

Програмне забезпечення для тривимірної візуалізації фракталів

Вінницький національний технічний університет

Анотація

Спроектовано функціональну систему для тривимірної візуалізації високодеталізованих моделей, для можливості відображення фракталів у просторі. Результатом дослідження є програмне забезпечення для високодеталізованої реалізації динамічних фігур, котре може використовуватись в наукових, ігрових та навчальних проектах.

Ключові слова: модель, шейдер, HLSL, Raymarching, Unity.

Abstract

A functional system for the three-dimensional visualization of highly detailed models has been designed to display fractals in space. The result of the research is software for highly detailed implementation of dynamic figures that can be used in scientific, gaming, and educational projects.

Keywords: model, shader, HLSL, Raymarching, Unity.

Вступ

Сучасні повсякденні комп'ютери здатні обчислювати кілька трильйонів операцій в секунду, що дозволяє виконувати багато різноманітних задач. Однак, навіть з такими показниками обчислювальних можливостей, в наш час цього все ще може бути недостатньо для розв'язку багатьох специфічних завдань. Зокрема існуючі методи відображення тривимірних моделей є недосконалими через значне навантаження від збільшення кількості деталей, тому такі завдання як візуалізація фракталів у просторі є занадто складними. Дане дослідження показує альтернативний підхід до відображення високодеталізованих моделей, зокрема й фракталів, що в десятки й сотні разів покращує продуктивність та деталізацію. Робота дозволяє значно збільшувати можливості деталізації, за рахунок використання відповідних комбінацій вхідних параметрів, що дозволяє відображати динамічні сцени та знизити поріг технічних вимог до виконання тих чи інших завдань.

Основний розділ

У межах дослідження було розроблено програмне забезпечення для тривимірної візуалізації фракталів, що базується на шейдерах мови HLSL. Головна ідея дослідження полягає в тому, аби розвантажити роботу центрального процесора передавши більшість обчислень графічній відеокарті пристрою, а також використовувати методи Raymarching задля зменшення й оптимізації можливих обчислень.

Високорівнева мова затінення (HLSL), яка є ядром проекрованої системи, дозволяє створювати швидкі й оптимізовані шейдери, оскільки використовує специфічно підлаштовані до графічної карти функції, що в результаті підвищує продуктивність центрального процесора дозволяючи паралельно виконувати великі потоки завдань. Також, за рахунок використання саме цієї мови, забезпечується високий рівень контролю над обчисленнями на рівні графічного процесора, а також підтримка багатьох платформ й пристроїв через її універсальність.

В основі шейдеру лежать raymarching підходи, щодо візуалізації тривимірної графіки. Основна ідея цих методів в тому, що промінь зміщується простором у відповідному напрямку, поступово наближаючись до поверхні об'єкта. На кожному кроці зміщення переобчислюється відстань до найближчої поверхні за допомогою знакової функції відстані signed distance function. Такий підхід дозволяє визначити, наскільки великим може бути слідуючий крок аби не пропустити можливої частини поверхні, при цьому з використанням відстеження сферою, буде оптимізовано кількість обчислень. Ray Marching передбачає відправлення променів на умовну сцену та перевірку зіткнень з

об'єктами, що в нашому випадку будуть математичними функціями відстані. Інакше кажучи, маючи камеру огляду, враховуючи її кут направляються промені з неї до кожного пікселя у вихідному зображенні. І як згадувалося вище, вздовж кожного променя з певним кроком перевіряється, чи присутнє умовне зіткнення з об'єктом функції. Якщо даний промінь зіткнувся з поверхнею, то для цього пікселя обчислення закінчені й далі немає сенсу їх проводити (відповідний піксель позначається значенням що позначатиме відстань, у контексті шейдера це будуть значення від -1 до 1, що візуально являтиме градації сірого), тому програма переходить до обчислення наступного; а якщо зіткнення не відбулось, то промінь продовжує збільшуватись на визначенні кроки до певного виставленого граничного значення. Описаний вище алгоритм, можна відобразити у вигляді схематичного малюнку (рис. 1).

