

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів проектування 3D простору для "triple-A" проектів
(тема)

Виконав:

Студент 2 курсу, групи ПІЗМ-22-2

Бронов І.В.
(ім'я, ініціали)

Спеціальність 121 Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Керівник проф. Смеляков С.В.
(посада, ім'я, прізвище)

Допускається до захисту

зав. кафедри

(підпис)

Дудар З.В.
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук
(повна назва)
Кафедра _____ програмної інженерії
(повна назва)
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 121 Інженерія програмного забезпечення
(код і повна назва)
Тип програми _____ освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітньо-наукова програма _____ 121 Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ

Зав. кафедри _____
(підпис)

« ____ » _____ 2024.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Бронову Іллі Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів проектування 3D простору для "triple-A" проектів»
Затверджена наказом університету від «29» березня 2024 р.
№ 250Ст
2. Термін подання студентом роботи до екзаменаційної комісії 10 червня 2024 р.
3. Вихідні дані до роботи: дослідження методів та засобів проектування 3D простору для "triple-A" проектів, дослідження ігрових рушіїв для різних потреб проекту.
4. Перелік питань, що потрібно опрацювати в роботі: аналіз предметної області методів проектування 3D простору для "triple-A" проектів, дослідження існуючих ігрових рушіїв, вдосконалення ігрових рушіїв при розробці AAA проектів, практична реалізація, проведення експерименту.

КАЛЕНДАРНИЙ ПЛАН

№	Назви етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області методів проектування 3D простору для "triple-A" проектів,	01.04.2024	виконано
2	Дослідження існуючих ігрових рушіїв,	10.04.2024	виконано
3	Удосконалення ігрових рушіїв при розробці AAA проектів,	20.04.2024	виконано
4	Практична реалізація	30.04.2024	виконано
5	Проведення експерименту	10.05.2024	виконано
6	Захист кваліфікаційної роботи	07.06.2024	виконано

Дата видачі завдання «1» квітня 2024 р.

Студент

_____ (підпис)

Ілля БРОНОВ

(прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Сергій СМЕЛЯКОВ

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота містить: 68 с., 17 рис., 2 таблиці, 24 джерел., 6 додатків

АНАЛІТИКА, ПАТЕРН, ПРОГРАМНА СИСТЕМА, ІГРОВІ РУШІЇ, AAA ПРОЕКТ.

Об'єктом дослідження є методи та засоби проектування 3D простору для "triple-A" проектів.

Метою роботи є дослідження предметної області методів проектування 3D простору для "triple-A" проектів, дослідження існуючих ігрових рушіїв, порівняльний аналіз та виведення рекомендацій.

В результаті буде знайдено оптимальний із сучасних ігровий рушій для розробки AAA проектів.

The thesis note contains 68 pages, 17 fig., 24 references, 6 apps.

ANALYTICS, PATTERN, SOFTWARE SYSTEM, GAME ENGINES, AAA PROJECT.

The object of research is the methods and means of 3D space design for "triple-A" projects.

The purpose of the work is to research the subject area of 3D space design methods for "triple-A" projects, research of existing game2 drivers, analysis, their comparative analysis and derivation of recommendations.

As a result, the optimal modern game engine for the development of AAA projects will be found.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу EIArKhNURE.

Я, Бронів Ілля Вікторович, студент(ка) гр. ІІЗм-22-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів проектування 3D простору для "triple-A" проектів, що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ	7
1 Аналіз предметної галузі	8
1.1 Загальний огляд 3D рушіїв та сфери їх застосування.....	8
1.2 Актуальність проблеми	10
1.3 Постановка задачі	11
2 Дослідження існуючих рушіїв.....	12
2.1 Дослідження рушія UnrealEngine.....	12
2.2 Дослідження рушія CryEngine.....	16
2.3 Дослідження рушія Unity.....	19
3 Вдосконалення методу проектування 3D-простору.....	24
3.1 Вдосконалення ігрових рушіїв при розробці AAA проектів.....	24
3.2 Переваги UnrealEngine із Nanite Geometry.....	26
3.3 Трасування променів у UnrealEngine.....	29
4 Практична реалізація.....	36
4.1 Основні функціональні методи.	36
4.2 Вхідні дані до тестування	40
5 Проведення експерименту	45
5.1 Результати бенчмаркінгу	45
5.2 Аналіз результатів	45
Висновки.....	48
Перелік джерел посилання.....	49
Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії.....	52
Додаток Б Архітектурна структура unity, її компоненти та зв'язок між ними	53
Додаток В Результати кадрів в секунду для кожної із 3х відстаней (ітерацій LOD).54	54
Додаток Г Слайди презентації.....	58
Додаток Д Результати перевірки на академічний плагіат.....	67

Додаток Ж	Експертний висновок	результатів перевірки	
роботи.....			68

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – Application Programming Interface

LOD – Level of Detail

FPS – Frames Per Second

CPU – Central Processing Unit

VR/AR – Virtual Reality / Augmented Reality

PC – Personal Computer

ВСТУП

Сучасний світ неможливо уявити без графічного контенту, це може бути все що завгодно, починаючи від маленьких зображень та компактних ігор – закінчуючи великими ігровими AAA проектами та графікою, яка використовується для зйомок фільмів.

Але все більшої популярності набирають віртуальні світи різної направленості, частіше за все це ігрові проекти, є маленькі а є великі на сотні годин проходження з великим відкритим світом, саме останні користуються великою популярністю серед геймерів по всьому світу.

На подібні проекти є попит, відповідно багато компаній та груп розробників та гейм-дизайнерів займаються розробкою подібних проектів.

Але це неможливо реалізувати без набору інструментів, подібний функціонал надають великі ігрові рушії для створення та моделювання трьох мірного простору. Таких рушіїв є доволі багато, тож розробникам необхідно без помилково визначитись із вибором рушія для їх проекту, адже кожен рушії має свої особливості.

Саме тому проведене дослідження із аналізу рушіїв та створення рекомендацій, опираючись на які розробники зможуть більш впевнено робити вибір щодо найбільш підходящого для них ігрового рушія є актуальним.

Результати роботи оформлені у вигляді пояснювальної записки до кваліфікаційної роботи, розділи якої містять: аналіз предметної області методів проектування 3D простору для "triple-A" проектів, дослідження існуючих ігрових рушіїв, вдосконалення ігрових рушіїв при розробці AAA проектів, практичну реалізацію, проведення експерименту. Кваліфікаційна робота оформлена відповідно до діючих стандартів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальний огляд 3D рушіїв та сфери їх застосування

Ігрові 3D рушії представляють собою комплексні програмні платформи, які об'єднують широкий спектр технологій та інструментів для створення тривимірних відеоігор.

Вони обробляють усе, від візуалізації складних графічних ефектів та реалістичної фізики до управління звуком та інтерактивного геймплею. Основою рушія є потужний графічний движок, який взаємодіє з фізичним движком для симуляції реалістичних взаємодій, а також з аудіо системою для створення звукового оформлення.

Штучний інтелект, системи скриптів та мережеві можливості дозволяють створювати комплексні та багатокористувацькі ігрові світи, а інтегровані інструменти розробки та підтримка різних платформ спрощують процес створення ігор. Все це робить 3D ігрові рушії ключовими інструментами у сучасній індустрії відеоігор, дозволяючи розробникам перетворювати креативні ідеї на захоплюючий ігровий досвід [1].

Ігрові рушії виникли як відповідь на потребу стандартизації та ефективності у розробці відеоігор. У ранні дні ігрової індустрії, розробники створювали ігри з нуля, що було затратним по часу та неефективним. Впродовж часу вони почали розуміти, що багато аспектів ігрової розробки можна пере використовувати між проектами.

Це призвело до створення перших ігрових рушіїв, які були сукупностями коду, призначеного для управління графікою, фізикою, звуком, та іншими ключовими аспектами ігрового процесу.

З часом ці рушії стали більш розвиненими, інтегруючи інструменти для моделювання, анімації, штучного інтелекту, та навіть мережевої гри. Сьогодні ігрові рушії відіграють ключову роль у ігровій індустрії, спрощуючи розробку та дозволяючи створювати більш складні та інтерактивні ігрові досвіди.

Перед аналізом AAA ігор – потрібно проаналізувати менш масштабні проекти типу AA ігор або казуальних та гіперказуальних [2].

AA ігри – це вид відеоігор, який займає проміжне місце між інді-іграми (незалежними) та AAA іграми (великобюджетними проектами). Хоча термін "AA" може бути не так чітко визначений, він зазвичай відноситься до ігор, які мають помірний бюджет та ресурси. Ці ігри часто виробляються середніми студіями, які мають достатньо ресурсів для створення якісного продукту, але не можуть змагатися з великобюджетними студіями по масштабах виробництва та маркетингу.

Особливості AA ігор:

- бюджет і ресурси: нижче, ніж у ааа ігор, але вище, ніж у більшості інді-ігор;
- якість графіки і геймплея: зазвичай висока, але може не досягати рівня AAA проектів;
- інновації та креативність: часто AA ігри можуть дозволити собі бути більш інноваційними або експериментальними, ніж високобюджетні ігри.

Казуальні ігри – це ігри, які характеризуються простими правилами, легким доступом та короткочасними ігровими сесіями. Вони розраховані на широку аудиторію гравців, включаючи тих, хто не має значного досвіду в іграх. Казуальні ігри ідеально підходять для мобільних пристроїв або онлайн платформ, де гравці можуть швидко зайти в гру, провести в ній кілька хвилин і вийти [3].

Особливості казуальних ігор:

- простота управління: інтуїтивно зрозумілі механіки, які легко освоїти;
- короткі ігрові сесії: дозволяє грати на ходу або під час коротких перерв;
- широка доступність: часто доступні на смартфонах, планшетах та через веб-браузери.

Казуальні ігри важливі для індустрії відеоігор, оскільки вони залучають широкий круг недосвідчених користувачів і можуть служити воротами для більш глибокого ігрового досвіду.

На рисунку 1.2 наведено приклад вигляду гіперказуального платформи.

