УДК 004.921

ГРАФІЧНА РЕАЛІЗАЦІЯ СИМПЛЕКС ШУМУ У СТВОРЕННІ ЛАНДШАФТУ

Бабій В., Акбаш К.С.

Анотація. *В статті розглядається застосування створених комп'ютерних ландшафтів у сучасних сферах діяльності. Також висвітлюються основні проблеми створення динамічного ландшафту в комп'ютерній графіці. Основною метою статті є висвітлення теоретичних положень симплексного шуму при створенні програмного забезпечення для автоматизації створення рельєфних ландшафтів. Окрім того, стаття містить детальний опис батьківського алгоритму, створення процедурних текстур, шум Перлина. Їх спорідненість та можливість використання в створенні ландшафту.*

Ключові слова: *генерація ландшафту, карта висот, симплекс шум, шум перлина, комп'ютерна графіка.*

GRAPHICAL IMPLEMENTATION OF SIMPLEX NOISE IN THE CREATION OF THE LANDSCAPE

Abstract. *The article deals with the application of created computer landscapes in modern spheres of activity. The main problems of creating a dynamic landscape in computer graphics are also covered. The main purpose of the article is to highlight the theoretical positions of simplex noise when creating software for the creation of relief landscapes. In addition, the article contains a detailed description of the parent algorithm, the creation of procedural textures, the pearl noise. Their kinship and the ability to use in the creation of the landscape.*

Keywords: *landscape generation, map of heights, simplex noise, pearl noise, computer graphics.*

Постановка проблеми. Створення динамічного ландшафту – один із напрямів модернізації комп'ютерної графіки, метою якого є отримання динамічної карти висот, рельєфних структур та інших елементів ландшафту. Основним завданням є досягти реалістичність структур, мінімізації витрат комп'ютерних ресурсів та автоматизації створення динамічних ландшафтів.

Метою є створення програмного додатку, завдяки якому будуються випадкові ландшафти в 3D графіці. Користувач зможе задавати початкові дані з яких будується карта висот та 3D ландшафт для наочності.

Аналіз досліджень і публікацій. Модернізація комп'ютерної графіки є одна із актуальних проблем сучасності і для її вирішення розвивають апаратні

можливості техніки, модернізують та автоматизують алгоритми створення графічних структур.

Тому створення реалістичного ландшафту є один із актуальних напрямів людської діяльності. Практично жодна гра не обходиться без використання алгоритмів для створення ландшафтів та інших об’єктів комп’ютерної графіки.

Також використовують и в інших сферах діяльності таких як рекламній, видавничій справі та анімаційній. Число віртуальних галерей і розважальних парків швидко зростає. А з приходом автоматичних пілотованих апаратів комп'ютерна графіка стала використовуватися навіть в космічній галузі.

Яскравим прикладом використання процедурного алгоритму для створення ландшафтів є робота Маркуса Перссона (Notch) в грі Minecraft [2], де використали алгоритм шуму Перлина, який є батьком симплекс шуму [1] для створення ландшафтів.

Спочатку генерується середовище гри чи симулятору. Тобто генерується карта, де будуть відбуватися всі події. Генерацією карти називають процес випадкового створення географічних та геологічних об’єктів на карті при першому запуску гри на пустому слоті для ігрового світу. Процес генерації відображається на шкалі, якій можливо побачити при першій генерації карти. Але при цьому карта генерується не до кінця. Вона буде продовжувати генеруватися по мірі проходження вами по карті. Для того, щоб не завантажувати комп’ютер генерацією при першому запуску.

Симплекс шумом називають примітивом процедурних текстур, який відноситься до градієнтних шумів. Його використовують для поліпшення комп'ютерної графіки та зробити її реалістичнішою.

Алгоритм має широке застосування в кінематографії, телебаченні та симуляторах, створені за допомогою тривимірної комп'ютерної графіки. Також симплекс шум використовують за умов дуже обмеженої пам'яті, наприклад у демо-сценах, чи у графіці реального часу в іграх [3].

