

МЕТОДИ ОПТИМІЗАЦІЇ ВІЗУАЛЬНИХ ЕФЕКТІВ У UNREAL ENGINE

Беліков Д. Ю.

Науковий керівник – ст. викл. каф. ПІ Новіков Ю. С.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

e-mail: danylo.bielikov@nure.ua

This research examines optimization methods for visual effects in Unreal Engine, focusing on enhancing game performance and graphics quality. It highlights the importance of balancing visual fidelity with performance, especially in the FPS genre. The study investigates techniques such as LOD (Level of Detail) for particles, optimization of particle materials, particle batching, and simulation caching to increase FPS, ensuring fluid gameplay and high-quality visuals. The goal is to offer developers guidance on using these optimization tools and techniques effectively. Through the application of these methods in the development of 'Tale of The Awakened,' the paper illustrates how developers can enhance game performance and visual quality with minimal compromises.

У сучасному світі відеоігор, особливо у жанрі FPS (First Person Shooter – Шутер від першої особи), розробники невпинно прагнуть знайти ідеальний баланс між візуальною якістю та продуктивністю. Метою даного дослідження є аналіз та розробка методів оптимізації візуальних ефектів в Unreal Engine, спрямованих на досягнення цього балансу без значних компромісів. Зокрема, акцент робиться на оптимізації за допомогою системи Niagara, яка відіграє ключову роль у підвищенні FPS (Frames Per Second – Кадрів за секунду), забезпечуючи плавність ігрового процесу та збереження високої якості візуальних ефектів, критично важливих для занурення гравця у ігровий світ.

Використання LOD (Level of Detail – Рівень Деталізації) для частинок в Unreal Engine є ключовою стратегією для оптимізації візуальних ефектів, зокрема, в сценах з інтенсивним використанням частинок. Цей метод дозволяє зменшувати деталізацію частинок залежно від їх відстані до камери, тим самим знижуючи навантаження на графічний процесор і підвищуючи продуктивність, що, в свою чергу, дозволяє збільшити FPS (Frames Per Second – Кадрів за секунду) і поліпшити ігровий досвід без суттєвої втрати візуальної якості.

Таблиця 1 – Порівняння часу рендеру ігрових сцен із застосуванням методу Level of Detail на системі частинок та без

	Кількість систем частинок на сцені	Кількість трикутників що рендериться	GPU time для рендеру одного кадру	Загальний час рендеру одного кадру
Без використання LOD	500	968000	33 мс	36 мс
З використанням LOD	500	357000	25 мс	31 мс

Оптимізація матеріалів частинок є критично важливою для підвищення продуктивності в Unreal Engine, особливо в сценах з великою кількістю візуальних ефектів. Ефективне використання матеріалів включає мінімізацію складності шейдерів та оптимізацію текстур, що дозволяє знизити час обробки кожної частинки, при цьому зберігаючи високу якість візуальних ефектів.

Таблиця 2 – Порівняння часу рендеру ігрових сцен із застосуванням складних шейдерів на системі частинок та простих

	Кількість систем частинок на сцені	Кількість трикутників що рендериться	GPU time для рендеру одного кадру	Загальний час рендеру одного кадру
Складні шейдери (з використанням параметру прозорості)	500	968000	157 мс	159 мс
Прості шейдери (без використання фізичних властивостей відображення)	500	968000	25 мс	31 мс

Оптимізація візуальних ефектів через batching (групування частинок) у системі Niagara в Unreal Engine є ключовою для збільшення продуктивності гри, забезпечуючи високу якість візуальних ефектів без необхідності жертвувати продуктивністю. Batching знижує загальну кількість draw calls – команд, які процесор відправляє графічному процесору для відображення об'єктів на екрані, об'єднуючи обробку великої кількості частинок в один такий запит. Це критично важливо у сценах з інтенсивним використанням частинок, оскільки зменшення кількості draw calls безпосередньо призводить до збільшення продуктивності.

