

АНАЛІЗ МЕТОДІВ Й АЛГОРИТМІВ ГЕНЕРАЦІЇ ФРАКТАЛЬНИХ ШУМІВ ДЛЯ ЗАДАЧ ГЕНЕРАЦІЇ АМОРФНИХ ОБ'ЄКТІВ

Сілагін Олексій, Стрижалов Олександр

Вінницький національний технічний університет

Анотація

В даній роботі розглянуто та проаналізовано методи й алгоритми генерації фрактальних шумів, які можуть бути використані для виконання задач генерації аморфних об'єктів. Як найбільш ефективний, обрано алгоритм *Diamond-Square*.

Abstract

This paper examines and analyzes the methods and algorithms for fractal noise generation that can be used to perform amorphous object generation tasks. Diamond-Square algorithm is chosen as the most efficient one.

Вступ

В даний час програмна генерація об'єктів в комп'ютерній графіці стає все більше актуальною. Окремою і специфічною є задача генерації аморфних об'єктів, таких як вогонь, хвилі, хмари і т.д. Зростають вимоги до якості та фотореалістичності зображень, наприклад пейзажів, що в свою чергу підвищує вимоги до пам'яті, так як більш якісні зображення займають все більші обсяги пам'яті. Вирішення даної проблеми - використання процедурних шумів, що вимагає незначної кількості обчислювальних ресурсів для програмної генерації саме аморфних об'єктів. В цій роботі будуть проаналізовані методи й алгоритми генерації фрактальних шумів, які можуть бути використані для виконання задач генерації аморфних об'єктів.

Постановка задачі на дослідження

Для задач процедурної генерації 3D об'єктів фрактальної комп'ютерної графіки використовуються такі методи та алгоритми: шум Уорлі, шум Перліна, симплексний шум, хвильовий шум, градієнтний шум, шум *OpenSimplex* та алгоритм *Diamond-Square*. Проведемо порівняльний аналіз цих методів що до їх ефективності в реалізації саме аморфних фрактальних об'єктів.

Аналіз фрактальних шумів

Шум Уорлі - це функція шуму, запроваджена Стівеном Ворлі в 1996 році. Шум Уорлі близький до імітації текстур каменю або води. Основна ідея алгоритму полягає в тому, щоб взяти випадкові точки в просторі, а потім для кожної точки простору взяти відстань до n -ї найближчої точки (наприклад, другої найближчої точки) як інформацію про колір. Випадково розподілити особливі точки в просторі, шум $F_n(x)$ - відстань до n -ї найближчої точки до точки x . У випадку двох вимірів потрібно створити двадцять п'ять квадратів (у розміщенні п'ять на п'ять), щоб точно знайти найближчі. Зазвичай для практичних застосувань вважається достатньо дев'яти точок квадрата (у розташуванні три на три) [1]. Основне призначення – генерація кольорових текстур.

Симплексний шум - це метод побудови функцій шуму n -мірних розмірів, порівнянних із шумом Перліна ("класичний" шум), але з меншою кількістю артефактів, у більших розмірах, і з меншими обчислювальними накладними витратами. Кен Перлін розробив алгоритм у 2001 році з метою зняття обмежень класичної шумової функції.

Переваги симплексного шуму над шумом Перліна є те, що він має меншу обчислювальну складність і вимагає меншого застосування множення. Прості шкали

шуму до більш високих розмірів (4D, 5D) із значно меншими обчислювальними витратами порівняно з класичним шумом. Симплексний шум не має помітних спрямованих артефактів (візуально ізотропний). Симплексний шум можна легко реалізувати апаратно. Він корисний для програм комп'ютерної графіки, де шум зазвичай обчислюється для 2, 3, 4 або, можливо, 5 вимірів [2].

Хвильовий шум є альтернативою шуму Перліна, що зменшує проблеми згладжування та втрати деталей, які виникають при зведенні шуму Перліна у фрактал. Він концептуально відрізняється від частотного і градієнтного шумів, прикладами якого є шум Перліна та Симплекс-шум. Для багатьох застосувань кілька октав цього шуму можна генерувати та потім підсумовувати разом, як це можна зробити з шумом Перліна та шумом Simplex, щоб створити форму фрактального шуму [3].

Шум OpenSimplex - це функція шуму n-мірних градієнтів, яка була розроблена з метою подолання проблем, пов'язаних з патентом навколо симплексного шуму. Також дозволяє уникати візуальних артефактів спрямованості, характерних для шуму Перліна. Шум OpenSimplex використовує більший розмір ядра, ніж шум Simplex [4].

Алгоритм Diamond-Square - це метод генерації аморфних об'єктів для комп'ютерної графіки. Це трохи кращий алгоритм, ніж тривимірна реалізація алгоритму зміщення середньої точки, яка виробляє двовимірні пейзажі. Він також відомий як фрактал випадкового зміщення середньої точки, фрактал хмари або фрактал плазми, через ефект плазми, що утворюється при застосуванні. Алгоритм Diamond-Square починається з двовимірного квадратного масиву ширини і висоти $2n + 1$, де потім випадковим чином генерується значення точок. Спочатку потрібно встановити значення чотирьох кутових точок масиву. Потім кроки Diamond та Square виконуються поперемінно, поки не буде встановлено всі значення масиву. Крок Diamond: для кожного квадрата масиву встановити середню точку цього квадрата як середнє з чотирьох кутових точок плюс випадкове значення. Крок Square: для кожного ромбу в масиві встановити середню точку цього ромбу як середнє з чотирьох кутових точок плюс випадкове значення. При кожній ітерації величина випадкового значення повинна бути зменшена. Під час кроків Square точки, розташовані на краях масиву, матимуть лише три суміжні значення, а не чотири. Існує декілька способів розв'язання цієї проблеми: найпростіше - взяти середнє значення тільки трьох суміжних значень. Іншим варіантом є "обгортання", де четверте значення одержуємо з іншої сторони масиву. Цей метод також дозволяє згенерованим фракталам об'єднуватися без розривів. Алгоритм може бути використаний як для створення аморфних об'єктів, так і для створення реалістичних ландшафтів, і різні реалізації його широко використовуються в програмному забезпеченні комп'ютерної графіки. Він також застосовується для генерації компонент процедурних текстур [5].

Висновок

Серед розглянутих алгоритмів обрано алгоритм Diamond-Square, який порівняно з іншими краще відповідає вимогам та цілям генерації аморфних об'єктів.

Список використаних джерел

1. [Електронний ресурс] // Навчальні матеріали – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Worley_noise
2. [Електронний ресурс] // Навчальні матеріали – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Simplex_noise
3. [Електронний ресурс] // Навчальні матеріали – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Wavelet_noise
4. [Електронний ресурс] // Навчальні матеріали – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/OpenSimplex_noise
5. [Електронний ресурс] // Навчальні матеріали – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Алгоритм_Diamond-Square