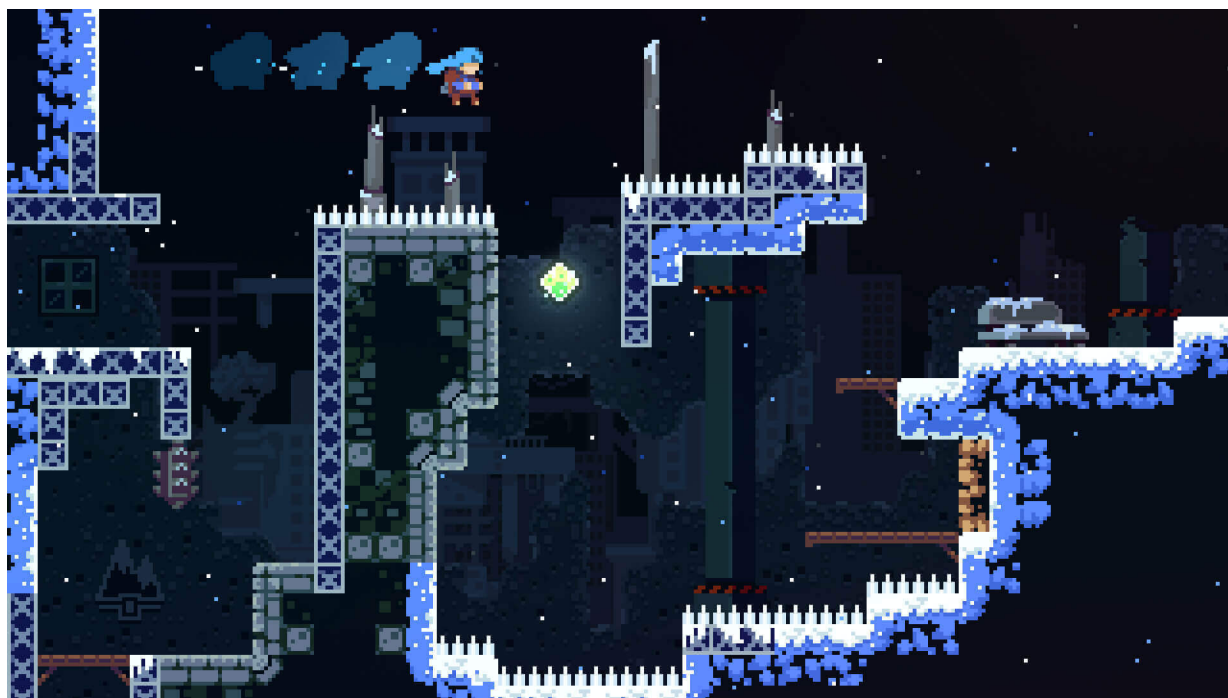


Рисунок 1.1 – Приклад казуального платформеру [2]

AAA-проекти у відео ігровій індустрії визначаються як великомасштабні, високо бюджетні ігрові проекти, які зазвичай розробляються великими командами у великих студіях або за підтримки великих видавців. Ці проекти характеризуються наступними ключовими аспектами:

AAA-ігри мають значно вищий бюджет порівняно з іншими категоріями ігор. Це включає фінансування розробки, тестування, маркетингу та дистрибуції.

Однією з найважливіших характеристик AAA-ігор є висока якість графіки та аудіовізуального дизайну. Вони часто використовують передові технології та ігрові рушії для створення реалістичних візуальних та звукових ефектів.

AAA-ігри пропонують складний і глибокий геймплей, з різноманітними сюжетними лініями, персонажами та ігровими механіками. Вони часто включають інноваційні елементи геймплею та високий рівень інтерактивності.

Великі маркетингові кампанії та глобальне розповсюдження є ключовими для AAA-ігор, оскільки вони націлені на широку аудиторію гравців.

Розробка AAA-проектів часто займає кілька років, оскільки вони вимагають детальної роботи над кожним аспектом гри, від графіки до геймплею.

Команди, які працюють над AAA-проектами, зазвичай великі та багатoproфільні, включаючи гейм дизайнерів, програмістів, художників, сценаристів, аудіо інженерів, тестувальників та інших спеціалістів.

Очікування відносно комерційного успіху AAA-ігор високі, оскільки вони повинні принести значний прибуток для компенсації великих інвестицій.

Ігри категорії AAA спонукають індустрію ставати все краще і краще в різних областях: вони прагнуть напружувати до краю роботу відеокарт, продуктивність та можливості ігрових консолей. Завдяки величезним командам розробників та обширним бюджетам вони мають і час, і сили для шліфування деталей, якими славляться великомасштабні ігри. Інді-ігри та ігри ІІІ не мають таких можливостей через обмеження бюджету або ресурсів.

Але інді-ігри, у свою чергу, спонукають ігри AAA розвивати творчіші підходи. Не можна сказати, що розробники ігор AAA не намагаються вийти за межі шаблонів, якщо надати їм шанс, але часто через обмеження батьківських компаній у них немає такої гнучкості, як у незалежних розробників.

Як тільки інді-гра доводить життєздатність будь-якого підходу, це одразу спрощує шлях до його використання для розробників ігор AAA. На рисунку 1.2 наведено приклад великого AAA проекту з відкритим світом.



Рисунок 1.2 – Приклад AAA проекту [3]

Таким чином, ігри AAA намагаються знайти межі здібностей обладнання в ігровій індустрії, а інді-ігри та ігри III так само намагаються знайти межі креативності. Для розвитку ігрової промисловості як єдиного цілого важливим є існування всіх типів ігор.

У підсумку, AAA-проекти відіграють значну роль у відео ігровій індустрії, визначаючись великим масштабом, високою якістю та значним впливом. Ці проекти зазвичай стають лідерами ринку, встановлюючи стандарти якості та інновацій у геймінгу.

Їх розробка вимагає значних інвестицій часу та ресурсів, але водночас пропонує великі можливості для творчості та технологічного прогресу в індустрії відеоігор.

Загалом успіх AAA-проекту часто визначається його здатністю вразити гравців неперевершеною графікою, захоплюючим геймплеєм та глибокою імерсією, що вимагає від розробників використання найсучасніших технологій та інноваційних підходів у створенні ігор.

1.2 Актуальність проблеми

Вибір ігрового рушія є критично важливим рішенням у процесі розробки AAA-ігор, оскільки він впливає на якість графіки, ефективність роботи команди та можливість реалізації геймплейних механік.

Сучасний ринок пропонує кілька потужних рушіїв, кожен з яких має свої унікальні особливості та обмеження. Це ставить перед розробниками складне завдання вибору найбільш підходящого рушія для їхніх конкретних потреб і цілей.

Актуальність цієї проблеми полягає у забезпеченні високої конкурентоспроможності та успішності готового продукту на ринку відеоігор, де вимоги до якості та інноваційності постійно зростають. Відповідний вибір ігрового рушія може значно підвищити шанси проекту на успіх, забезпечуючи не тільки вражаючу графіку та плавність геймплея, але й ефективність розробки та гнучкість у реалізації креативних ідей.

Розглянуто чому важливо не помилитись при виборі ігрового рушія для AAA проекту.

Вибір неподходящого ігрового рушія може призвести до різних проблем для розробників, серед яких:

- технічні обмеження – якщо рушій не підтримує потрібні функції або має обмеження щодо графіки чи фізики, це може обмежити творчість та вплинути на якість гри;
- проблеми з продуктивністю – неправильно підібраний рушій може призвести до проблем з продуктивністю, включаючи низьку частоту кадрів та довгі часи завантаження;
- складності у розробці – невідповідність інструментів рушія потребам команди може ускладнити процес розробки, збільшуючи час та витрати на реалізацію проекту;
- обмеження платформи – обмежена підтримка різних платформ може знизити аудиторію гри та обмежити можливості її розповсюдження;
- проблеми зі спільнотою та підтримкою – рушії з менш активною спільнотою або обмеженою підтримкою можуть ускладнити вирішення проблем та зменшити доступ до ресурсів та навчальних матеріалів.

Тому важливо вибирати ігровий рушій, враховуючи не тільки поточні потреби проекту, але й майбутні перспективи розвитку та підтримки.

У цій роботі ми маємо намір сформулювати обґрунтовані рекомендації щодо вибору ігрового рушія для AAA-проектів, базуючись на глибокому аналізі можливостей основних рушіїв на ринку. Це дозволить розробникам зробити інформоване рішення, яке найкраще відповідає їхнім потребам та цілям.

1.3 Постановка задачі

У ході даної кваліфікаційної роботи потрібно провести дослідження предметної області застосування та вибору ігрового рушія під великі AAA проекти які реалізують відкритий світ, дослідження популярних ігрових рушіїв, які використовуються розробниками по усьому світу та на яких реалізовані найбільш популярні та великі ігри.

Буде проведено порівняльний аналіз кожного із рушіїв, порівняно з огляду на візуальну та продуктивну складову для великих проектів.

Ця робота спрямована на виявлення найкращого вибору для великих проектів, та буде спрямований на комплексний підхід та великі команди розробників та ігрових дизайнерів.

Реалізація наукового дослідження складається з наступних етапів:

- аналіз предметної області – методів проектування 3D простору для "triple-A" проектів;
- дослідження архітектурних особливостей сучасних ігрових рушіїв, розгляд їх основних особливостей;
- порівняльний аналіз сучасних ігрових рушіїв та виведення підсумків за вибором.
- вдосконалення ігрових рушіїв при розробці AAA проектів,
- практична реалізація,
- проведення експерименту.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ РУШІВ

2.1 Дослідження рушія UNREALENGINE

UnrealEngine – це потужний ігровий рушій, який розроблений і підтримується компанією Epic Games. Він широко використовується у розробці відеоігор різних жанрів, але його можливості сягають значно далі.

Рушій підтримує більшість сучасних платформ, включаючи PC, консолі, мобільні пристрої та VR, також пропонує високоякісну візуалізацію з реалістичним освітленням, тінями та ефектами. Це одна з ключових причин, чому UnrealEngine є популярним серед розробників.

Blueprint Visual Scripting є ключовою особливістю Unreal Engine, що забезпечує потужний та гнучкий спосіб програмування, не вдаючись до традиційного текстового кодування.

Візуальний інтерфейс надає графічний інтерфейс, де логіка гри візуалізується у формі "вузлів" та "з'єднань". Кожен вузол представляє певну функцію або дію, а з'єднання вказують на потік даних або послідовність виконання.

Ця технологія робить процес розробки більш доступним для дизайнерів, художників та інших спеціалістів, які не мають глибоких знань у програмуванні.

Незважаючи на свою візуальність, Blueprint дозволяє створювати складні інтерактивні системи, що можуть включати контроль персонажів, управління ігровими механіками, інтерфейсами користувача та багато іншого [4].

Blueprint тісно інтегрована з C++, що дозволяє розробникам писати більш складний код на C++ та використовувати його разом з Blueprint.

Система сприяє швидкій ітерації та тестуванню, оскільки зміни можна вносити "на льоту" без необхідності частого компілювання проекту.

Також варто відмітити що UnrealEngine має динамічну та ресурсно-багату спільноту, яка сприяє розвитку та інноваціям серед розробників ігор і додатків. Ця спільнота підтримується через різноманітні платформи та ініціативи. Наприклад, існують численні форуми, як офіційні, так і сторонні, де розробники можуть

обмінюватися знаннями, вирішувати проблеми та ділитися досвідом. Це дозволяє швидко знаходити відповіді на запитання та розвивати професійні зв'язки.

У свою чергу, магазин UnrealEngine є центром, де можна знайти широкий спектр готових активів, плагінів, анімацій, текстур та інших необхідних компонентів, що значно спрощують процес розробки.

Також, Epic Games активно надає обширну документацію, навчальні відео та інші ресурси для освоєння UnrealEngine, включаючи безліч безкоштовних і платних курсів та вебінарів, які охоплюють як основи, так і більш просунуті аспекти роботи з рушієм.

Крім цього, регулярні конференції, мітапи та хакатони є важливою частиною спільноти, де розробники можуть зустрічатися, ділитися досвідом та вчитися один у одного. Відкритий код деяких версій UnrealEngine дозволяє більш досвідченим розробникам глибше занурюватися в механізми рушія та адаптувати його під свої специфічні потреби. Також існує Marketplace для спільноти, який сприяє обміну ресурсами та співпраці між користувачами [5].

