

УДК 004.92

ВИКОРИСТАННЯ МЕТОДІВ ОБРОБКИ ЗОБРАЖЕНЬ ДЛЯ ОПТИМІЗАЦІЇ 3D-МОДЕЛЕЙ У КОМП'ЮТЕРНИХ ІГРАХ

Ніколаєв І.Д.

e-mail: illia.nikolaiev@nure.ua

Науковий керівник – Ібулаєв В.В.

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

The article discusses the main methods of optimizing 3D models for computer games, ensuring a balance between image quality and performance. Particular attention is paid to the techniques of levels of detail (LOD), normal maps and texture compression methods. The use of these approaches allows to reduce the load on hardware resources, improving the performance of the game engine. Also, modern algorithms and tools used in popular game engines, such as Unity and Unreal Engine, are considered.

Розвиток комп'ютерних ігор та підвищення якості графіки призводять до зростання вимог до апаратного забезпечення. Для забезпечення стабільної роботи ігрового рушія необхідна ефективна оптимізація 3D-моделей. Оптимізація передбачає зменшення кількості полігонів, покращення роботи з текстурами та застосування різних технологій обробки зображень. Сучасні методи дозволяють зменшити використання відеопам'яті та потужностей процесора, забезпечуючи при цьому якісне зображення. Основними техніками оптимізації є рівні деталізації (LOD), нормал-мапи та методи стиснення текстур.

Основні методи оптимізації 3D-моделей

1. Рівні деталізації (LOD)

Метод рівнів деталізації передбачає створення кількох варіантів однієї 3D-моделі з різною кількістю полігонів (рис. 1). Ігровий рушій вибирає відповідний рівень залежно від відстані до камери. Це значно зменшує кількість обчислень для об'єктів, що знаходяться далеко, тим самим покращуючи продуктивність [1]. Наприклад, у сучасних іграх використовується технологія автоматичного генератора LOD-моделей, яка спрощує процес оптимізації для розробників.



Рисунок 1 – Приклад рівнів деталізації LOD

2. Нормал-мапи

Нормал-мапи (normal maps) дозволяють зменшити кількість полігонів, зберігаючи деталізацію поверхні об'єкта. Вони використовуються для імітації рельєфу, створюючи ілюзію додаткової геометрії без збільшення кількості вершин. Використання нормал-мап значно покращує продуктивність у випадку складних моделей, зокрема при створенні персонажів або об'єктів оточення [2].

Методи генерації нормал-мап включають використання високополігональних моделей, з яких запікаються карти нормалей, що дозволяє отримати реалістичні тіні та рельєф без збільшення полігонального бюджету сцени (рис. 2).

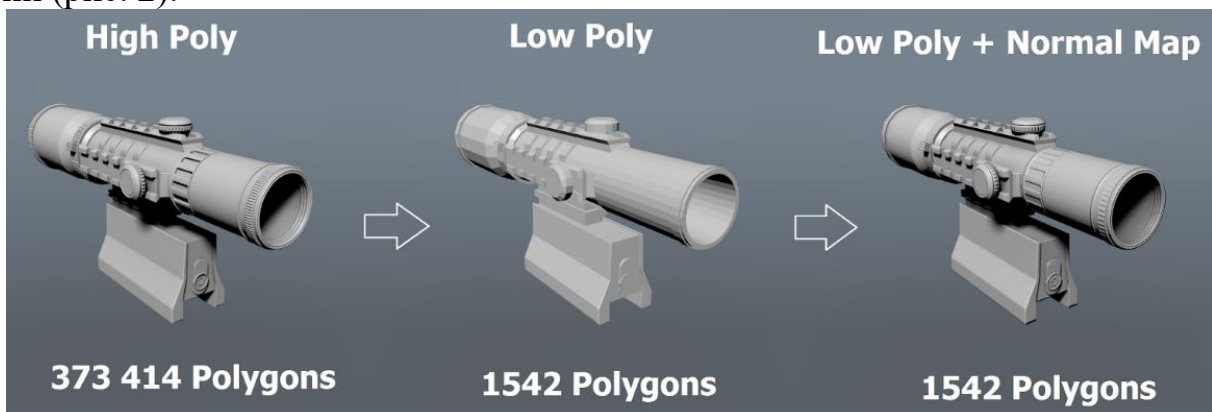


Рисунок 2 – Результат використання нормал-мапи

3. Методи стиснення текстур

Текстури займають значну частину відеопам'яті, тому їх стиснення є важливим кроком оптимізації. Існують різні алгоритми, такі як DXT, BC та ASTC, які дозволяють зменшити розмір файлів без помітної втрати якості. Наприклад, у рушії Unity існує можливість автоматичного стиснення текстур залежно від платформи, що дає змогу адаптувати гру під мобільні пристрої, ПК та консолі.

Додаткові методи оптимізації

Окрім основних технік, існує ряд додаткових методів, які дозволяють знизити навантаження на апаратне забезпечення:

- Baking (запікання текстур та світла) дозволяє зменшити кількість динамічних джерел світла, що знижує навантаження на графічний процесор;
- Occlusion Culling (вимкнення невидимих об'єктів) – це алгоритм, який не рендерить об'єкти, що знаходяться поза зоною видимості камери [1];
- Mipmaps (мультирівневі текстури) – це створення декількох версій текстури з різним рівнем деталізації, що дозволяє використовувати оптимальну версію в залежності від відстані до об'єкта.

Висновки

Оптимізація 3D-моделей у комп'ютерних іграх є необхідною для підвищення продуктивності без втрати візуальної якості. Методи LOD, нормал-мапи та стиснення текстур є ефективними підходами, що дозволяють знизити навантаження на апаратне забезпечення. Використання цих методів сприяє створенню ігор з високою якістю графіки та плавною роботою на різних пристроях. Крім того, застосування додаткових технік, таких як запікання текстур, виключення невидимих об'єктів та мультирівневі текстури, дозволяє ще більше покращити продуктивність. Подальший розвиток алгоритмів оптимізації сприятиме створенню ще більш реалістичних та ефективних візуальних ефектів у майбутніх відеоіграх.

Список використаних джерел:

1. Unreal Engine Documentation. Level of Detail (LOD) in Unreal Engine. <https://docs.unrealengine.com>;
2. Unity Documentation. Texture Compression Guide. <https://docs.unity3d.com>.