

УДК 004.032.6

МЕТОДИ ОПТИМІЗАЦІЇ ТРИВИМІРНОГО КОНТЕНТУ ДЛЯ ІГРОВОГО РУШІЯ

Рогинський С.В., Гаєвий М.С.

Науковий керівник - к.т.н., проф. Колендовська М.М.

Харківський національний університет радіоелектроніки

(61166, Харків, пр. Науки, 14, кафедра Медіаінженерії та інформаційних
радіоелектронних систем, тел. (057) 702-15-87

e-mail: serhii.rohynskyi@nure.ua

In Unreal Engine 5, performance optimization is the process of modifying an engine and a project to make it work more efficiently and execute more rapidly. It generally focuses on improving some aspects of the game's performance, like frame rate, memory usage, disk space, bandwidth, and many others. Any game creator wants their game to be able to provide the best possible gameplay experience on the widest possible number of chosen devices. The game build should be performed in the most optimal way for a game to provide the acceptable frame rate and experience on a chosen set of devices. Optimization often helps to get to this point.

В Unreal Engine 5 оптимізація продуктивності — це процес модифікації двигуна та проекту, щоб зробити його ефективнішим і швидшим. Зазвичай він зосереджується на покращенні деяких аспектів продуктивності гри, наприклад частоти кадрів, використання пам'яті, місця на диску, пропускну здатності та багатьох інших. [1]

Запікання вершин високої роздільної здатності в нормальну карту для низькополігональної моделі може бути складним процесом, і багато факторів можуть знизити якість нормальної текстури карти до того часу, коли вона знаходиться в движку. Існує багато наборів інструментів для запікання нормальних карт, наприклад Substance Painter і Marmoset Toolbag. Щоб переконатися, що нормалі поверхні, які використовуються в процесі запікання, такі ж, як і в механізмі, слід експортувати оптимізовані нормалі з Unreal. Треба імпортувати модель запікання в Unreal, вибрати створення власних нормалей, а потім експортувати модель запікання з Unreal для запікання в Substance Painter. Це важливий крок під час створення високоякісних карт нормалей, оскільки Substance Painter має знати про нормалі поверхні сітки, щоб застосувати зсуви з моделі високої роздільної здатності.

Виклики малювання — це пошук активів, який відбувається кожного кадру. Кількість викликів малювання, які використовує ваша програма, залежить від кількості унікальних сіток у вашій сцені, а також кількості унікальних ідентифікаторів матеріалів, які використовує кожна сітка. Наразі низька графічна продуктивність є найбільшою причиною великої кількості викликів малювання, тому вам слід якомога менше їх зменшити.

Найпростіший спосіб зменшити виклики малювання — це зменшити кількість унікальних сіток, які відображаються у світі. Це можна зробити кількома способами: за допомогою вбудованих інструментів у рушії для об'єднання сіток і за допомогою відсічення видимості.

Існує кілька можливих варіантів зменшення кількості унікальних матеріалів на сітці. Найпростішим методом є використання програми на зразок Substance Painter, яка об'єднує кілька матеріалів в одну текстуру. Це дає змогу скористатися перевагами великої кількості типів матеріалів у дуже простому матеріалі Unreal, який потім можна використовувати як основу для екземплярів матеріалів із простим введенням текстури. Це також може зменшити кількість матеріальних інструкцій, що додатково покращує продуктивність. Складність матеріалу може збільшити вартість пікселів візуалізованого кадру. [2] Що більше матеріальних інструкцій є для кожного пікселя, то більше часу візуалізації потрібно витратити на обчислення його остаточного значення. Непрозорі матеріали є найдешевшими, але можуть значно відрізнитися залежно від моделі затінення або базового коду затінення.

Кількість інструкцій також збільшуватиметься залежно від кількості математичних функцій у матеріалі. Чим більше вузлів, тим дорожче обійдеться матеріал для обробки. Деякі специфічні операції також мають вищу вартість. Тому треба обмежити кількість інструкцій під час створення складніших матеріалів.

Світлопрозорі матеріали є одними з найдорожчих видів матеріалів. Окремі шари прозорості мають високу вартість пікселя, а коли кілька шарів прозорості складаються та відображаються, вартість набагато більша. Це відомо як овердрафт.

Для текстурування, можна використовувати вбудоване віртуальне текстурування, яке дає змогу використовувати текстури майже будь-якого розміру та відтворювати лише ті частини, які потрібні на екрані, або можна використовувати стандартні методи текстурування.

Таким чином, в роботі розглянуті методи оптимізації тривимірного контенту, що інтегровані в тому чи іншому вигляді у програмне забезпечення і дозволяють покращити процес проектування і відтворення тривимірного контенту

Список використаних джерел:

1. Pingle [інтернет ресурс]. – Режим доступу: <https://pinglestudio.com/blog/full-cycle-development/top-4-unreal-engine-tools-for-optimizing-game-development> (дата звернення: 12.04.2022)
2. Unreal Engine Documentation [інтернет ресурс]. – Режим доступу: <https://docs.unrealengine.com/5.0/en-US/guidelines-for-optimizing-rendering-for-real-time-in-unreal-engine/> (дата звернення: 13.04.2022)