Останнім, але не менш важливим є фінансова підтримка від Epic Games, яка час від часу пропонує гранти для підтримки інноваційних проектів на Unreal Engine, стимулюючи творчість та інноваційність у спільноті. Ця всебічна підтримка та різноманітність ресурсів роблять UnrealEngine не просто інструментом для розробки ігор, а справжнім екосистемою для розвитку та співпраці у сфері цифрової креативності.

UnrealEngine має складну архітектуру, яка складається з різних ключових структурних елементів. Ці елементи взаємодіють між собою, щоб забезпечити потужні можливості для створення ігор та інших інтерактивних досвідів.

Ключові структурні елементи Unreal Engine [6]:

- engine core – це основа рушія, яка включає в себе низькорівневі системи, такі як управління пам'яттю, системи вводу/виводу, обробка потоків та інші базові сервіси. вона забезпечує фундамент, на якому побудовані всі інші частини рушія;
- renderer – відповідає за візуалізацію сцен у рушії. Рендерер обробляє все, від

освітлення та тіней до текстур та ефектів. цей компонент є ключовим для створення візуально привабливих ігор;

- система фізики (physics engine) – відповідає за симуляцію фізичних взаємодій у ігровому світі, включаючи колізії, гравітацію, та інші фізичні ефекти;
- ігровий фреймворк (gameplay framework) – включає в себе класи та компоненти, які становлять основу ігрової логіки, такі як персонажі, актори (об'єкти у грі), контролери та ігровий світ;
- blueprint visual scripting – система візуального скриптування, яка дозволяє розробникам створювати ігрову логіку без необхідності програмування на c++. вона тісно інтегрована з ігровим фреймворком, дозволяючи легко маніпулювати об'єктами та їх поведінкою;
- інтерфейс користувача (UI engine) – складається з інструментів для створення інтерфейсу користувача, включаючи меню, панелі інструментів, вікна налаштувань тощо;
- система аудіо (audio engine) – відповідає за управління звуками у грі, включаючи музику, звукові ефекти, голосові доріжки тощо;
- інтеграція мережі (networking) – надає можливості для мережевої гри, включаючи синхронізацію стану гри між різними гравцями та серверами;
- система частинок (particle system) – використовується для створення складних ефектів, таких як вогонь, дим, іскри тощо. це важлива частина створення реалістичних та візуально привабливих ігрових сцен;
- AI та навігація (ai and navigation) – підтримка штучного інтелекту для неперсонажних об'єктів (npcs) та систем навігації, які дозволяють прс орієнтуватися в ігровому світі;
- матеріали та шейдери (materials and shaders) – дозволяють розробникам створювати складні візуальні ефекти та поверхні, використовуючи гнучку систему матеріалів та шейдерів.

На рисунку 2.1 наведено архітектурну структуру UnrealEngine, її компоненти та зв'язок між ними.

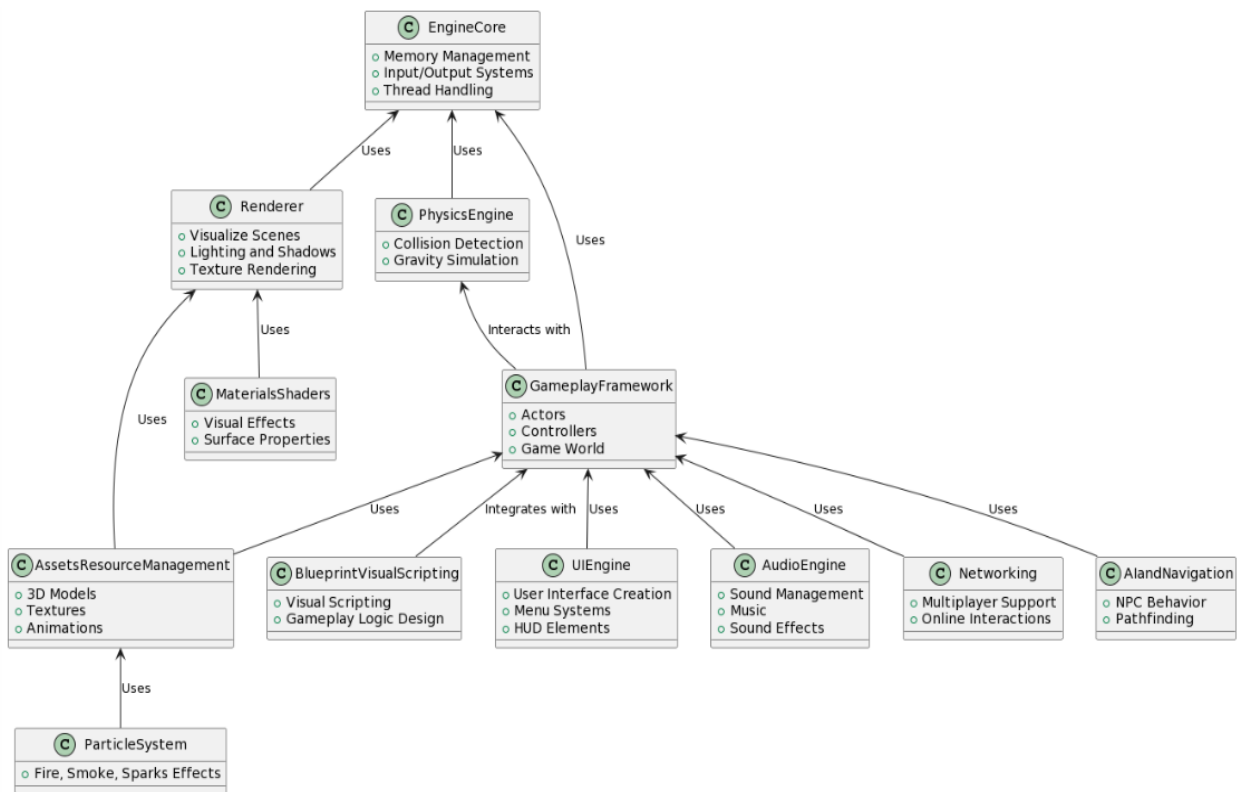


Рисунок 2.1 – Архітектурна структура Unrealengine, її компоненти та зв'язок між ними [5]

Розглянуто переваги та недоліки рушію Unrealengine [7].

Переваги Unrealengine:

- одна з ключових переваг UnrealEngine – це його здатність створювати дуже реалістичну візуалізацію та вражаючі графічні ефекти;
- система blueprint дозволяє розробникам швидко та ефективно створювати ігрову логіку без глибоких знань у програмуванні;
- рушій підтримує велику кількість платформ, включаючи pc, консолі, мобільні пристрої, та VR;
- UnrealEngine дозволяє розробникам масштабувати проекти від малих до великих, підтримуючи як інді-проекти, так і великі AAA-ігри;
- сильна підтримка спільноти та наявність багатої бібліотеки ресурсів, навчальних матеріалів і готових активів;

- еріс games регулярно оновлює UnrealEngine, додаючи нові функції та покращення.

Недоліки Unrealengine:

- для створення та запуску ігор на UnrealEngine часто потрібні потужні комп'ютери, особливо для роботи з високоякісною графікою;
- незважаючи на наявність blueprint, загальна складність і обсяг функцій рушія може бути викликом для новачків;
- для комерційних проєктів, які перевищують певний поріг доходу, еріс games вимагає відсоток від прибутку;
- проєкти, створені на UnrealEngine, можуть мати великий розмір файлів, що може бути проблемою для мобільних пристроїв або платформ з обмеженим місцем для зберігання;
- для повноцінної роботи з UnrealEngine та використання всіх його можливостей потрібні глибокі знання мови програмування c++.

2.2 Дослідження рушія Cryengine

CryEngine – це високо потужний ігровий рушій, розроблений компанією Crytek. Він відомий своїми першокласними графічними можливостями та здатністю створювати візуально вражаючі ігри, особливо в жанрах, де ключову роль відіграє реалістичність середовища. CryEngine використовується у розробці відеоігор різних жанрів і масштабів, але він особливо популярний у створенні графічно інтенсивних AAA-проєктів.

Основною сильною стороною CryEngine є його вражаючі візуальні можливості. Рушій пропонує одні з найкращих на ринку інструментів для рендерингу, включаючи реалістичне освітлення та високоякісні візуальні ефекти. Це робить його ідеальним вибором для проєктів, де графіка відіграє вирішальну роль.

Цей рушій також має вбудовану систему фізики, яка дозволяє створювати реалістичні інтерактивні середовища. Поряд з цим, рушій підтримує широкий спектр

платформ, включаючи PC, консолі та VR, забезпечуючи розробникам гнучкість у виборі цільової аудиторії [6].

Однак, рушій може бути складнішим у використанні порівняно з іншими ігровими рушіями, особливо для новачків у сфері розробки ігор. Він вимагає від розробників глибших технічних знань і розуміння складних інструментів рушія. Також важливо зазначити, що хоча CryEngine пропонує безкоштовну версію, комерційне використання рушія може вимагати ліцензування та відрахувань від доходів.

Спільнота, хоч і активна, може бути не такою великою чи ресурсно багатою, як у випадку UnrealEngine або Unity. Проте Crytek регулярно оновлює рушій, додаючи нові функції та покращення, та надає документацію та підтримку для розробників.

Загалом, CryEngine є потужним вибором для розробки ігор, особливо коли мова йде про створення візуально вражаючих та технічно вимогливих AAA-проектів.

Далі варто розглянути головну особливість рушія – шейдинг CryENGINE.

CryENGINE 3 використовує PBR, який імітує взаємодію світла з фізичними матеріалами в реалістичний спосіб. Це дозволяє досягти більш достовірних та узгоджених візуальних результатів у різних освітлювальних умовах.

PBR використовує матеріали з реалістичними властивостями, такими як відблиск, шорсткість та металічність, щоб точно відтворити, як світло взаємодіє з поверхнями.

Варто розглянути розширені шейдери окремо:

- шейдери освітлення: cryengine 3 включає широкий спектр шейдерів для обробки різних типів освітлення, включаючи пряме сонячне світло, розсіяне світло та штучне освітлення;

- шейдери матеріалів: рушій також пропонує різноманітні шейдери для створення реалістичних матеріалів, які можуть імітувати різні текстури та властивості, такі як метал, скло, тканина тощо.

CryENGINE 3 використовує різні методи оптимізації для підвищення продуктивності, включаючи LOD (Level of Detail), який зменшує деталізацію об'єктів на далекій відстані, та інші техніки для зменшення обчислювального навантаження.

Також він включає шейдери та техніки для створення спеціальних ефектів, таких як відображення, тіні, частинкові системи, водні ефекти та інше.

Шейдинг у CryENGINE 3 спрямований на створення реалістичного та візуально привабливого відображення сцен і об'єктів. Використання PBR, розширених шейдерів та оптимізаційних технік дозволяє рушію ефективно візуалізувати складні сцени з високим рівнем деталізації, забезпечуючи при цьому високу продуктивність при рендерингу.

На рисунку 2.2 наведено архітектурну структуру cryengine, його компоненти та зв'язок між ними.

