

ОПТИМІЗАЦІЯ 3D МОДЕЛЕЙ ДЛЯ ІГРОВИХ РУШІВ

Манченко А.В.

Науковий керівник – асист. Солодов В.Д.

Харківський національний університет радіоелектроніки, каф. МІРЕС,

м. Харків, Україна

e-mail: andrii.manchenko@nure.ua

Over the past decades, the computing power of computers has grown rapidly. Thanks to this, the modern gaming industry has the opportunity to create graphics that are close to reality, and in recent years projects have appeared whose graphics can hardly be distinguished from realism. However, even with current capabilities, optimization of 3D models should be observed when creating them, since without it it will be impossible to fit many objects into one game, as well as perform high-quality animation. This work will describe ways to optimize 3D models for successful export to game engines.

На сьогоднішній день існує велика кількість комп'юторних ігор різної якості графіки. З кожним роком параметри гарної графіки змінювалися разом з розвитком технологій. Протягом розвитку технологій у розробників завжди стояла ціль створити відмінну графіку для ігор, яка була б здатна викликати позитивні враження у гравців, а також могла бути реалізована завдяки сучасним можливостям техніки. Як раз для реалізації таких задумок розробники вивчали оптимізацію 3D моделей для створення ігор, які повинні були мати технічно вірний підхід та вражаючу візуальну сторону.

Для оптимізації моделі, необхідно знати наступні параметри:

По перше, найважливішим пунктом є кількість полігонів моделі, чим більше полігонів тим більш вона є детальною, але при цьому відповідно витрачає потужність техніки. Сам по собі полігон складається з точок (вершин), з'єднаними між собою ребрами. У 3D графіці існує 3 види полігонів: триси (3 вершини), квади (4 вершини) та n-гони (5 та більшк вершин). Стандартом в процесі моделювання є квади, так як чотирикутна форма є зручною для роботи та дозволяє досягти коректної геометрії моделі. Для кожної вершини полігону необхідно зберігати її координати у тривимірному просторі (x, y, z), що зазвичай становить 3 числа з плаваючою точкою (зазвичай 32 біти на кожне число). Таким чином, на кожен вершину полігону потрібно $3 \cdot 32 = 96$ біт або 12 байт. Враховуючи, що у вас 4 вершини, загальний обсяг даних для зберігання координат вершин становить $4 \cdot 12$ байт = 48 байт.

По друге, оптимізація моделі залежить від роздільної здатності текстур, накладених на модель. Тут все по тому ж принципу як з полігонами, чим більше роздільна здатність текстури, тим якісніше буде виглядати модель, але при цьому буде більше використовуватись обчислювальна потуж-

жність техніки.

Створення 3D моделі завжди починається з редагування мешу (сітки моделі), який складається з сукупності полігонів. Саме тому оптимізацію слід починати зі зменшення кількості полігонів. Процес перебудови мешу називається ретопологія (retopology). Одним з найкращих інструментів ретопології є ZRemesher у програмі Zbrush. Завдяки ньому можна швидко зменшити розмір моделі навіть на мільйони полігонів. Додатково до цього інструменту існує пензлик «ZRemesherGuides», який дозволяє локально корегувати сітку. Зокрема це може бути зручно при створенні ретопології моделі обличчя, так як ділянки навколо очей, носа та рота повинні мати округлу будову полігонів. Цей інструмент дозволяє вказати бажану кількість полігонів та здатен за дуже короткий час зробити коректну сітку моделі. Цей інструмент настільки гарно себе зарекомендував, що для інших 3d пакетів, таких як Blender, Maya, 3Ds Max було створено плагіни з цим інструментом.

Але автоматичної ретопології часто буває недостатньо, особливо тому що далеко не всі інструменти вийшли на той рівень щоб можна було повністю покладатися на них. Тому частіше всього розробники роблять ретопологію вручну.

