

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Платформа розробки інтерактивних 3D-середовищ для відеоігор на
основі Unreal Engine 4»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ *Марк КРУШЕНИЦЬКИЙ*
(підпис) (Ім'я, Прізвище здобувача)

Виконав:
здобувач вищої освіти
група ІСДМ-61

Марк КРУШЕНИЦЬКИЙ

Керівник:
науковий ступінь,
вчене звання

Оксана ТКАЛЕНКО
к.т.н., доцент

Рецензент:
науковий ступінь,
вчене звання

Ім'я, ПРИЗВИЩЕ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти – «Магістр»

Спеціальність - 126 - Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Сторчак К.П.

“ ____ ” _____ 2024 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Крушеніцькому Марку Олександрович

(прізвище, ім'я, по батькові)

1.Тема роботи: «Платформа розробки інтерактивних 3D-середовищ для відеоігор на основі Unreal Engine 4»

Керівник роботи: Ткаленко Оксана Миколаївна, к.т.н., доц.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «15» жовтня 2024 року
№320.

2.Строк подання студентом роботи: 27 грудня 2024 року.

3. Вихідні данні до кваліфікаційної роботи:

Рушії для розробки 3D-додатків: Unreal Engine 4, Unity, CryEngine.

Інструменти для 3D-моделювання: Blender, Autodesk Maya, ZBrush.

Технічна література, присвячена інтерактивним 3D-середовищам і розробці відеоігор.

4. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

4.1. Аналіз ігрових рушіїв для створення інтерактивних 3D-середовищ.

4.2. Застосування ігрових рушіїв у популярних проектах.

4.3. Розробка інтерактивного 3D-середовища на основі Unreal Engine 4.

5. Перелік ілюстративного матеріалу: презентація

Огляд платформ для розробки інтерактивних 3D-середовищ.

Порівняння можливостей Unreal Engine 4, Unity і CryEngine.

Приклади популярних проєктів, створених на Unreal Engine 4.

Інструменти для 3D-моделювання: Blender

Етапи розробки інтерактивного 3D-середовища на основі Unreal Engine 4.

6. Дата видачі завдання 01 вересня 2024 року

Календарний план

№/п	Назва етапів магістерської роботи	Срок виконання роботи	Примітка
1	Аналіз навної науково-технічної літератури	15.10 – 01.11.24	Виконано
2	Аналіз платформи Unreal Engine 4 та її можливостей	02.11 – 09.11.24	Виконано
3	Початок роботи з платформою Unreal Engine 4 (аналіз, створення)	10.11 – 16.11.24	Виконано
4	Аналіз ринку інтерактивних 3D-середовищ	17.11 – 20.11.24	
5	Дослідження методології створення інтерактивних середовищ	21.11 – 10.12.24	Виконано
6	Оформлення роботи: вступ, висновки, реферат	11.12 – 20.12.24	Виконано
7	Розробка демонстраційних матеріалів	21.12 – 25.12.24	Виконано

Здобувач вищої освіти

(підпис)

Марк КРУШЕНІЦЬКИЙ
(Ім'я, ПРІЗВИЩЕ)

Керівник роботи
кваліфікаційної роботи

(підпис)

Оксана ТКАЛЕНКО
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 77 с., 23 рис., 30 джерел.

Об'єкт дослідження – процес розробки інтерактивних 3D-середовищ для відеоігор.

Предмет дослідження – платформа Unreal Engine 4 та її функціональні можливості для створення 3D-контенту.

Мета роботи – дослідження та оцінка інструментів і технологій Unreal Engine 4 для забезпечення розробки високоякісного інтерактивного 3D-середовища.

Методи дослідження – методи моделювання, методи текстурювання, методи анімації, методи оптимізації, емпіричні методи тестування.

У даній магістрській роботі проведено дослідження функцій та інструментів платформи Unreal Engine 4, спрямованих на розробку інтерактивних 3D-середовищ для відеоігор. У роботі проаналізовано процес створення 3D-моделей, їх анімацію та текстурювання для реалізації реалістичних візуальних ефектів. Окрема увага приділена системі Blueprints, яка дозволяє створювати сценарії та інтерактивні механіки без використання коду, що полегшує процес розробки.