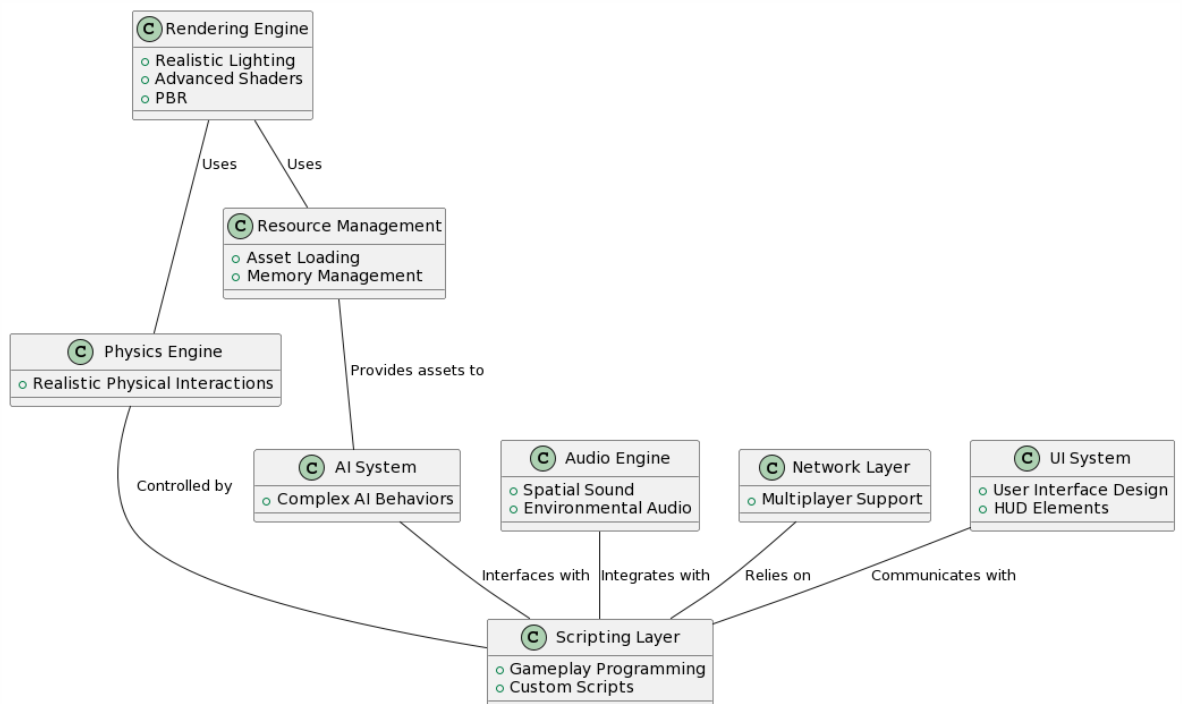


Рисунок 2.2 – Архітектурна структура cryengine, її компоненти та зв'язок між ними

Розглянуто кожен з елементів діаграми:

- rendering engine (двигун рендерингу) – це основа cryengine, яка відповідає

за візуалізацію сцен у грі, воно включає в себе передові технології для реалістичного освітлення, тіней та шейдерів. також підтримує physically based rendering (pbr) для створення реалістичних матеріалів та текстур;

- physics engine (двигун фізики) – забезпечує реалістичну симуляцію фізичних взаємодій у ігровому світі, включаючи рухи об'єктів, зіткнення, гравітацію та інші фізичні ефекти;

- AI system (система штучного інтелекту) – використовується для створення розумної поведінки неперсонажних персонажів (npc) у грі, це включає шляхопошук, взаємодію з гравцем та автономні рішення;

- audio engine (аудіо двигун) – відповідає за управління звуками у грі, включаючи музику, звукові ефекти, просторовий звук та аудіо взаємодію з середовищем;

- network layer (мережевий шар) – надає функціональність для мультиплеєрних ігор та інших мережевих взаємодій, включаючи синхронізацію даних між різними гравцями;

- resource management (управління ресурсами) – керує завантаженням, вивантаженням та оптимізацією ресурсів гри, таких як текстур, моделі, звуки тощо;

- UI system (система інтерфейсу користувача) – забезпечує інструменти та компоненти для створення інтерфейсів користувача в іграх, включаючи меню, індикатори стану, інформаційні панелі тощо;

- scripting layer (шар скриптів) – дозволяє розробникам ігор створювати ігрову логіку та механіки через скрипти, що полегшує розробку та дозволяє швидко вносити зміни.

2.3 Дослідження рушія Unity

Unity – це універсальний ігровий рушій, який розроблений та підтримується компанією Unity Technologies. Він використовується для розробки відеоігор різних жанрів та інтерактивних додатків, включаючи ігри для мобільних пристроїв, ПК, консолей та VR/AR. Його універсальність і доступність роблять Unity популярним

вибором серед інди-розробників та невеликих студій, хоча рушій також використовується для створення деяких великих комерційних проєктів.

Рушій підтримує більшість сучасних платформ, що робить його ідеальним вибором для розробки кросплатформних ігор. Рушій пропонує гнучкі інструменти для створення візуалізації з реалістичним освітленням, тінями та ефектами, хоча він може бути сприйнятий як менш потужний у порівнянні з UnrealEngine у контексті створення AAA-ігор.

Однією з ключових особливостей Unity є його система скриптів на основі C#. Ця система є більш доступною та простішою у порівнянні з традиційним текстовим кодуванням, роблячи рушій привабливим для новачків та недосвідчених розробників. Візуальний інтерфейс Unity дозволяє логіку гри візуалізувати у формі компонентів та інспекторів, забезпечуючи інтуїтивне та гнучке середовище розробки [8].

Unity має активну та ресурсно-багату спільноту, яка сприяє розвитку та інноваціям серед розробників. Існують численні форуми, офіційні та сторонні, де розробники можуть обмінюватися знаннями та вирішувати проблеми. Unity Asset Store є центром, де розробники можуть знайти широкий спектр готових активів, плагінів, анімацій та текстур, що спрощує процес розробки.

Компанія Unity Technologies активно надає обширну документацію, навчальні відео та інші ресурси, включаючи безліч безкоштовних і платних курсів та вебінарів, охоплюючи як основи, так і більш продвинуті аспекти роботи з рушієм.

Переваги Unity

- доступність та легкість використання, unity відомий своєю простотою і доступністю, особливо для новачків та інди-розробників;
- підтримка різних платформ, широка підтримка платформ, від мобільних до консолей та vr;
- активна спільнота та великий вибір ресурсів у unity asset store;
- використання c# робить скриптинг доступнішим для широкого кола розробників.

Недоліки Unity:

- хоча unity пропонує хороші графічні можливості, він може бути менш потужним для створення ігор з високоякісною графікою порівняно з UnrealEngine;
- розробники можуть стикатися з викликами у оптимізації продуктивності для деяких платформ;
- unity пропонує різні моделі ліцензування, що може вимагати підписок для доступу до деяких розширених функцій.

Unity є міцним рішенням для широкого спектру проектів розробки ігор, особливо для тих, хто шукає баланс між функціональністю, доступністю та багатоплатформністю.

Далі розглянуто особливості моделювання текстур за допомогою Substance 3D Plugin.

Substance 3D Plugin від Unity – це плагін, який інтегрує можливості створення і редагування текстур Substance в Unity. Substance є продуктом від компанії Adobe, який включає в себе набір інструментів для створення та редагування високоякісних PBR (Physically-Based Rendering) матеріалів та текстур. Ця інтеграція надає розробникам Unity можливість створювати, імпортувати та використовувати динамічні та високо деталізовані текстури безпосередньо у своїх проектах.

Технологія дозволяє розробникам створювати складні матеріали з контролем над різними аспектами, такими як колір, шорсткість, механічність та інші фізичні властивості.

Завдяки Substance, розробники можуть використовувати обширну бібліотеку готових матеріалів або створювати свої власні унікальні текстури. Плагін забезпечує плавну інтеграцію з Unity, дозволяючи імпортувати і використовувати Substance матеріали безпосередньо в редакторі Unity.

Можливість динамічно змінювати текстури в реальному часі в Unity, що може бути корисним для різних сценаріїв геймплея та інтерактивних досвідів. Також Substance 3D Plugin дозволяє оптимізувати матеріали для різних платформ, забезпечуючи високу якість та продуктивність.

Переваги використання Substance в Unity

- використання substance матеріалів може значно підвищити візуальну якість ігор і додатків, створених у unity;
- гнучкість у дизайні надає розробникам більше контролю над візуальними аспектами їх проектів;
- інтеграція з unity зменшує час, необхідний для створення та імпортування складних матеріалів.

Substance 3D Plugin може бути використаний у широкому спектрі проектів від інді-ігор до великих AAA-проектів, де потрібні деталізовані та реалістичні текстури та матеріали.

У додатку Б наведено архітектурну структуру unity, її компоненти та зв'язок між ними.

Проаналізуємо ці елементи більш детально.

Детально розглянуто кожен з ключових елементів Unity Engine, що наведені у PlantUML діаграмі:

- Unity Engine – це основний фреймворк, який забезпечує основну функціональність для розробки ігор та інтерактивних додатків. Він включає в себе інструменти для рендерингу, фізики, аудіо, скриптів, мережевих функцій та багато іншого;
- Graphics Rendering відповідає за візуалізацію графіки в іграх та додатках. Включає підтримку шейдерів, глобального освітлення в реальному часі та систем частинок, що дозволяє створювати деталізовані та візуально привабливі сцени;
- Physics Engine забезпечує реалістичну симуляцію фізики, включаючи зіткнення, динаміку твердих тіл і багато іншого. Це ключовий компонент для створення ігор, де важлива реалістична взаємодія об'єктів;
- Audio Engine управляє всіма аспектами аудіо у грі, включаючи просторовий звук, мікшування, а також створення та відтворення звукових ефектів і музики;
- Networking надає можливості для мережевої взаємодії у мультиплеєрних іграх, включаючи підтримку мережевих протоколів, синхронізацію стану гри між

гравцями та інші мережеві функції;

- Scripting System – система скриптів на базі мови програмування C# дозволяє розробникам створювати ігрову логіку, контролювати поведінку об'єктів та реалізовувати індивідуальні механіки гри;

- Asset Store – це онлайн-магазин, де розробники можуть знайти та придбати готові до використання активи, інструменти та розширення, які можуть допомогти в розробці ігор та прискорити процес створення контенту;

- Editor Extensions – Unity дозволяє розширювати свій редактор за допомогою налаштовуваних інструментів та імпортерів активів, що дозволяє розробникам адаптувати робочий процес під свої унікальні потреби.

3 ВДОСКОНАЛЕННЯ МЕТОДУ ПРОЕКТУВАННЯ 3D-ПРОСТОРУ

3.1 Вдосконалення ігрових рушіїв при розробці AAA проектів

Вдосконалення методу проектування 3D-простору буде проведено відносно AAA проектів, використовуючи розглянуті раніше ігрові рушії. Це має сенс більше ніж до більш простих сценаріїв використання, таких як мобільні ігри чи платформери або інді ігри із обмеженими можливостями деталізації.

