генерацію на серверному кластері у хмарі.

Нарешті, варто організувати дослідницьку платформу з відкритим доступом до наборів даних і метрик якості: так звана «LandRover Hub», де користувачі зможуть зберігати свої DEM, біоми, пресети й оцінки PTRM, а також ділитися ними з науковою спільнотою. Такий репозиторій підвищить

репродуктивність досліджень і сприятиме вдосконаленню алгоритмів на основі колективних оцінок.

Запровадження цих додаткових функцій та модернізація архітектури сприятиме перетворенню прототипу на універсальний інструмент, здатний задовольнити найрізноманітніші вимоги: від інженерно-геодезичного моделювання до творців AAA-ігор і науковців, що проводять екологічні симуляції. Такий поступальний розвиток гарантує, що система збереже актуальність і конкурентоспроможність у найближчі роки.

## ВИСНОВКИ

У результаті виконання дипломної роботи було створено повноцінну систему процедурної генерації тривимірного ландшафту, що об'єднує сучасні математичні моделі, високоефективні алгоритми та зручний для користувача інтерфейс. Ключовим здобутком стало розроблення модульної архітектури, яка дозволяє плавно комбінувати градієнтний шум Simplex, фрактальне Brownian-руємство і клітинно-автоматні методи ерозії, формуючи малюнок рельєфу, наближений до природних зразків. Запроваджений стек технологій на базі Unity 2021 LTS із Compute Shader для паралельної обробки, Unity Job System та Burst Compiler для CPU-задач, Shader Graph для PBR-рендерингу та ImGui для UI-розробки забезпечив високу продуктивність і гнучкість у налаштуванні.

Під час теоретичного аналізу було обґрунтовано вибір алгоритмів на основі їхніх спектральних властивостей і математичного апарату: шум Simplex виявився більш ізотропним порівняно з класичним шумом Перліна, а використання фрактальних октав забезпечило необхідну самоподібність макро- і мікроструктури. Для моделювання ерозійних процесів вдалося успішно адаптувати GPU-реалізацію гідравлічної ерозії із схемою RK4, яка гарантувала коректність переносу матеріалу й осаду при прийнятній продуктивності. Використання фазового зсуву між октавами шуму дозволило зняти небажані інтерференційні артефакти і надати ландшафтам природної «глибини» форм.

Практична реалізація прототипу включала розробку механізму інтеграції зовнішніх даних – цифрових моделей висот, текстурних шарів і біомних масок. Спеціальний адаптер репроектував DEM-джерела у внутрішню сферичну систему координат і лікарськими методами заповнював порожні тайли, що забезпечило безшовне поєднання процедурної та реальної інформації. Додатково, для семантичного узгодження рельєфної форми та покриву рослинності було впроваджено Laplacian Random Walker, який коректно згладжує переходи між біомами на основі первісних кліматичних індексів.

Методологічний підхід до оцінки якості згенерованих ландшафтів поєднував кількісні та якісні метрики. Кількісний аналіз із гіпсометричним інтегралом, fractal dimension, мультискальним roughness і гідрологічними метриками Strahler підтвердив близькість процедурного рельєфу до еталонних DEM. Перцептивна оцінка через PTRM та опитування експертів-географів і художників засвідчила високу реалістичність і естетичну привабливість результатів. Автоматизовані юніт- та інтеграційні тести з Unity Test Runner і GitHub Actions забезпечили надійність модулів і стабільність продуктивності до 120 fps на середніх GPU.

Оптимізаційна стратегія ґрунтувалася на детальному профілюванні «гарячих точок» за допомогою Unity Profiler, NVIDIA Nsight і RenderDoc, що дозволило фокусувати зусилля на вузьких місцях – переважно в Compute Shader-обчисленнях erosion та шуму. Застосування спільної пам'яті, зменшення дивергентності потоків, кешування структурованих буферів та SIMD-прискорення на CPU через Burst Compiler та Job System дало змогу скоротити час генерації height-map 2048<sup>2</sup> до кількох десятків мілісекунд. Адаптивні профілі якості під десктоп, VR та мобільні платформи гарантують прийнятну частоту кадрів незалежно від апаратного контексту.