КЛЮЧОВІ СЛОВА: UNREAL ENGINE, 3D-СЕРЕДОВИЩЕ, ІНТЕРАКТИВНІСТЬ, ВІДЕОІГРИ, МОДЕЛЮВАННЯ, АНІМАЦІЯ, ТЕКСТУРУВАННЯ, РЕНДЕРИНГ, СИСТЕМА BLUEPRINTS, ОПТИМІЗАЦІЯ.

ABSTRACT

The text section of the qualification paper for obtaining the educational degree: 77 pages, 23 figures, 30 sources.

Object of the research - the process of developing interactive 3D environments for video games.

Subject of the research – the Unreal Engine 4 platform and its functional capabilities for creating 3D content.

Purpose of the work – to study and evaluate the tools and technologies of Unreal Engine 4 to ensure the development of a high-quality interactive 3D environment.

Research methods – modeling methods, texturing methods, animation methods, optimization methods, empirical testing methods.

Summary of the work: In this master's thesis, research was conducted on the functions and tools of the Unreal Engine 4 platform aimed at developing interactive 3D environments for video games. The study analyzed the process of creating 3D models, their animation, and texturing to implement realistic visual effects. Special attention was paid to the Blueprints system, which allows the creation of scripts and interactive mechanics without using code, thereby facilitating the development process.

KEYWORDS: UNREAL ENGINE, 3D ENVIRONMENT, INTERACTIVITY, VIDEO GAMES, MODELING, ANIMATION, TEXTURING, RENDERING, BLUEPRINTS SYSTEM, OPTIMIZATION.

ЗМІСТ

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ РОЗРОБКИ 3D-СЕРЕДОВИЩ У ВІДЕОІГРАХ	10
1.1 Огляд сучасних 3D-движків: порівняння та функціональні можливості	10
1.2 Технологічні особливості Unreal Engine 4	13
1.3 Використання інших програмних інструментів: Blender, Substance Painter, ZBrush.....	19
1.4 Аналіз ринку інтерактивних 3D-середовищ для відеоігор.....	21
РОЗДІЛ 2. МЕТОДОЛОГІЯ СТВОРЕННЯ ІНТЕРАКТИВНИХ 3D-СЕРЕДОВИЩ НА ОСНОВІ UNREAL ENGINE 4.....	28
2.1 Основи проєктування і розробки 3D-середовищ	28
2.2 Робота з системою Blueprints у Unreal Engine 4	34
2.3 Використання фізичних симуляцій і реалістичного рендерингу	41
2.4 Підхід до оптимізації 3D-середовищ.....	45
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ ГРИ	53
3.1 Діаграма класів.....	53
3.2 Опис функціоналу гри.....	54
3.3 Механіка пересування персонажа.....	59
3.4 Алгоритм пошуку шляху	63
3.5 Створення мобів.....	65
3.6 Розробка механік взаємодії.....	67
3.7 Інструкція користувача гри.....	70
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	Error! Bookmark not defined.
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	77

ВСТУП

В сучасному світі цифрових технологій значно зросла популярність інтерактивних 3D-середовищ, які широко використовуються в таких галузях, як розробка відеоігор, архітектурне проєктування, віртуальна реальність та навчальні симуляції. Завдяки можливості створювати реалістичні і захоплюючі середовища, які взаємодіють з користувачем, 3D-технології дозволяють значно підвищити залучення користувачів і досягти нових рівнів реалістичності та інтерактивності.

Одним із провідних інструментів для розробки 3D-середовищ є платформа Unreal Engine 4, яка надає потужний функціонал для створення графічно насичених та інтерактивних середовищ. Цей движок відрізняється високою якістю рендерингу, системою Blueprints для програмування без кодування, можливістю інтеграції з іншими програмами для моделювання, такими як Blender, і підтримкою складних фізичних симуляцій. Unreal Engine 4 активно використовується в ігровій індустрії, але завдяки своїм можливостям також підходить для різноманітних проєктів у сфері візуалізації, симуляцій та навчання.

Об'єктом дослідження цієї роботи є процес розробки інтерактивних 3D-середовищ для відеоігор, а *предметом дослідження* – технології та інструменти платформи Unreal Engine 4, що забезпечують ефективне створення таких середовищ.

Метою даної роботи є дослідження функціональних можливостей Unreal Engine 4 та інших інструментів, зокрема Blender, для створення високоякісних інтерактивних 3D-середовищ, які могли б використовуватися у відеоіграх і симуляціях. Зокрема, буде досліджено процес моделювання, текстурування, анімації, оптимізації та застосування інтерактивних механік для забезпечення реалістичного досвіду користувача.