Вдосконалення ігрових рушіїв з точки зору їх придатності для AAA проектів має важливе значення, оскільки AAA ігри висувають особливо високі вимоги до якості, продуктивності та функціональності. Ці ігри, зазвичай, розробляються великими студіями з значним бюджетом і спрямовані на досягнення високого рівня графіки, глибокої геймплейної механіки та високоякісного сюжету.

AAA ігри вимагають висококласного візуального та аудіо оформлення. Рушії повинні мати здатність ефективно обробляти деталізовану графіку, складні анімації та високоякісний звук.

Подібні проекти часто включають складні ігрові механіки, інтерактивні елементи та розгалужені сюжетні лінії. Рушії повинен надавати гнучкі інструменти для реалізації цих вимог.

З огляду на великі масштаби та складність AAA проектів, ефективна оптимізація і продуктивність є ключовими. Це включає ефективне управління пам'яттю, швидке завантаження рівнів та плавність гри на різних платформах.

Також AAA ігри часто розробляються для різних платформ, включаючи консолі, ПК та іноді мобільні пристрої. Рушії повинен підтримувати легку адаптацію проектів під різні платформи.

Підтримка потужних інструментів для розробки, включаючи візуальне скриптування, розширені інструменти для анімації, інтеграція зі стороннім ПЗ та ефективна технічна підтримка, є критично важливими для AAA проектів.

Варто відмітити що наявність міцної спільноти та багатий набір ресурсів, включаючи готові активи, допомагає в процесі розробки, зокрема у скороченні часу та витрат.

Після аналізу переваг та недоліків кожної системи можна зробити висновок що UnrealEngine є доволі оптимальним вибором для розробки AAA-ігор, головним чином завдяки його передовим графічним можливостям і потужному рендерингу. Завдяки технології Nanite, UnrealEngine дозволяє створювати вражаючі, високодеталізовані сцени, які забезпечують неперевершений рівень реалізму та візуальної якості, особливо коли йдеться про освітлення та тіні завдяки підтримці ray tracing.

Це робить його ідеальним для створення графічно інтенсивних ігрових середовищ, що є ключовим для AAA-проектів.

Система Blueprint Visual Scripting в UnrealEngine є ще однією важливою особливістю, яка дозволяє розробникам швидко реалізувати складні ігрові механіки без глибоких знань у програмуванні, що сприяє більш ефективному процесу розробки. Окрім того, UnrealEngine забезпечує високий рівень оптимізації для різних платформ, що дозволяє створювати ігри, які працюють плавно та ефективно на різноманітних пристроях.

Додатково, сильна підтримка спільноти та наявність багатих ресурсів, включаючи Unreal Marketplace, надають розробникам широкий спектр активів та інструментів для їх проектів. Всі ці фактори разом створюють потужний набір інструментів та можливостей, роблячи UnrealEngine переважним вибором для розробників, які прагнуть до створення візуально вражаючих, технологічно складних ігор з високою продуктивністю.

Також необхідно відмітити що з початку варто розглянути візуальну складову результатів розробки, це найважливіша деталь простору для AAA проектів, адже користувачі частіше за все орієнтуються на візуальну складову проекту.

На рисунку 3.1 наведено візуальну різницю якості об'єкту між її реалізацією на UnrealEngine та unity [10].



Рисунок 3.1 – Порівняння якості текстур на різних рушіях

На рисунку представлена картинка справа на UnrealEngine а зліва на unity, тож чітко можна побачити більшу деталізацію на UnrealEngine та роботи технології Lumen для більш об'ємного та чіткого освітлення.

Тому далі необхідно детальніше розглянути, які технології дозволяють UnrealEngine бути краще за інші рушії.

3.2 Переваги UnrealEngine із Nanite Geometry

Спочатку проаналізуємо Nanite Geometry від UnrealEngine.

Nanite Geometry є однією з революційних функцій Unreal Engine, запровадженою в UnrealEngine5. Ця технологія представляє собою новий спосіб обробки та візуалізації геометрії у високій деталізації, що відкриває нові горизонти для графічної якості та реалізму у відеоіграх та інших цифрових додатках.

Nanite використовує мікро полігональний рендеринг, що дозволяє гейм-дизайнерам використовувати надзвичайно деталізовані моделі без значного впливу на продуктивність. Це може включати моделі з мільйонами або навіть мільярдами полігонів.

Nanite інтелектуально вирішує, які полігони потрібно візуалізувати в залежності від їх видимості та важливості, оптимізуючи продуктивність та зменшуючи навантаження на апаратне забезпечення.

Завдяки можливості використання високо деталізованих моделей та текстур, Nanite відкриває нові можливості для створення неймовірно реалістичних сцен та об'єктів.

Технологія дозволяє розробникам використовувати високо деталізовані ассети без необхідності їх ручного оптимізування, що спрощує робочий процес і скорочує час розробки.

Далі треба відмітити важливу фішку та механізм оптимізації графічних об'єктів у UnrealEngine – це Fallback Mesh.

Fallback Mesh у контексті Nanite Geometry в UnrealEngine має важливе значення. Оскільки Nanite Geometry — це технологія для високодеталізованої візуалізації 3D моделей, Fallback Mesh служить як "запасний" або "альтернативний" меш, який використовується в певних ситуаціях. Ось ключові аспекти цієї концепції:

Fallback Mesh — це більш проста, низькополігональна версія деталізованої 3D моделі. Цей меш використовується, коли основний високодеталізований Nanite Mesh не може бути візуалізований ефективно через обмеження апаратного забезпечення або інші обставини.

У контексті Nanite Geometry, Fallback Mesh використовується як запасний варіант, коли візуалізація високодеталізованого мешу не є оптимальною або можливою. Наприклад, це може відбуватися при перегляді сцени з великої відстані, де деталізація високополігонального мешу не є важливою, або на менш потужних системах, де висока деталізація може надмірно навантажувати обчислювальні ресурси.

Переваги використання Fallback Mesh:

- оптимізація продуктивності: використання менш деталізованого fallback mesh дозволяє зберегти високу продуктивність, особливо в сценах з великою кількістю об'єктів або на менш потужних системах;

– гнучкість в рендерингу: наявність fallback mesh забезпечує гнучкість, дозволяючи рушію вибрати найбільш ефективний спосіб візуалізації в залежності від обставин;

– підтримка широкого діапазону апаратних специфікацій: завдяки цій функції, ігри та додатки на базі UnrealEngine можуть бути оптимізовані для роботи на різноманітних пристроях, від високопродуктивних ПК до менш потужних платформ.

Багатьом частинам UnrealEngine потрібен доступ до традиційного буфера вершин, який надають традиційно відтворені сітки. Коли Nanite увімкнено для статичної сітки, він генерує приблизне представлення високодеталізованої сітки, яка доступна та використовується там, де недоступні дані Nanite. Резервна сітка – це згенерована сітка, яка використовується, коли візуалізація Nanite не підтримується.

На рисунку 3.2 наведено порівняння деталізованої сітки та сітки без Nanite.



Рисунок 3.2 – Оптимізація шейдерних ресурсів

На рисунку можна чітко побачити економію графічних ресурсів через Fallback Mesh.

Таким чином, Fallback Mesh в контексті Nanite Geometry відіграє важливу роль у забезпеченні гнучкості, оптимізації продуктивності та широкої підтримки апаратних платформ, доповнюючи високі графічні можливості Nanite.

3.3 Трасування променів у UnrealEngine

Ray Tracing у UnrealEngine є однією з ключових інновацій, яка значно підвищує реалізм і візуальну якість ігор та інших цифрових досвідів, створених за допомогою цього рушія.

Ray Tracing (променеве трасування) — це техніка візуалізації, яка імітує шляхи світла, як вони взаємодіють з об'єктами у віртуальному середовищі. Це дозволяє досягти надзвичайно реалістичних візуальних ефектів, включаючи відображення, тіні, розсіювання світла та інші складні освітлювальні феномени.

Розглянуто детальніше особливості Ray Tracing у UnrealEngine:

- реалістичні відображення та тіні, ray tracing у UnrealEngine забезпечує значно більш реалістичні відображення та тіні в порівнянні з традиційними методами растрового рендерингу – це особливо помітно у сценах зі складним освітленням та відблисками;
- поліпшена якість освітлення – технологія дозволяє більш точно імітувати фізичні властивості світла, що робить освітлення в сцені значно реалістичнішим;
- Ray tracing підтримує глобальне освітлення в реальному часі, дозволяючи світловим променям відбиватися від поверхонь і розсіюватися по сцені, що створює більш природню атмосферу.

Підвищення реалізму та іммерсивність. Використання Ray Tracing робить ігрові сцени та віртуальні середовища набагато більш занурюючими та реалістичними, значно підвищуючи якість візуального досвіду.

Приклад візуальної відмінності віддзеркалень поверхонь за допомогою трасування світла наведено на рисунку 3.3.



Рисунок 3.3 – Віддзеркалення від поверхонь за допомогою трасування світла

Ліворуч зображення без трасування променів. На середньому зображенні показано першу реалізацію розробником трасування променів два роки тому. Крайне праворуч зображення з реалізацією трасування променів 2021 року.

Проблема з трасуванням променів полягає в тому, що воно дуже важке для гри, яка його використовує, особливо враховуючи, що для отримання кращих результатів, ніж коли його вимкнено, необхідна хороша реалізація трасування променів, що збільшує ймовірність того, що більше променів має бути відстежено (з більшою кількістю відмов), а отже, ймовірніше, що це буде більш обтяжливим.

Але технології розвиваються доволі швидко, та сучасні графічні процесори здатні видавати прийнятну продуктивність для цієї технології.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1 Основні функціональні методи

Після того, як було обґрунтовано вибір рушіїв із достатнім потенціалом до великих проектів, слід обрати їх спільну функціональну рису для проведення дослідження по ній, та виявлення найефективнішого рушія та підходу до вирішення проблеми.

Почати варто із загального огляду функціональності.

Працювати з моделями та середовищем у графічних та ігрових двигунах включає кілька ключових аспектів, які допомагають створити багатогранний і реалістичний цифровий світ. Ось більш широкий список головних елементів, з якими зазвичай працюють розробники:

- імпорт і експорт 3d моделей;
- моделювання;
- текстурування;
- матеріали та шейдери;
- освітлення;
- анімація;
- система lod (level of detail);
- фізика та взаємодія;
- пейзаж і терен;
- контроль якості і оптимізація.

Кожен з цих елементів відіграє важливу роль у процесі створення цифрових світів, і розуміння їх взаємодії та впливу на кінцевий продукт є ключовим для успішної розробки ігор та інших візуальних проектів.

У дослідженні зосереджено на порівнянні ключових функціональних елементів роботи з моделями у трьох популярних ігрових рушіях: UnrealEngine, CryEngine та Unity. Особливу увагу ми приділимо системам Level of Detail (LOD), які мають велике значення для оптимізації візуальної продуктивності ігор.

Виявлено, що всі три двигуни підтримують широкий спектр форматів для імпорту та експорту 3D моделей, забезпечуючи гнучкість при роботі з різними 3D редакторами. UnrealEngine та Unity відомі своєю сумісністю з форматом FBX, який є стандартом у індустрії, тоді як CryEngine добре інтегрується зі своїми власними інструментами, такими як CryTIF плагіни.