Таблиця 3 – Порівняння часу рендеру ігрових сцен із застосуванням batching'у та без

	Кількість систем частинок на сцені	Кількість трикутників що рендеряться	GPU time для рендеру одного кадру	Загальний час рендеру одного кадру
Без застосування batching'у (окремі емітери для кожного кольору частинок)	500	968000	45 мс	47 мс
З використанням batching'у (зміна кольору досягається за допомогою instance ID та його використанні у шейдері частинки)	500	968000	26 мс	30 мс

Кешування симуляцій у системі Niagara в Unreal Engine представляє собою передову практику оптимізації, яка дозволяє значно підвищити продуктивність ігрового процесу за рахунок зберігання результатів складних симуляцій для їх подальшого використання без необхідності повторного обчислення. Такий підхід особливо ефективний для візуальних ефектів, які вимагають великих обчислювальних ресурсів, таких як динамічні імітації рідин, вогню, диму або складних погодних умов. Використовуючи кешовані симуляції, можна досягти високої деталізації та реалістичності цих ефектів при одночасному зниженні навантаження на процесор і графічний адаптер, що підвищує загальну продуктивність і забезпечує плавність ігрового процесу. Проте, слід враховувати, що використання кешування симуляцій може мати певні недоліки, зокрема, збільшене використання оперативної пам'яті для зберігання кешованих даних. Це може стати проблемою для систем з обмеженими ресурсами. Крім того, кешування може обмежити гнучкість візуальних ефектів, оскільки попередньо обчислені симуляції не зможуть адаптуватися до динамічних змін у ігровому середовищі так само ефективно, як ефекти, обчислені в реальному часі.

Незважаючи на ці потенційні обмеження, правильно налаштоване кешування симуляцій може значно покращити візуальну якість та продуктивність ігор, розроблених з використанням Unreal Engine. Важливо знайти оптимальний баланс між використанням кешованих симуляцій для ефектів, які не вимагають високої динаміки, та збереженням обчислень в реальному часі для елементів, чия взаємодія з ігровим світом має бути максимально реалістичною та гнучкою. Ретельне планування, тестування та оптимізація симуляцій є ключовими для досягнення цієї мети,

дозволяючи розробникам створювати багаті та імерсивні ігрові середовища.

Таблиця 4 – Порівняння часу рендеру ігрових сцен із застосуванням кешованих систем частинок та тих, що прораховуються у реальному часі

	Кількість систем частинок на сцені	Кількість трикутників що рендериться	GPU time для рендеру одного кадру	Загальний час рендеру одного кадру
Без кешування (прорахунок у реальному часі)	500	968000	25 мс	650 мс
Закешована система	500	968000	25 мс	48 мс

На основі проведених тестів та аналізу різних методів оптимізації в системі Niagara в Unreal Engine можна зробити висновок, що ефективність кожного методу значною мірою залежить від конкретних умов та сценаріїв їх використання. Кешування складних симуляцій демонструє велику важливість для підвищення продуктивності, дозволяючи знизити обчислювальне навантаження і забезпечити плавність ігрового процесу. Водночас, інші методи, такі як інстанціювання частинок, хоча і є корисними для певних ситуацій, можуть мати менший вплив на загальну продуктивність в залежності від специфіки ігрового проекту. Важливо проводити індивідуальні тести та аналіз для кожного проекту, щоб ідентифікувати найефективніші стратегії оптимізації, зважаючи на унікальні вимоги та обмеження.

Список використаних джерел:

1. Комплекс навчально-методичного забезпечення навчальної дисципліни "Основи ігрової графіки" підготовки бакалавра для студентів усіх форм навчання спеціальності 121 – Інженерія програмного забезпечення: освітня програма "Програмна інженерія" / ХНУРЕ ; розроб. Н. А. Валенда. – Харків, 2017.

2. Офіційна документація з Unreal Engine: вебсайт. URL: <https://docs.unrealengine.com/5.3/en-US/> (дата звернення: 06.03.2024).

3. Онлайн форум з Unreal Engine: вебсайт. URL: <https://forums.unrealengine.com/tags/intersection/unreal-engine/blueprint> (дата звернення: 06.03.2024).