Для досягнення мети роботи було поставлено такі *завдання*:

1. Проаналізувати сучасні рішення для розробки 3D-середовищ і визначити можливості Unreal Engine 4.

2. Дослідити функціональні можливості системи Blueprints, фізичних симуляцій та рендерингу у Unreal Engine 4.
3. Ознайомитися з процесом моделювання та текстурювання в Blender та розробити інтеграцію з Unreal Engine 4.
4. Розробити прототип інтерактивного 3D-середовища, що демонструє можливості Unreal Engine 4 у створенні реалістичних середовищ.
5. Провести тестування та оптимізацію розробленого середовища для забезпечення стабільної продуктивності на різних платформах.

Методи дослідження включають огляд наукової та технічної літератури, методи моделювання, текстурювання та анімації, а також тестування та оптимізацію результатів. Виконання завдань потребує комплексного підходу, що включає практичне застосування знань з комп'ютерної графіки, програмування та системного проектування.

Наукова новизна роботи полягає у комплексному підході до інтеграції інструментів Unreal Engine 4 та Blender для створення інтерактивного 3D-середовища з високим рівнем деталізації та інтерактивності. Результати роботи можуть бути використані для подальших досліджень у сфері 3D-візуалізації та розробки програмного забезпечення для відеоігор.

Апробація результатів магістерської роботи:

Крушеніцький М. О. «Технології створення інтерактивних 3D-середовищ у відеоіграх на Unreal Engine 4». Тези доповіді на II Всеукраїнській науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу». – Київ, 18 листопада 2024 року, С.322-323.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ РОЗРОБКИ 3D-СЕРЕДОВИЩ У ВІДЕОІГРАХ

1.1 Огляд сучасних 3D-движків: порівняння та функціональні можливості

Розробка 3D-середовищ для відеоігор є складним процесом, який включає низку етапів, таких як моделювання, текстурювання, анімація, освітлення та програмування інтерактивних елементів. Існує кілька популярних інструментів і движків для створення таких середовищ, серед яких виділяються Unreal Engine, Unity, CryEngine та власні розробки великих компаній. У цьому розділі буде розглянуто кожен із цих інструментів, їх особливості, переваги та обмеження.

Unreal Engine, розроблений компанією Epic Games, є одним з найпопулярніших і найпотужніших інструментів для розробки 3D-ігор. Движок забезпечує високу якість графіки завдяки своїй здатності до рендерингу, системам анімації, фізики та спеціальній системі Blueprints, яка дозволяє створювати ігрову логіку без написання коду. Це дозволяє значно спростити процес розробки та зробити його доступнішим для дизайнерів та художників, які не володіють програмуванням.

Unreal Engine також підтримує інтеграцію з різними програмами для 3D-моделювання, такими як Blender і Autodesk Maya, що полегшує імпорт і роботу з моделями. Основні переваги Unreal Engine включають:

- висока якість графіки: можливість рендерингу реалістичних текстур, тіней і освітлення;
- система Blueprints: візуальне програмування ігрової логіки;
- розширені можливості фізики: симуляція фізичних взаємодій та динаміка частинок;
- широкі можливості оптимізації: інструменти для зниження навантаження на апаратне забезпечення, що дозволяє використовувати движок на різних платформах.

Unity — це ще один популярний движок, який часто використовується для створення як 2D, так і 3D-ігор. Його сильні сторони включають кросплатформенність, зручність у використанні, потужну спільноту та великий набір додаткових інструментів, доступних у магазині Unity Asset Store. Незважаючи на те, що якість рендерингу в Unity нижча, ніж в Unreal Engine, Unity залишається привабливим вибором для розробників, які орієнтовані на мобільні та інді-проекти.

Переваги Unity включають наступне:

- Кросплатформенність: підтримка розробки для мобільних пристроїв, консолей, ПК та веб-браузерів;
- unity Asset Store: доступ до тисяч готових ресурсів для швидкої розробки
- можливості для роботи в 2D і 3D: Unity часто використовується для створення 2D-ігор завдяки потужним інструментам для роботи з 2D-графікою.