У контексті роботи з матеріалами та текстурами, UnrealEngine вирізняється своєю потужною системою на базі вузлів, що дозволяє створювати складні матеріали з високим рівнем контролю над візуальними ефектами. Unity також надає гнучкі можливості через Shader Graph, дозволяючи користувачам візуально створювати шейдери. CryEngine забезпечує реалістичний рендеринг завдяки своїм деталізованим налаштуванням матеріалів.

LOD є критично важливим аспектом у всіх трьох двигунах, оскільки це ключ до оптимізації продуктивності без жертвування якістю графіки. У UnrealEngine LOD реалізований через автоматичну та ручну систему, дозволяючи детально контролювати перехід між різними рівнями деталізації. Unity використовує компоненти LOD Group для управління різними рівнями деталізації залежно від відстані до камери. CryEngine також дозволяє ручне налаштування LOD, але з більшим акцентом на настройку візуальних параметрів для досягнення максимальної реалістичності [11].

Підводячи підсумок, варто зупинитись на дослідженні LOD функціоналу, адже він надає наступні можливості для великого проекту:

- підвищення продуктивності – lod дозволяє значно покращити продуктивність додатків шляхом зниження кількості полігонів, що мають бути оброблені і відображені у віддаленій частині сцени. це критично для забезпечення плавної графіки та високої частоти кадрів, особливо в іграх та додатках віртуальної реальності, де швидкодія є ключовим фактором;

- підвищення якості графіки – дослідження lod дозволяє розробникам детальніше зрозуміти, як створювати більш реалістичні і деталізовані сцени без негативного впливу на продуктивність. застосування різних рівнів деталізації

допомагає зберегти візуальну якість при мінімальному навантаженні на ресурси;

- масштабованість – lod сприяє масштабованості проектів, адже дозволяє іграм та додаткам ефективно працювати на різноманітному обладнанні, від високопродуктивних настільних комп'ютерів до мобільних пристроїв. це робить ігри більш доступними для широкої аудиторії, оскільки користувачі з менш потужними системами також можуть насолоджуватися високоякісною графікою;

- економія ресурсів розробки – використання lod може значно скоротити час і ресурси, потрібні для створення деталізованих моделей, оскільки дозволяє розробникам створювати менш деталізовані версії об'єктів для віддалених сцен, тим самим скорочуючи час, потрібний для ручної оптимізації;

- глибина дослідження – дослідження lod механізмів відкриває двері для інновацій у способах реалізації і використання цих технік, спонукаючи до вдосконалення існуючих методів і розробки нових підходів, які можуть ще більше покращити візуальний досвід і продуктивність.

LOD, або Level of Detail (рівень деталізації) – це техніка в комп'ютерній графіці, яка використовується для підвищення продуктивності шляхом зменшення складності відображення 3D об'єктів на більшій відстані від спостерігача. Основна ідея полягає в тому, щоб використовувати більш деталізовані моделі для близької візуалізації, а менш деталізовані – для далекої. Це дозволяє зберегти ресурси обчислювальної системи, знижуючи навантаження на графічний процесор і підвищуючи загальну продуктивність.

Рівень LOD – це сітка яка визначає рівень деталізації, що візуалізується ігровими рушіями для геометрії GameObject. Коли GameObject використовує LOD, рушій відображає відповідний рівень LOD для цього GameObject залежно від відстані GameObject від камери.

Кожен рівень LOD існує в окремому GameObject, кожен з яких має сітчастий рендерер-компонент, який відображає цей рівень LOD. На рисунку 4.1 нижче показано, як змінюються рівні LOD залежно від відстані від камери.

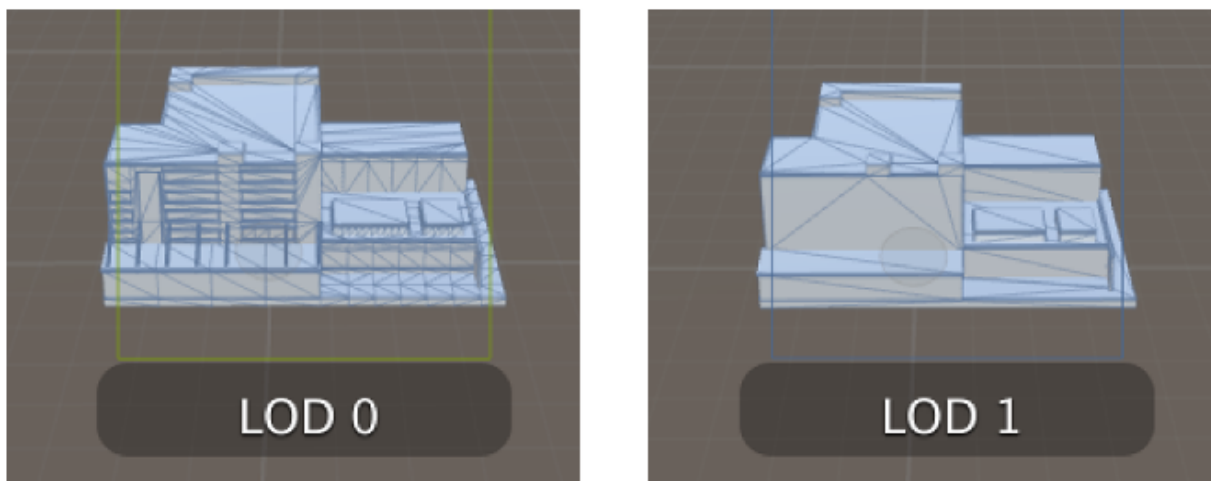


Рисунок 4.1 – Зміна LOD залежно від відстані

Далі будуть розглянуті способи створення лодів у обраних рушіях: unity та UnrealEngine.

В UnrealEngine та Unity існують різні способи створення LOD (Level of Detail), що дозволяє розробникам ефективно управляти деталізацією моделей для оптимізації продуктивності. Розглянуто основні методи, які використовуються в цих двох популярних ігрових движунах.

Unreal Engine:

- автоматичне генерування lod – UnrealEngine має засоби для автоматичного створення декількох lod рівнів для мешів. це робиться через редактор мешів, де користувачі можуть налаштувати кількість lod рівнів та відсоток зменшення полігонів для кожного наступного lod. система автоматично обраховує оптимальні налаштування lod на основі вказаних критеріїв;

- ручне створення lod – розробники можуть самостійно створювати та імпортувати власні lod версії моделей. це зазвичай використовується для особливо важливих асетів, де деталізація і контроль якості є критичними. у редакторі мешів також є можливість ручно налаштовувати перехід між різними lod рівнями за допомогою параметра screen size.

Аналогічна ситуація і у Unity:

- LOD Group Component – Unity використовує компонент LOD Group, який

дозволяє легко налаштувати різні LOD рівні для об'єкта. Компонент автоматично управляє переключенням між LOD рівнями на основі відстані до камери. Користувачі можуть вручну додавати або видаляти LOD рівні і встановлювати відсотки переключення для кожного LOD;

– автоматичні LOD Генератори – у Unity також можна використовувати сторонні плагіни або розширення, такі як Simplygon або Mesh Baker, для автоматизації процесу створення LOD. Ці інструменти надають додаткові можливості для більш складних проектів [12-14].

Обидва двигуни надають інтуїтивні інструменти для роботи з LOD, дозволяючи розробникам ефективно оптимізувати свої ігри під різні апаратні платформи. Вибір між автоматичним та ручним підходами залежить від конкретних вимог проекту та ресурсів, доступних для оптимізації моделей.

Але нами буде розглянуто виключно автоматичний спосіб генерації що більше задіяти програмну частину двигунів, адже вони будуть автоматично оновлювати та вираховувати відстань до об'єкту та перебудовувати його на ходу.

4.2 Вхідні дані до тестування

Щоб провести коректне тестування, було встановлено обидва рушії на одну систему, кожен з них має свій , які беруть свою частину ресурсів системи, далі ці поправки буде внесено у кінцеву формулу.

Для вимірювання продуктивності було взято також одну модель в одному розмірі трикутників поверхні та висот (див. рис. 4.3), сама модель зображена на рисунку 4.2. Було прийнято рішення взяти велику за обсягом модель для більшого скейлінгу різниці між рушіями.

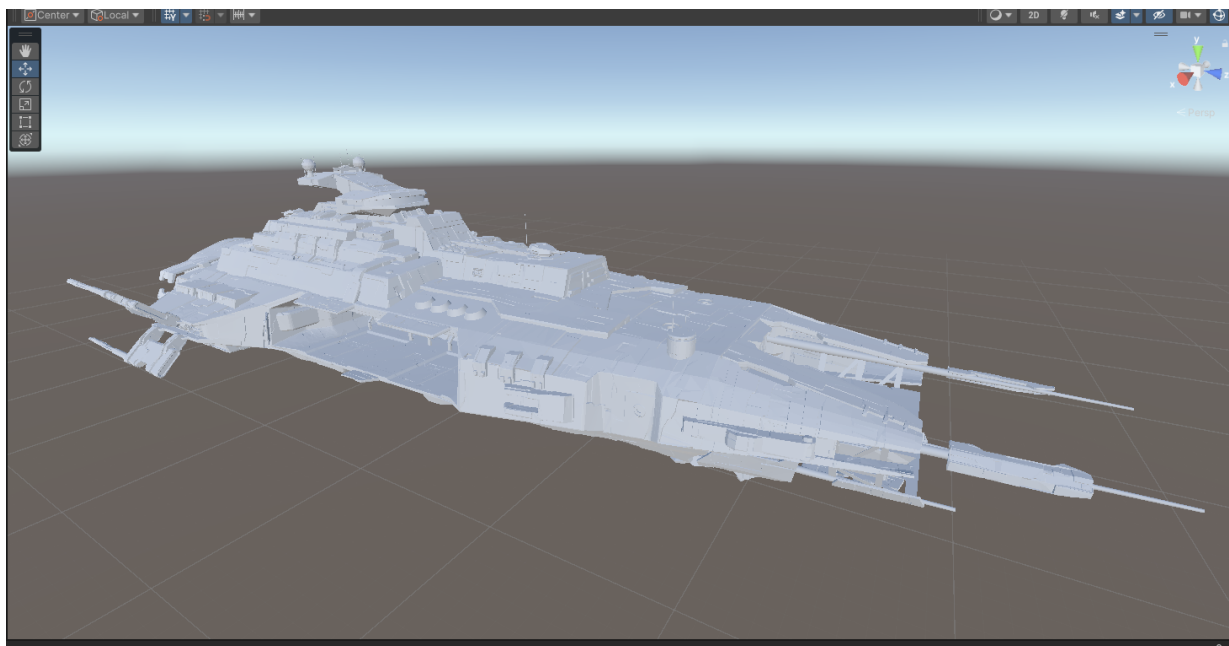


Рисунок 4.2 – Статична тестова модель

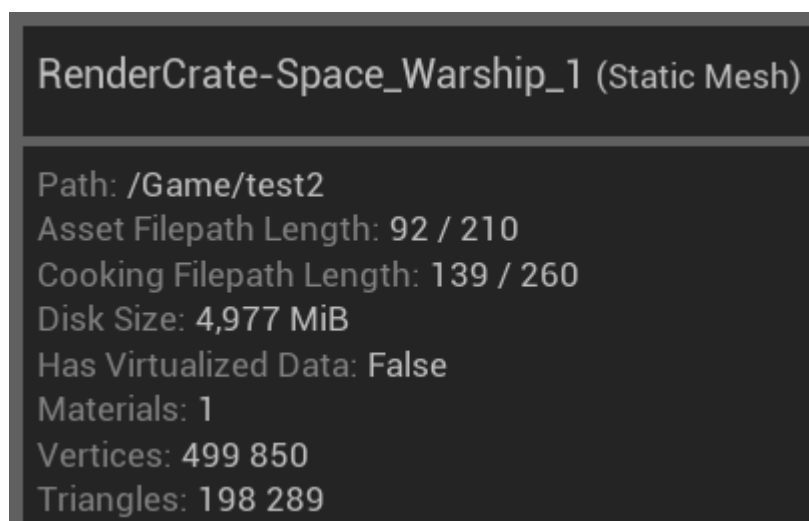


Рисунок 4.3 – Розміри моделі

Отже, як можна побачити: вершин – майже 500 000, трикутників – 198 289.

