

УДК 004.92

Гетман С.В.¹, Киричек Г.Г.²

¹ студ. гр. КНТ-511 НУ «Запорізька Політехніка»

² канд. техн. наук, доц. НУ «Запорізька Політехніка»

ВИКОРИСАННЯ ШЕЙДЕРІВ ДЛЯ СТИЛІЗОВАНИХ СЦЕН

Реалізація та оптимізація шейдерів для стилізованих сцен є важливою складовою сучасної комп'ютерної графіки, оскільки дозволяє не лише покращити продуктивність, а й впливати на сприйняття сцени [1]. Шейдери виступають інструментом художнього представлення, визначаючи освітлення, кольорову палітру та візуальні ефекти, що передають певний настрій або емоцію [2]. Завдяки правильній оптимізації, унікальний стиль досягається без надмірного навантаження на графічний процесор, що є особливо важливим при використанні інтерактивних застосунків [3]. Стилiзація сцени через шейдери дозволяє змінювати її сприйняття, підкреслюючи ключові елементи та надаючи їй індивідуальності. Тому проведення досліджень є актуальним для розробників, які створюють ефектні й атмосферні візуальні світи [1]. Метою роботи є проведення досліджень, аналіз, реалізація та оптимізація стилізованих шейдерів у середовищі Godot, що дозволить змінювати сприйняття однієї й тієї ж сцени, задаючи різний настрій та використовуючи прийоми *environmental storytelling*. Робота спрямована на створення ефективних графічних рішень, які передаватимуть емоційний і художній характер сцени без значного навантаження на продуктивність. Дослідження включає аналіз сучасних методів оптимізації шейдерів та пошук балансу між стилізацією і ефективним використанням ресурсів. Результатом має стати набір шейдерів, які можна інтегрувати у проекти із різними візуальними стилями, зберігаючи при цьому оптимальну продуктивність [4].

Для досягнення поставленої мети спочатку необхідно створити базову сцену в Godot, що використовується як основа при тестуванні стилізованих шейдерів [4]. Сцена матиме мінімальні зміни в 3D-моделях, оскільки головний акцент робиться саме на графічні ефекти. Реалізація шейдерів,

кожен з яких змінюватиме сприйняття сцени, планується із впливом на її атмосферу та підтримкою емоційного забарвлення. Це можуть бути ефекти, які імітують різні художні стилі, час доби, зміну настрою чи навіть історичні напрями. Окремий етап реалізації передбачає оптимізацію шейдерів, з метою досягнення відмінного візуального результату без значного навантаження на продуктивність [5]. Оптимізація передбачає мінімізацію обчислень у піксельному та вершинному шейдерах, використання текстурних атласів, ефективне управління освітленням і затіненням (рис. 1). У підсумку кожен створений шейдер має працювати з часом рендерингу не більше ніж 5 мс, що є важливою умовою для забезпечення стабільної продуктивності сцени [6].



Рисунок 1 – Сцена до та після пост-процесінгу

В процесі проведення досліджень та реалізації системи, створено декілька шейдерів, кожен із яких стилізуватиме сцену по-різному. Першим є Dither Shader який, при імітації градацій кольору за допомогою патернів, що нагадують стиль старих дисплеїв або принтерів, використовує техніку дізерингу. Другим є Volumetric Light Shader, який додає ефект об'ємного світла та розсіювання, створюючи атмосферний вигляд сцени. Cell Shader реалізує техніку toon shading, яка надає об'єктам вигляд намальованої графіки, схожій на аніме або комікси (рис. 2).

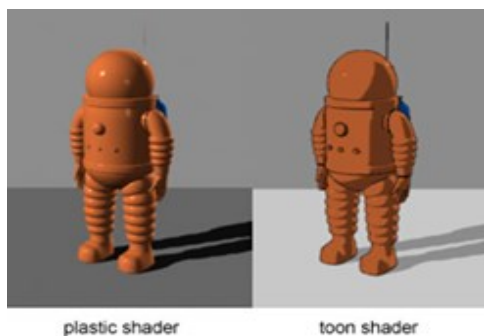


Рисунок 2 – Приклад cell-shaded моделі

Для додавання контурів навколо об'єктів, з метою кращого їх візуального виділення, реалізуємо Outline Shader, а для створення ефектів цифрових помилок планується впровадити Glitch Shader, який симулює артефакти пошкодження зображення. Іншу групу шейдерів реалізуємо: для відтворення вигляду старих кінескопних моніторів із характерними спотвореннями (CRT Shader); створення ефекту пікселізації, який можна використати як частину стилізації сцени або для підтримки ефектів переходу між рівнями деталізації (Pixelation Shader).

Важливим етапом проведення досліджень та впровадження системи є тестування реалізованих шейдерів у різних умовах застосування, з метою оцінки їх ефективності та впливу на продуктивність. А також аналіз і застосування різних методів та способів стилізації, таких як експерименти з кольоровими палітрами, контрастами та освітленням, які дозволяють кожному шейдеру створювати унікальний настрій або атмосферу [2]. Тому, кожен шейдер має бути оптимізований таким чином, щоб його рендеринг не перевищував 5 мс, що є критичним для забезпечення високої продуктивності. Це означає, що необхідно застосовувати різні підходи проведення оптимізації: мінімізація складних обчислень у піксельному і вершинному шейдерах; використання текстурних атласів; кешування результатів та інші методи, що дозволяють зберегти продуктивність без втрати якості візуальних ефектів. Важливим є також правильне використання цих технік та методів для підкреслення важливих елементів сцени, а також для підтримки концепції environmental storytelling, де реалізовані шейдери є інструментом передачі емоцій та історії без використання тексту чи додаткових елементів.

В результаті, за допомогою цих шейдерів можна створювати стилізовані сцени, які не лише виглядають яскраво і вражають, але й демонструють високу продуктивність. Оскільки всі шейдери повинні працювати без значного навантаження на систему, це дозволяє застосовувати їх в реальних проектах, не знижуючи комфорт гравців або користувачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Bailey M., Graphics shaders: theory and practice. / Bailey M., Cunningham S. // AK Peters/CRC Press, 2009. – 420 p.
2. Halladay K. Practical Shader Development: Vertex and Fragment Shaders for Game Developers. // Apress, 2019. – 381 p.
3. Dally W. J., Evolution of the graphics processing unit (GPU). / Dally W. J., Keckler S. W., Kirk D. B. // IEEE Micro. 2021. – Vol. 41. №. 6. – pp. 42-51.
4. Obuz K. Game Development with Blender and Godot: Leverage the combined power of Blender and Godot for building a point-and-click adventure game. // Packt Publishing Ltd, 2022. – 313 p.

5. Toivonen J. Creating a dynamic snow shader asset with shader-player interactions: industry pipelines in shader production. 2024. – 69 p.
6. Kim M. 3D graphics rendering pipeline implementation based on the openCL massively parallel processing. / Kim M., Baek N. A // The Journal of Supercomputing. 2021. – Vol. 77. – pp. 7351-7367.