CryEngine, розроблений компанією Crytek, відомий своїми можливостями у рендерингу реалістичної графіки, зокрема зображенням води, світла та тіней. CryEngine також пропонує ряд інструментів для створення великих відкритих світів, що робить його ідеальним вибором для створення ігор з реалістичним довкіллям.

До переваг CryEngine можна віднести наступне:

- реалістичне освітлення та рендеринг: просунуті алгоритми для створення реалістичних ефектів освітлення;
- інструменти для великих відкритих світів: підтримка великих карт з детальним опрацюванням довкілля;
- потужна фізика: CryEngine включає потужні фізичні моделі, що забезпечують реалістичну динаміку.

Недоліком CryEngine є досить високі вимоги до апаратного забезпечення, що може обмежувати використання цього движка для мобільних платформ та деяких ПК.

Багато великих ігрових компаній розробляють власні движки для специфічних потреб. Такі інструменти оптимізовані під конкретні вимоги ігор і

пропонують більш високий рівень кастомізації. Прикладами таких движків є RE Engine від Capcom для серії Resident Evil або Frostbite від Electronic Arts, який використовується в серіях Battlefield та FIFA.

Переваги власних рішень включають:

- оптимізація під конкретні проекти: висока ефективність та оптимізація для певних типів ігор;
- кастомізація: можливість глибокої налаштування движка відповідно до потреб гри.

Проте, використання власних рішень вимагає значних ресурсів та фахівців, що обмежує їх доступність для невеликих компаній та індивідуальних розробників.

Кожен із розглянутих движків має свої особливості та сфери застосування. Unreal Engine є ідеальним вибором для високоякісних 3D-проектів завдяки своїм графічним можливостям і простоті використання через систему Blueprints. Unity, хоча і менш потужний у плані графіки, є більш гнучким для кросплатформної розробки та має великий вибір готових ресурсів.

CryEngine виділяється реалістичністю графіки та можливістю створення великих відкритих світів, проте вимагає значних ресурсів. Власні рішення підходять для великих проектів, але обмежені у доступі через високу вартість розробки та потребу в спеціалізованих знаннях.

На основі аналізу можна зробити висновок, що Unreal Engine є оптимальним вибором для розробки високоякісних інтерактивних 3D-середовищ завдяки своїм графічним можливостям, інструментам оптимізації та гнучкій системі програмування. Unity є привабливим вибором для менш ресурсомістких проектів або мобільних платформ, а CryEngine ідеально підходить для створення реалістичних відкритих світів.

1.2 Технологічні особливості Unreal Engine 4

Unreal Engine 4 (UE4) — це один із найпотужніших і найпоширеніших 3D-двигків, який забезпечує високу якість графіки, масштабованість, багатоплатформеність та широкий набір функцій для створення інтерактивних середовищ. Його технологічні можливості дозволяють використовувати UE4 як для створення відеоігор, так і для віртуальної реальності, архітектурної візуалізації, кіновиробництва та інших індустрій.

Однією з головних переваг Unreal Engine 4 є його сучасна система рендерингу, яка підтримує фізично коректне освітлення (Physically Based Rendering, PBR). PBR дозволяє досягти реалістичного відображення матеріалів, таких як метал, дерево, скло, тканина. Крім того, UE4 підтримує такі технології, як:

- **Dynamic Global Illumination (SVOGI):** забезпечує динамічне глобальне освітлення в реальному часі, створюючи природне й реалістичне освітлення.
- **Screen Space Reflections (SSR):** дозволяє відображати рефлексії на поверхнях без значного навантаження на апаратну частину.
- **High Dynamic Range (HDR):** підвищує якість кольорів та освітлення.

Однією з ключових особливостей Unreal Engine 4 (UE4) є його передова система рендерингу, яка забезпечує високу якість візуалізації. Ця система інтегрує сучасні технології для створення фізично коректного освітлення (Physically Based Rendering, PBR), яке дозволяє досягти реалістичного відображення матеріалів. Завдяки PBR моделювання таких матеріалів, як метал, дерево, скло або тканина, виглядає максимально правдоподібно та природно, враховуючи всі нюанси їхньої взаємодії зі світлом.

PBR є основою для досягнення реалістичності в 3D-середовищах. Цей підхід базується на точних фізичних моделях поведінки світла, що робить поверхні об'єктів природними незалежно від умов освітлення. Наприклад, металеві поверхні правильно відображають навколишнє середовище, зберігаючи

унікальні характеристики глянцевої й блискучості, а дерев'яні поверхні демонструють матовість і легке розсіювання світла.