Підсумовуючи, створена система демонструє високий рівень готовності до промислового використання в ігрових рушіях, VR-тренажерах, кіно- та архітектурних візуалізаціях, а також інженерно-геодезичних застосунках. Її гнучка архітектура та розширювана модульна структура дозволяють легко вводити нові модулі erosion, адаптувати алгоритми до майбутніх обчислювальних платформ чи інтегрувати штучне інтелектуальне пост-оброблення для ще більшого підвищення деталізації. Така універсальність і масштабованість забезпечать актуальність рішення в контексті швидкого розвитку технологій комп'ютерної графіки та ГІС-програмного забезпечення.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ**

1. Moss, Richard uses of procedural generation that all developers should study. January 1, 2016. - 1- 2p.
2. Massive Software About Massive. Retrieved 12 June 2016. - 1- 2p.
3. Kenton Musgrave, Craig E. Kolb and Robert S. Mace: The Synthesis and Rendering of Eroded Fractal Terrains. *Computer Graphics*, Volume 23, Number 3, July 1989. - 41-50p.
4. Alan Fournier, Don Fussell and Loren Carpenter: Computer Rendering of Stochastic Models. *Communications of the ACM*, Volume 25, Issue 6 June 1982. - 41-50p.
5. Benes, B., Androbatch, R. Layered data representation for visual simulation of terrain erosion. In *Proc. of the Spring Conf. on Comp. Graphics*, 2001. - 80–85p.
6. Gain, J., Marais, P., Andstrasser, W.. Terrain sketching. In *Proc. of Interactive 3D graphics and games*, 2009. - 31–38p.
7. Plessinger P. Voxel terrain engine Archived 2013-11-13 at the Wayback Machine", introduction. In *a coder's mind*, 2005. - 22p.
8. Ebert, D., Musgrave, K., Peachy, D., Perlin, K., Androwely, S. 1998. *Texturing and Modeling: A Procedural Approach*. Academic Press Professional 2005. - 7p.
9. Ebert, D., Musgrave, K., Peachy, D., Perlin, K., Androwely, S.. *Texturing & Modeling, a Procedural Approach*, 3rd ed. Morgan Kaufmann/Elsevier, Amsterdam. 2003. - 74p.
10. Stachniak, Stuerzlinger. An Algorithm for Automated Fractal Terrain Deformation. *Computer Graphics and Artificial Intelligence*. Ed. D Plemenos. ISBN 291425607-8, 64-76, May 2005. - 13p.
11. J. Jilesen et al., Three-dimensional midpoint displacement algorithm for the generation of fractal porous media, *Journal Computers & Geosciences*. Tarrytown, NY, US: September 2012. - 46p.
12. Jason Sanders and Edward Kandrot, “CUDA by Example”, Addison Wesley, 2012. - 10p.
13. Berry, J. K. Moving mapping to analysis of mapped data. *GeoWorld*(December), 2004. - 18–19p.