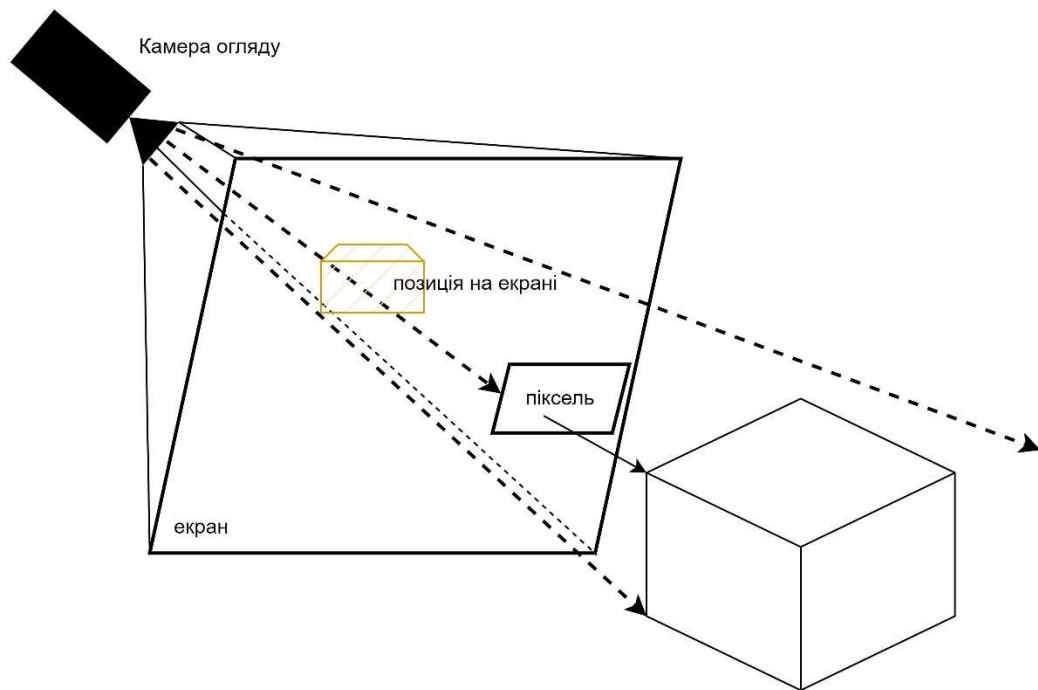


Рисунок 1 – Схематичне відображення трасування променя

Обрана мова HLSL також потребує і відповідного середовища для запуску й компіляції. Тому в ролі рендерингової системи, для інтеграції написаних обраною мовою шейдерів, було обрано графічний рушій Unity, котрий є одним з найоптимізованішим і найпопулярнішим серед наявних конкурентів. Зважаючи на цей вибір, для проєктованої системи дослідження відкривається доступ до багатоплатформенності, та можливості створення інтерфейсу користувача, який буде логічно поєднано із шейдером, завдяки підтримці C#. Наявність гнучкого інтерфейсу користувача відкриватиме великі можливості, щодо налаштування відповідної системи, та адаптації до різноманітних задач даного напрямку.

В першу чергу, було досліджено сутність й походження фракталів, а також можливі підходи до їх тривимірної реалізації, зокрема статичні моделі задані вручну користувачем, використання шейдерів для динамічного відображення моделей, а також створення самої форми в просторі. Варто зазначити, що через різну за походженням природу фракталів, як от комплексні, котрі базуються на математичних функціях з комплексними числами чи геометричні, котрі створюються шляхом геометричних дій за алгоритмом – різняться й спосіб їх підходу до візуалізації в тривимірному просторі. Для формування тривимірного вигляду фракталу було розглянуто фактичне видавлення двовимірного рисунку по товщині, що слугувало додаванням третьої осі, переведення з декартової до сферичної системи координат та ітеративних підхід комбінування відомих примітивних форм відповідного алгоритму візуалізації в просторі, котрий застосовувався для геометричних фракталів.

Серед можливих перспектив розвитку системи, є розділення на вузькоспеціалізовані підсистеми для визначення конкретних задач, як наприклад, моделювання турбулентності рідин, створення мереж, візуалізації судин в тілі а також дослідження природних явищ.

Висновки

Дана робота представляє альтернативне відображення деталізованих тривимірних моделей з можливістю динамічної зміни, що дозволяє детальніше вивчити поведінку й вигляд фракталів, систем числення, пояснення певних явищ у природі. Така система дозволяє проводити навчальні й наукові дослідження з мінімальними вимогами щодо апаратного забезпечення, що робить її доступною, передовою й актуальною.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Fractal Geometry of Nature Hardcover / Benoit B. Mandelbrot — 1982. — P. 468.
2. Real-Time 3D Rendering with DirectX and HLSL: A Practical Guide to Graphics Programming / Paul Varcholik — 2014. — P. 1443.
3. High-level shader language [Електронний ресурс]. Режим доступу: <https://learn.microsoft.com/en-us/windows/win32/direct3dhls/dx-graphics-hlsl>
4. Raymarching SDFs [Електронний ресурс]. Режим доступу: <https://iquilezles.org/articles/raymarchingdf/>
5. Unity Documentation [Електронний ресурс]. Режим доступу: <https://docs.unity.com/>

Бурдейний Олег Володимирович — студент групи ІСП-21б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: burdeyniy.007.com@gmail.com

Черняк Олександр Іванович — кандидат технічних наук, доцент кафедри обчислювальної техніки Вінницького національного технічного університету, Вінниця, e-mail: chernyak@vntu.edu.ua

Burdeinyi Oleh Volodymyrovych. — student of group ISP-21b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: burdeyniy.007.com@gmail.com

Cherniak Oleksandr Ivanovych — PhD, associate professor of the Department of Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: chernyak@vntu.edu.ua