Також важливо не завантажувати початковий простір, тож було прибрано майже все оточення включно із освітленням.

Далі необхідно співставити системи координат системи LOD у кожному рушії. Для початку варто розібрати що таке LOD-groups. LOD-groups, або групи рівнів деталізації, є важливим інструментом у 3D моделюванні та ігрових двигунах для управління продуктивністю сцен. Вони дозволяють автоматично адаптувати

деталізацію 3D об'єктів залежно від їх відстані від камери або іншого виду взаємодії з користувачем.

Отже на рисунках 4.4 та 4.5 приведено співставлення відстаней та груп лодів у обох рушіях.

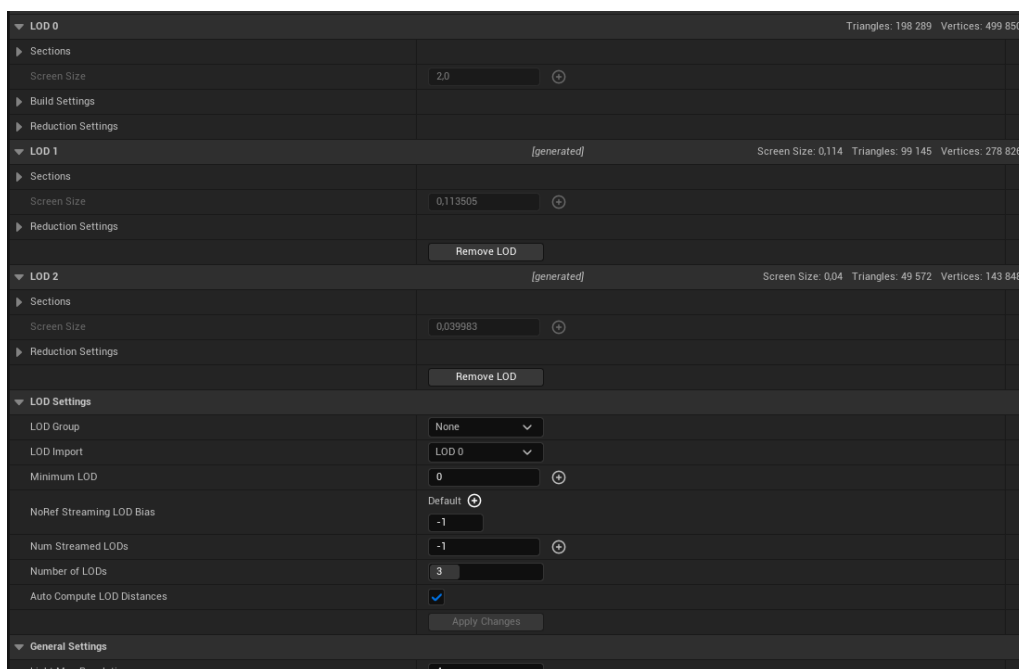


Рисунок 4.4 – Unreal engine LODs

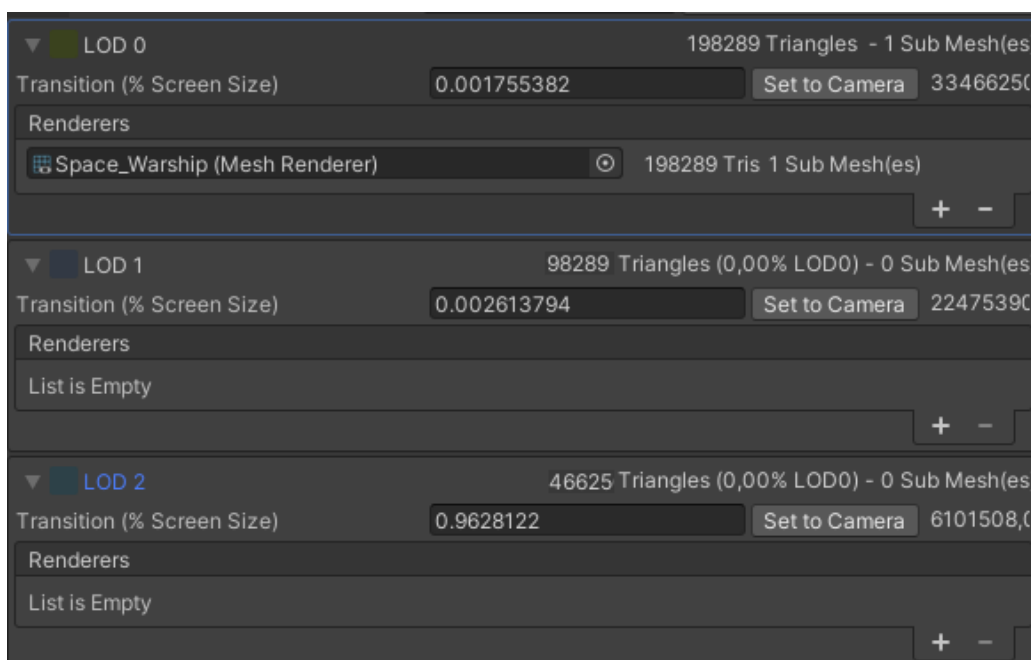


Рисунок 4.5 – Unity LODs

Як видно з наведеного прикладу є наступні розміри лодів у трикутниках значення зображені у таблиці 4.1.

Таблиця 4.1 – LODs size

LODs	Кількість трикутників у моделі залежно від 3x відстаней
LOD1	198 289 трикутників
LOD2	99 145 трикутників
LOD3	49 572

Слід відмітити, що unity трохи округлив значення, також вони були прораховані відносно відстані мануально у самому рушії[15].

Варто розуміти, що для кожної ітерацію лоду необхідно на обох рушіях встановити однакові величини, та однакову роздільну здатність.

Надалі необхідно розглянути інструмент для виміру продуктивності систем. Та було обрано використати FPS Monitor. Приклад інтерфейсу наведено на рисунку 4.6.

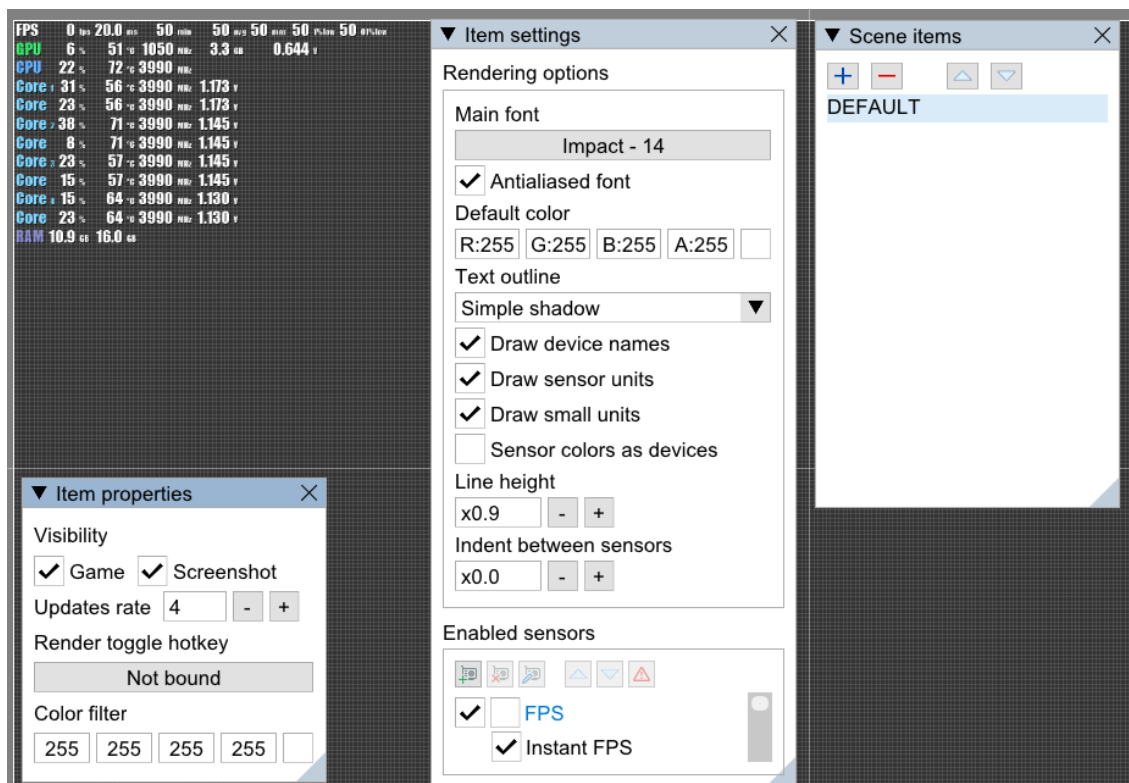


Рисунок 4.6 – FPS Monitor

Це необхідний інструмент, адже продуктивність роботи функціоналу буде виміряна за кадрами на секунду (FPS), це показник який демонструє кількість кадрів які рендерить відеочіп на секунду часу. Чим стабільніша система, тим більше кадрів на секунду.

FPS Monitor (Frames Per Second Monitor) – це програмне забезпечення, яке використовується для відстеження продуктивності вашого комп'ютера під час гри або іншої активності, яка інтенсивно використовує ресурси системи. Цей інструмент дозволяє користувачам бачити в реальному часі такі показники, як кількість кадрів за секунду (FPS), використання процесора (CPU), використання відеокарти (GPU), температури компонентів[12], швидкість вентиляторів та інші системні параметри. Основні аспекти, які варто знати про FPS Monitor, наведені нижче:

- відображення fps: найбільш часто використовувана функція – це моніторинг fps, який є критичним для геймерів, оскільки високий fps забезпечує плавність зображення в іграх;
- відстеження ресурсів системи: перегляд статистики використання cpi, гри, оперативної пам'яті та дискового простору;
- діагностика апаратних збоїв: визначення перегріву компонентів або інших потенційних проблем із апаратним забезпеченням завдяки моніторингу температури та швидкостей вентиляторів;
- налаштування інтерфейсу: користувачі можуть налаштовувати, де і як інформація відображається на екрані під час ігор або інших додатків.