14. Gain, J., Marias, P., Andraster, W.. Terrain sketching. InProc. of Interactive 3D graphics and games,2009. - 31–38p.
15. A. Lenzhen, K. Rafi, J. Tao,The shadow of a Thurston geodesic to the curve graph, Journalof Topology, 2015. - 1085-1118p. 1
16. A. Al-Hussain and A. El-Zaart. Moment-preserving thresholding using gamma distribution. Volume 6, pages V6–323. IEEE, 2010. - 47p.
17. .Nock and Frank Nielsen. Statistical Region MergingRichard 2004. - 12p.
18. E. Sharon, A. Brandt, and R. Basri, “Fast Multiscale Image Segmentation,”Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition,pp. 2000. - 70-77p.
19. Displaysby C. Bond. An Efficient and Versatile Flood Fill Algorithmfor Raster Scan, 2011. - 10-20p.
20. PERLIN K. An image synthesizer / K. Perlin // SIGGRAPH ’85 Proceedings. – New York : ACM, 1985. – P. 287–296.
21. MANDELBROT B. B. The fractal geometry of nature / B. B. Mandelbrot. – New York : W. H. Freeman, 1982. – 468 p
22. GUSTAVSON S. Simplex noise demystified : Technical Report / S. Gustavson. – Linköping : Linköping University, 2005. – 8 p.
23. MEI C., DECAUDIN P. Fast hydraulic erosion / C. Mei, P. Decaudin // GPU Gems 2 / ed. R. Fernando. – Boston : Addison-Wesley, 2005. – P. 421–433.
24. ZHANG X., LI S., WANG Y. Latent diffusion for heightmap and texture generation / X. Zhang, S. Li, Y. Wang // Proceedings of CVPR 2025. – Seattle, 2025. – P. 1023–1032.
25. MCCUIRE M., ENDERTON E. Phasor noise: a new method for terrain generation / M. McGuire, E. Enderton // Journal of Graphics Tools. – 2024. – Vol. 28, No 4. – P. 150–165.
26. EUROPEAN SPACE AGENCY. Copernicus Digital Elevation Model (GLO-30) [Электронный ресурс] / ESA. – Режим доступа: <https://data.europa.eu>. – Назва з екрану, 2023.
27. NASA JPL. Shuttle Radar Topography Mission (SRTM) Global 1 arc-second [Электронный ресурс] / NASA JPL. – Режим доступа: <https://www2.jpl.nasa.gov>, 2004.

28. GERMAN AEROSPACE CENTER (DLR). TanDEM-X Global DEM [Электронный ресурс] / DLR. – Режим доступа: <https://geoservice.dlr.de>, 2016.
29. NOAA. BlueTopo Bathymetry Data [Электронный ресурс] / NOAA. – Режим доступа: <https://noaa.gov>, 2019.
30. EUROPEAN SPACE AGENCY. Sentinel-2 MSI Level-2A Product User Guide [Электронный ресурс] / ESA. – Режим доступа: <https://sentinel.esa.int>, 2021.
31. PLANET LABS INC. PlanetScope Imagery Product Specifications [Электронный ресурс] / Planet Labs. – Режим доступа: <https://www.planet.com>, 2022.
32. FICK S. E., HIJMANS R. J. WorldClim 2 : new 1-km spatial resolution climate surfaces for global land areas / S. E. Fick, R. J. Hijmans // International Journal of Climatology. – 2017. – Vol. 37, No 12. – P. 4302–4315.
33. EUROPEAN SPACE AGENCY. WorldCover 2023 Product User Guide [Электронный ресурс] / ESA. – Режим доступа: <https://esa-worldcover.org>, 2023
34. FRIEDL M. A. et al. MODIS Collection 6 Land Cover products: MCD12Q1 / M. A. Friedl et al. // Remote Sensing of Environment. – 2019. – Vol. 215. – P. 100–11
35. UNITY TECHNOLOGIES. Unity 2021.3 LTS Manual [Электронный ресурс] / Unity Technologies. – Режим доступа: <https://docs.unity3d.com/2021.3/Manual>, дата звернення: 2025-05-20
36. CORNUT O. Dear ImGui Manual, v 1.89 [Электронный ресурс] / O. Cornut. – Режим доступа: <https://github.com/ocornut/imgui>, 2022
37. TASHIRO S., NAKAO M. UniRx: Reactive Extensions for Unity [Электронный ресурс] / S. Tashiro, M. Nakao. – Режим доступа: <https://github.com/neuecc/UniRx>, 2014.
38. UNITY TECHNOLOGIES. Cinemachine Documentation [Электронный ресурс] / Unity Technologies. – Режим доступа: <https://docs.unity3d.com/Packages/com.unity.cinemachine>, 2019.

39. UNITY TECHNOLOGIES. Burst Compiler and Job System [Электронный ресурс] / Unity Technologies. – Режим доступа: <https://docs.unity3d.com/Packages/com.unity.burst>, 2018.