Метою статті є висвітлення теоретичних положень симплексного шуму при створенні програмного забезпечення для автоматизації створення рельєфних ландшафтів.

Алгоритм шуму Перлина. Шум Перлина реалізується довільною кількістю вимірів, але найчастіше в двох або трьох вимірною функцією. Алгоритм складається з трьох кроків: визначення сітки, обчислення скалярного добутку градієнтних векторів, та інтерполяція між цими значеннями [3].

Визначення сітки. Спочатку створюємо потрібну n-вимірну сітку. Треба кожній координаті присвоїти n-вимірний одиничний вектор. Наприклад для двох вимірної сітки, всім її координатам присвоюється випадковий вектор з одиничного кола, а для більшої кількості вимірів теж присвоюється випадковий n-вимірний одиничний вектор.

Одиничні вектори для двох вимірної сітки:

(1, 0)

(-1,0)

(0, 1)

(0,-1)

Одно чи двох вимірні сітки є тривіальні. Якщо більшої кількості вимірів, тоді для їх представлення пропонується наближення Монте Карло, де випадкові Картезіанські координати вибираються з одиничного куба, а точки за межами одиничної сфери відкидаються [3]. Алгоритм виконується поки для кожної координати сітки не отримаємо випадковий градієнт. Потім отримані вектори нормалізують.

Іноді для зменшення витрат на обчислення використовують кеш-таблиці та таблиці пошуку для зменшення кількості попередньо обчислюваних градієнтних векторів.

Скалярний добуток. Другим кроком буде визначення того, до якої комірки на сітці потрапить окрема точка. З кожного вузла сітки обчислюємо вектор відстані від точки до координат вузла. Потім до кожного вузла комірки обраховуємо скалярні добутки векторів та відстані градієнтних векторів.

На цьому кроці найбільша складність обчислень, при збільшенні вимірності сітки збільшується кількість обчислень в n разів. Наприклад, для двовимірної сітки алгоритм вимагає 4 операції. Отже складність обчислень становить $O(2^n)$.

Отже маємо 4 градієнтних вектори від чотирьох вузлів двохвимірної сітки. Також від кожного вузла спрямовані вектори (не нормалізовані) до цієї вибраної точки сітки (рис. 1).

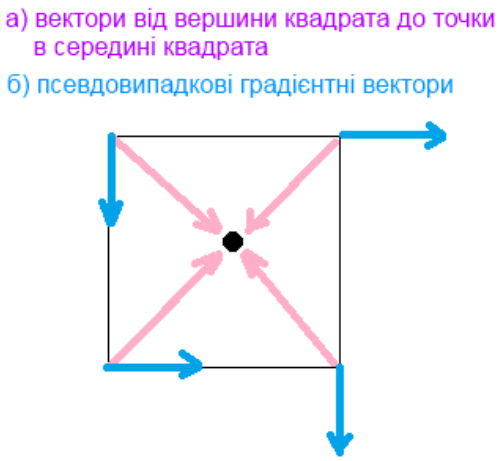

Рис. 1. Градієнтні вектори та вектор к центру

Інтерполяція. Кінцевим кроком є інтерполяція значень скалярних добутків, визначених для кожного вузла. Для інтерполяції використовують функцію, яка має першу або другу нульову похідну на обох кінцевих точках. Прикладом, може бути лінійна функція, для кінцевих точок на 0 та 1, зі значеннями a_0 та a_1, може бути такою:

$$f\ x\ = a_0\ 1 - x\ + a_1 x.$$

У комп'ютерній графіці зазвичай використовують функції шуму, що знаходяться у проміжку між [-1.0, 1.0]. Тоді результати шуму Перлина залишаються у цьому проміжку, інтерпольовані значення можуть коригуватися через масштабний фактор.

Симплекс шум. Симплекс-шум є модифікацією алгоритму Шуму Перлина і теж може складатися з довільної кількості вимірів. Реалізація цього алгоритму складається з чотирьох кроків: відхилення координат, поділ на симплекси, вибір градієнта і сумування ядер [1].