В 3D графіці існують такі поняття як high poly та low poly. High poly називають моделі з великою кількістю полігонів, а моделі з невеликою кількістю полігонів – low poly. High poly моделі ніколи не використовуються в ігрових рушіях через велику кількість полігонів. Необхідність в них виникає через те, що такі моделі мають велику деталізацію, яку потім можна буде передати на low poly моделі, які як раз і підходять для імпорту в ігрові рушії.

Ручна ретопологія виконується наступним чином: поверх high poly моделі починають вибудовувати новий меш, який повинен містити значно меншу кількість полігонів, але також повторювати форму наскільки це можливо, прилягаючи до оригіналу. Цей самий новий меш і бути називається low poly. Для того щоб новий меш гарно прилягав до high poly, слід використовувати інструменти, які дозволяють примагнічувати новий меш поверх high poly, а потім вже будувати оптимізовану версію моделі. Зазвичай прийнято створювати ручну ретопологію у програмі Maya 3D, зокрема тому що вона є однією з програм, які використовують для створення гейм-дев контенту.

Також, роблячи ретопологію, слід урахувати чи буде анімуватись модель або вона буде статичною. Ретопологія, яка робиться для моделей що будуть анімуватись, має містити певну будову мешу, яка буде дозволяти робити коректну деформацію. Якщо, наприклад, створюється модель людини, то потрібно опрацювати місця згинання суглобів, таких як коліна, лікті, кисті рук, фаланги пальців тощо. Особливих вимог потребує обличчя, так як воно є найскладнішою частиною тіла в процесі анімації. Існує ще

один момент, стосовно оптимізації геометрії. Перед тим як експортувати модель, з нею потрібно провести таку процедуру як триангуляція. Триангуляція – це процес розбиття полігонів на трикутники. Це дозволяє робити обробку графічної інформації більш ефективно, оскільки більшість алгоритмів та операцій розроблені для роботи з трикутниками. Трикутник є простою геометричною фігурою, що полегшує їх обробку та візуалізацію.

Для того щоб текстура моделі виглядала добре і при цьому не використовувала зайву потужність техніки, розробники користуються таким поняттям як Texel Density. Texel Density – це величина, яка є співвідношенням розміру текстури у пікселях до габаритів 3D моделі у сцені. Texel Density характеризує щільність, тобто якість текстури. Виглядає в техзаданні приблизно так: 256px/m, 128px/inch. Тобто щоб розрахувати Texel Density, необхідно розділити здатність текстури поділити на розмір геометрії. Таким чином, текселерація визначає, яка площа текстури буде віддана на модель виходячи з її розмірів. Високий Texel Density означає більшу деталізацію текстури, низький - меншу, розмиту текстуру.

Існує відносна текселерація (в межах однієї моделі) та абсолютна (в межах усієї ігрової сцени.) Відносна текселерація повинна відповідати таким вимогам: єдиний Texel Density по всій моделі (допускається незначне збільшення текселю на дрібних деталях UV розгортки). З цього правила є ряд винятків: важливі деталі повинні займати більше місця на розгортці, наприклад, обличчя персонажа повинно мати більший Texel Density. Ті деталі, які не видно у кадрі або погано видно повинні мати менший тексель. Також, для зручності роботи, можна використовувати скрипти, яку допоможуть підрахувати Texel Density, такі як: Texel Density Checker у Substance Painter, UV Tools в Blender, Maya Texel Density Tool.

Користуючись цими правилами та методами, можна створювати якісні комп'ютерні ігри, які будуть здатні запускатися на багатьох пристроях, що вже є високою перевагою, адже ігри які потребують необґрунтовано багато потужності мають погану репутацію та низькі рейтинги як серед критиків, так и серед звичайних гравців.

Список використаних джерел

1. Retopology: Unlocking new horizons in 3D artistry. URL: <https://www.autodesk.com/solutions/retopology> (дата звернення 05.03.2024)
2. Topology Guides URL: <https://topologyguides.com/> (дата звернення 05.03.2024)
3. Texel Density URL: <https://www.beyondextent.com/deepdives/deepdive-texeldensity> (дата звернення 05.03.2024)