

Одинцов О. В.,
студент 4 курсу
фізико-математичний факультет
Науковий керівник: **Мосіюк О. О.,**
кандидат педагогічних наук, ст. викладач
кафедри прикладної математики та інформатики
Житомирського державного університету імені Івана Франка

ОПТИМІЗАЦІЯ ГРАФІКИ В UNITY 5.X

Тривимірною комп'ютерна графіка є однією із найбільш перспективних інформаційних технологій, яка розвивається найбільш динамічно. Це пов'язано із великою кількістю напрямів, де якісне представлення реалістичного 3D зображення є критично важливим для виробництва. До таких сфер необхідно віднести наступні напрями: створення візуальних ефектів у кінематографії, розробка комп'ютерних ігор, візуалізація наукових досліджень, проектування САПР тощо.

Аналізуючи питання оптимізації графіки у науковій літературі необхідно звернути увагу на таких авторів як: Князева Г. [4], Тихоміров Ю. [5], Хофман Кріс [1], Шикін А., Боресков А. [6], Шутс Дж. [3] та інші.

У контексті проаналізованої літератури зазначимо, що питання оптимізації не є достатньо розкритим для спеціалізованих ігрових рушіїв.

Таким чином метою статті є опис процесу оптимізації продуктивності графіки для ігрового рушія Unity 5.X.

Графічна частина комп'ютерної гри навантажує дві системи комп'ютера: GPU (графічний процесор) і CPU (центральний процесор). Стратегія оптимізації для GPU і CPU має істотні відмінності (іноді навіть виникає ситуація, коли при оптимізації для GPU більше навантаження лягає на CPU і навпаки).

Загалом процес візуалізації будь-якого тривимірного об'єкта на екрані комп'ютера складається із двох етапів. На початку CPU має з'ясувати, які джерела світла впливають на об'єкт, налаштувати шейдер і його параметри, відправити дані для опрацювання графічному драйверу, який підготує їх у вигляді спеціалізованих команд для виконання на відеокарті. Після формування відповідного списку вони опрацьовуються уже GPU.

Таким чином для зниження навантаження на CPU в Unity 5.X, необхідно: зменшити кількість видимих об'єктів (наприклад об'єднати близько розташовані об'єкти), використовувати менше матеріалів, об'єднувати текстури в великі текстурні атласи. Для оптимізації продуктивності CPU потрібно також слід враховувати, що об'єкти, які об'єднуються, мають використовувати одну текстуру.

Оптимізація продуктивності 3D графіки для GPU передбачає: зменшення кількості трикутних граней у тривимірній моделі; правильне моделювання UV-розгорток (зменшити кількість «швів» та жорстких ребер); застосовувати для розрахунку освітлення спеціалізовані карти (освітлювальні карти), оскільки це не вимагатиме значних обчислювальних потужностей; уникати ситуацій, коли кілька джерел світла освітлюють один об'єкт.

Важливим етапом оптимізації відображення графічних об'єктів у Unity 5.X є встановлення пріоритетності джерел світла. Для кожного

джерела світла його пріоритет обчислюється на основі відстані і яскравості певного джерела світла. Його можна змінити за допомогою параметра `Render Mode`, який може приймати два значення: `Important` або `Not Important`. Так джерела світла, позначені як `Not Important` мають більш низький пріоритет для опрацювання графічним рушієм ніж інші, у яких такі позначки відсутні.

Розглянемо приклад. Нехай розробляється комп'ютерна гра, у якій гравець керує автомобілем, що рухається в темряві із увімкнутими фарами. Однозначно, передні фари будуть найбільш важливим джерелом світла в грі і параметр `Render Mode` буде встановлений для них в значенні `Important`. Задні ліхтарі будуть менш важливі, не надаючи значного впливу на кінцеве зображення, так що для них `Render Mode` можна встановити в `Not Important` і тим самим заощадити апаратні ресурси.

Загалом при оптимізації будь-якої гри, яка створюється за допомогою ігрового рушія `Unity 5.X` варто дотримуватися наступних правил.

Використовуйте від 200 000 до 3 000 000 вершин в кожному кадрі, якщо цільова платформа `PC`.

Якщо застосовуються вбудовані шейдери, то необхідно перевіряти їх категорії `Mobile` і `Unlit`. Вони чудово працюють і на не мобільних платформах, але є спрощеними версіями складніших шейдерів.

Кількість різних матеріалів в сцені має бути мінімізовано, або ж використовуйте один матеріал для декількох об'єктів, якщо це можливо.

Користуйтеся властивістю `Static` для нерухомих об'єктів, щоб використовувати внутрішні оптимізацію `static batching`.

Не застосовуйте піксельні джерела світла, якщо немає необхідності. Також намагайтеся освітлювати один об'єкт тільки одним піксельним джерелом світла.


Застосовуйте архівацію текстур, коли це можливо, а також віддавайте перевагу 16-бітовим текстурі перед 32-бітними.

Намагайтеся мінімізувати використання систем для моделювання атмосферних явищ, якщо немає необхідності в цьому.

Підводячи підсумки зауважимо, що оптимізація графіки є важливим процесом, що дозволяє системам раціонально використовувати наявні обчислювальні ресурси. Запропоновані поради, щодо підвищення ефективності відображення тривимірних об'єктів для ігрового носія Unity 5.X також можливо використовувати і при створенні та рендері 3D моделей у таких середовищах як 3DS MAX, MODO, Maya, Blender тощо.

Література

1. Hoffman Ch. How to Improve Gaming Performance with Intel HD Graphics Chips / Ch. Hoffman, [Електронний ресурс]. – Режим доступу: <https://www.howtogeek.com/245592/how-to-improve-gaming-performance-with-intel-hd-graphics-chips/>
2. Optimizing graphics performance, [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html>
3. Schuetz J. Comparing Game Engines: Unity vs Unreal vs Corona vs GameMaker / J. Schuetz, [Електронний ресурс]. – Режим доступу: <https://www.pubnub.com/blog/comparing-game-engines-unity-unreal-corona-gamemaker/>.
4. Князева Г. В. Методы оптимизации программ компьютерной графики OpenGL / Г. В. Князева. [Электронный ресурс]. – Режим доступу: <https://cyberleninka.ru/article/v/metody-optimizatsii-programm-kompyuternoy-grafiki-opengl>.
5. Тихомиров Ю. Программирование трехмерной графики / Ю. Тихомиров. – СПб.: ВУН, 1998. – 256 с.



6. Шикин А. В., Боресков А. В. Компьютерная графика.
Полигональные модели / А. В. Шикин, А. В. Боресков. – М.:
ДИАЛОГ-МИФИ, 2001. – 288 с.