Відхилення координат. Вхідні координати трансформують за допомогою формул:

$$x' = x + \ x + y + \cdots \ * F$$
$$y' = y + \ x + y + \cdots \ * F$$
$$\ldots$$

де $F = \frac{\overline{n+1}-1}{n}$ [1].

Результатом є розміщення координатних точок на сітці, яка є представленням гіперкуба, роздавленого вздовж своєї головної діагоналі так, щоб відстань між точками (0, 0, …, 0) і (1, 1, …, 1) дорівнювала відстані між (0, 0, …, 0) і (1, 0, …, 0).

Отримані координати (x', y', …) визначають, в яку комірку зміщеного одиничного гіперкубу потрапляє вхідна точка, (xb'=floor(x'), yb'=floor(y'), …), та її внутрішні координати (xi'=x'-xb', yi'=y'-yb', …).

Поділ на симплекси. На наступному кроці відбувається сортування внутрішніх координат (xi', yi', …) за спаданням, щоб дізнатися до якої схеми Шлефлі симплекса відноситься вхідна точка. З вершин створюється результуючий симплекс, який відповідає впорядкованому обходу від (0,0, …, 0) до (1, 1, …, 1), також є n! варіантів, що можуть відповідати одній перестановці координат. Іншими словами, починаємо від нульової координати, потім послідовно додаємо ті, що відповідають найбільшому значенню внутрішньої координати, закінчуючи відповідником найменшого значення.

Нехай є точка (0.6, 0.8, 0,5), яка знаходиться в симплексі з вершинами (0, 0, 0), (0, 1, 0), (1, 1, 0), (1, 1, 1). Так як на вісі ординат значення найбільше, тоді знаходження координат симплекса починатимемо з нього. Далі збільшуємо значення по вісі абсцис, і так далі.

Вибір градієнта. Потім вершини симплекса додаються до базової координати відхиленого гіперкуба, результати хешуються у псевдовипадкові градієнтні вектори. Хешування реалізовується великою кількістю способів.

Для зменшення артефактів потрібно звертати увагу на вибір множини градієнтів.

Сумування ядер. Внесок кожної з n+1 вершин симплекса враховується за допомогою суми радіально симетричних ядер з центром у кожній вершині. Спочатку за допомогою оберненої формули визначаються не відхилені координати:

$$x' = x + \ x + y + \cdots \ * G$$
$$y' = y + \ x + y + \cdots \ * G$$
$$\ldots$$

де $G = \frac{1/\ \overline{n+1} - 1}{n}$ [4].

Ця точка віднімається від вхідних координат, щоб отримати не відхилений вектор переміщення. Цей вектор потрібен для двох цілей:

1) обчислити екстрапольоване значення градієнта з використанням скалярного добутку;

2) Визначити d^2, квадрат відстані до точки.

Далі внесок ядра кожної суми визначається з рівняння

$$(r^2 - d^2)^4 * (< \Delta x, \Delta y, \ldots > \cdot < grad.x,\ grad.y, \ldots >)$$

де r^2зазвичай обирають рівним 0,5 чи 0,6. 0,5 забезпечує відсутність розривів, в той час як 0,6 може покращити зовнішній вигляд у застосуваннях, де розриви непомітні. Значення 0,6 використовувався у оригінальній реалізації Кена Перлина.

Висновки. При написанні статті було проаналізовано еволюцію алгоритму шум Перлина в його модифікацію – Симплекс шум. Детально розглянуто основні проблеми створення динамічного ландшафту в комп'ютерній графіці. Також, проаналізовано основні теоретичні положення симплексного шуму та шуму Перлина, їх спорідненість та можливість використовувати в створенні ландшафту.

Список використаної літератури

1. Симплекс-шум [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Симплекс-шум

2. Вільшин Д. Алгоритм «diamond-square» для побудови фрактальних ландшафтів [Електронний ресурс] / Денис Вільшин // Хабрахабр. – 2011. – Режим доступу до ресурсу: https://habrahabr.ru/post/111538/

3. Шум Перлина [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Шум_Перлина.