5 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

5.1 Результати бенчмаркінгу

У кваліфікаційній роботі проведено запуски ігор, в яких, як зазначалось раніше, присутня лише одна модель, було проведено віддалення від об'єкту(моделі) задля того, щоб спрацювали встановлені LODs, на кожній ітерації зменшення полігонів (трикутників та вершин) було знято показники кадрів на секунду.

У додатку В на рисунках В.1-В.6 можна побачити результати кадрів в секунду для кожної із 3х відстаней (ітерацій LOD), на рисунку В.1-В.3 для UnrealEngine, на рисунку В.4-В.6 для Unity. Важливо відмітити, що було спочатку встановлено еталонне значення кадрів на секунду для кожного рушія, відповідно при статичній моделі ми отримали кількість кадрів яка стала відправною точкою, надалі буде вираховано значення відхилення на кожній ітерації LOD, та порівняно між двома рушіями. Еталонні значення наведено на рисунках В.1, В.2 відповідно.

5.2 Аналіз результатів

Для візуального сприйняття результатів експерименту результати наведено у таблиці 5.2.

Таблиця 5.2 – Результати бенчмаркінгу

LOD ітерації	Рушії	
	UnrealEngine	unity
LOD-1	40 fps	267 fps
LOD-2	45 fps	290 fps
LOD-3	46 fps	300 fps

Для визначення, який із двох рушіїв (UnrealEngine чи Unity) впорався краще з LOD ітераціями не лише на основі абсолютних значень FPS, але і враховуючи тенденції змін FPS, можна використовувати різні статистичні підходи. Один із

ефективних методів – це обчислення коефіцієнта зростання FPS від однієї LOD ітерації до наступної. Такий підхід дозволить врахувати динаміку змін продуктивності між ітераціями.

Щоб визначити, який рушій показав кращу тенденцію до покращення продуктивності з кожною новою LOD ітерацією, можна використовувати наступну формулу:

$$x = \frac{u_n - u_{n(n-1)}}{u_{n(n-1)}} \times 100 \% \quad (5.1)$$

Де u_n – кадри за секунду на ітерації n та $u_{n(n-1)}$ – кадри за секунду на попередній ітерації.

Виходячи із розрахунків – ми маємо періудне зростання у UnrealEngine у $12.5\% + 2.22\% = 14.7\%$. У unity – $8.6\% + 3.4\% = 12\%$.

На гістограмі, що представлена на рисунку 5.1, наведено динаміку зростання кадрів на кожній ітерації обох рушіїв.

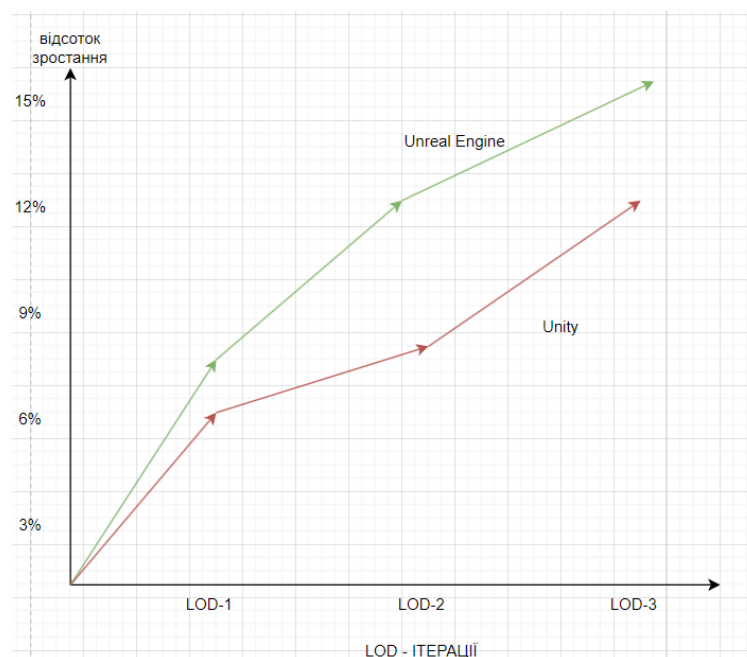


Рисунок 5.1 – Гістограма зростання кадрів в процентах

За результатами можна побачити, що при бенчмаркінгу продуктивності результати, як і очікувалось, дуже близькі відносно моделі такого розміру. Але загалом маємо більш ефективну систему оптимізації текстур у UnrealEngine.

ВИСНОВКИ

У кваліфікаційній роботі проведено всебічний аналіз основних ігрових рушіїв – UnrealEngine, CryEngine, і Unity з акцентом на їхню придатність для створення AAA-проектів. Розглянуто ключові характеристики та функціональності кожного рушія, включаючи їх графічні можливості, підтримку скриптів, оптимізацію продуктивності та підтримку спільноти.

На основі детального порівняльного аналізу, обґрунтовано висновок, що UnrealEngine є найкращим вибором для розробки великомасштабних AAA-ігор.

UnrealEngine вирізняється своїми вражаючими графічними можливостями, зокрема завдяки технології Nanite, що дозволяє створювати високодеталізовані сцени та середовища з неперевершеним рівнем реалізму. Система Blueprint Visual Scripting значно спрощує процес розробки, дозволяючи розробникам швидко реалізувати складні ігрові механіки без глибоких знань у програмуванні.

Окрім того, UnrealEngine забезпечує високу оптимізацію для різних платформ та має потужну підтримку спільноти та багатий набір ресурсів. Все це робить UnrealEngine ідеальним рішенням для розробки AAA-ігор, що вимагають високого рівня графіки, глибокої геймплейної механіки та високоякісного сюжету.

У результаті проведено дослідження, здебільшого шейдерної генерації та загальної якості текстур у різних ігрових рушіях відносно AAA проектів, надалі необхідно буде провести аналіз оптимізації та загальної швидкодії для отримання остаточного результату.

Також проведено порівняння ключових функціональних елементів (LOD), за результатами яких було встановлено перевагу підходу в UnrealEngine.

Однак, слід звернути увагу на те, що CryEngine повинен випустити оновлення, яке принесе нові можливості, тоді рекомендації та результати дослідження можуть змінитися.

Кваліфікаційна робота пройшла апробацію на Міжнародному молодіжному форумі, 2024 р. [24].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аналіз предметної області, URL: <https://www.indeed.com/career-advice/finding-a-job/what-is-web-architecture> (дата звернення: 01.04.2024).
2. Аналіз казуальних проєктів, URL: <https://store.epicgames.com/news/what-makes-a-aaa-game-a-aaa-game> (дата звернення: 01.04.2024).
3. Аналіз AAA проєктів, URL: <https://store.epicgames.com/ru/news/what-makes-a-aaa-game-a-aaa-game> (дата звернення: 01.04.2024).
4. Best AAA Games - Everything You Should Know About Triple A, URL: <https://www.g2a.com/news/features/best-aaa-games/> (дата звернення: 01.04.2024).
5. What are AAA Games, URL: <https://www.arm.com/glossary/aaa-games> (дата звернення: 20.04.2024).
6. What makes a AAA game a AAA game? , URL: <https://store.epicgames.com/news/what-makes-a-aaa-game-a-aaa-game> (дата звернення: 05.04.2024).
7. Indie, AAA, and AA Games: A Comparison, URL: <https://www.gameopedia.com/category/aaa-games/> (дата звернення: 05.04.2024).
8. Find the apps and games you're looking for, URL: <https://www.meta.com/experiences/section/1440155176607031/> (дата звернення: 20.04.2024).
9. WHAT IS AAA GAME DEVELOPMENT, URL: <https://ejaw.net/what-is-aaa-game-development/> (дата звернення: 05.04.2024).
10. Stoyan Y. G., Smelyakov S. V. An approach to the problems of routing optimization in the regions of intricate shape. *Information Processing Letters*. 1981. Vol. 13, no. 1. P. 39–43. URL: [https://doi.org/10.1016/0020-0190\(81\)90148-4](https://doi.org/10.1016/0020-0190(81)90148-4)(дата звернення: 05.04.2024).
11. Falatiuk H., Shirokopetleva M., Dudar Z. Investigation of Architecture and Technology Stack for e-Archive System. *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv,

Ukraine, 8–11 October 2019. 2019. URL: <https://doi.org/10.1109/picst47496.2019.9061407> (дата звернення: 05.04.2024).

12. Smelyakov S. V., Stoyan Y. G. Modelling of the space of paths in problems of constructing optimal trajectories. *Computational Mathematics and Mathematical Physics*. 1993. Vol. 23, no. 1. P. 50–55. URL: [https://doi.org/10.1016/s0041-5553\(83\)80009-3](https://doi.org/10.1016/s0041-5553(83)80009-3) (дата звернення: 05.04.2024).

13. Аналіз unreal engine: <https://www.quora.com/What-is-the-new-Lumen-technology-in-Unreal-Engine-5> (дата звернення: 05.04.2024).

14. Аналіз unreal engine, URL: <https://www.quora.com/What-is-the-new-Lumen-technology-in-Unreal-Engine-5> (дата звернення: 05.04.2024).

15. Аналіз cry engine, URL: <https://substance3d.adobe.com/plugins/substance-in-unreal-engine/> (дата звернення: 15.04.2024).

16. Аналіз unity, URL: <https://substance3d.adobe.com/plugins/substance-in-unreal-engine/> (дата звернення: 15.04.2024).

17. Візуальне порівняння рушіїв, URL: <https://www.ixbt.com/live/games/igra-mechty-svoimi-rukami-subektivnyy-obzor-sovremennyh-dvizhkov.html> (дата звернення: 15.04.2024).

18. Дослідження роботи LOD Unrel Engine, URL: <https://docs.unrealengine.com/4.26/en-US/WorkingWithContent/Types/StaticMeshes/> (дата звернення: 15.04.2024).

19. Дослідження роботи LOD Unity, URL: <https://docs.unity3d.com/530/Manual/class-LODGroup.html> (дата звернення: 21.04.2024).

20. Аналіз та визначення LODs size, URL: <https://dev.epicgames.com/documentation/> (дата звернення: 21.04.2024).

21. Аналіз FPS Monitor, URL: <https://fpsmon.com/en/> (дата звернення: 21.04.2024).

22. GAMEG. Cover, Editorial Board, Acknowledgement and Table of Contents. *Gameology and Multimedia Expert*. 2024. Vol.1, no.1. URL: <https://doi.org/10.29103/game.v1i1.14573> (date of access: 21.04.2024).

23. Аналіз методів сегментації зображень автомобільних реєстраційних номерів\ Володін, Д. О.,Афанасьєва, І. В., Видавництво: ХНУРЕ, 2019р., URL: <https://openarchive.nure.ua/handle/document/18415> (дата звернення: 21.04.2024).

24. Бронов І. В. Дослідження методів проектування 3D простору для «Triple-A проектів» // Матеріали 28 Міжнародного молодіжного форуму «Радіоелектроніка і молодь у ХХІ столітті». Харків: ХНУРЕ, 2024. С. 183-184.