Особливості PBR у UE4 дозволяють досягти:

- Чіткої адаптації текстур до будь-яких змін у джерелах світла.
- Ефективного поєднання статичного і динамічного освітлення.
- Максимального контролю над параметрами матеріалів, такими як відбивна здатність (reflectivity) та мікрошорсткість (micro-roughness).

Це особливо важливо для ігор, де увага до деталей, таких як текстури зброї, будівель або персонажів, впливає на загальне сприйняття гравцем графіки.

UE4 підтримує набір унікальних технологій, які додають глибини, реалістичності та деталізації до рендерингу.

Система динамічного глобального освітлення є однією з головних інновацій UE4. Вона дозволяє в реальному часі створювати природне освітлення сцени, яке враховує взаємодію світла з об'єктами. Наприклад, світло, що відбивається від яскравої поверхні, освітлює сусідні об'єкти, додаючи глибини до сцени. Завдяки SVOGI:

- Динамічні зміни в освітленні — наприклад, захід сонця чи освітлення від лампи, яку гравець може перемістити — автоматично відображаються на всіх об'єктах.
- Створюється реалістичний ефект розсіювання світла у великих просторах, таких як печери чи інтер'єри будівель.

Для передачі рефлексій без значного навантаження на апаратне забезпечення UE4 використовує технологію Screen Space Reflections. Вона дозволяє точно відображати об'єкти на блискучих поверхнях, таких як дзеркала, вода або метал. При цьому SSR працює швидко, оскільки обчислює відбиття лише для об'єктів, що знаходяться у полі зору камери. Це забезпечує:

- Чітке й деталізоване відображення для ключових об'єктів сцени.
- Мінімізацію витрат ресурсів порівняно з методами, які потребують розрахунків для всієї сцени.

HDR забезпечує більш широкий динамічний діапазон освітлення, що дозволяє передавати насичені кольори та природну взаємодію світла з об'єктами. Використання HDR особливо помітне в сценах, де є різкі контрасти між темними й яскравими ділянками, наприклад, у кадрах із сонцем на горизонті або у нічних сценах із джерелами світла. Завдяки HDR:

- Світлі об'єкти не здаються перенасиченими, а темні не втрачають деталей.
- Досягається плавність переходів між освітленими й затіненими зонами.

Сучасна система рендерингу UE4 є ідеальною для розробки ігор будь-яких жанрів — від реалістичних симуляторів до барвистих фентезійних світів. Наприклад:

- У шутерах, таких як Fortnite, динамічне освітлення й рефлексії забезпечують високий рівень деталізації під час бойових сцен.
- У пригодницьких іграх, як-от Hellblade: Senua's Sacrifice, освітлення й текстури створюють атмосферу, яка передає емоційний стан персонажа.

Крім ігор, Unreal Engine 4 знаходить застосування у фільмах і віртуальній реальності. Високоякісний рендеринг дозволяє створювати сцени з реалістичною глибиною та деталізацією, що розширює можливості використання технології в інших сферах.

Важливою складовою є оптимізація рендерингу для збереження продуктивності без втрати якості. UE4 надає розробникам інструменти для балансування ресурсів. Наприклад:

- Зменшення рівня деталізації для далеких об'єктів.
- Використання спрощених моделей у сценах із великою кількістю об'єктів.
- Розподіл ресурсів для освітлення, що враховує потреби конкретних платформ, як-от ПК чи мобільні пристрої.

Високоякісний рендеринг у Unreal Engine 4 — це поєднання передових технологій і гнучких інструментів, які дозволяють створювати реалістичні,

деталізовані та технічно досконалі 3D-сцени. Завдяки PBR, SVOGI, SSR та HDR UE4 дає змогу досягати вражаючої графіки навіть у складних ігрових проектах. Ця система є незамінним інструментом для розробників, які прагнуть створити якісні інтерактивні світи з глибоким зануренням.

Unreal Engine 4 пропонує широкий спектр інструментів для спрощення процесу створення 3D-середовищ:

1. **Blueprints Visual Scripting:** система візуального скриптингу дозволяє розробляти складну логіку без написання коду. Це знижує вхідний бар'єр для новачків та прискорює процес розробки.

2. **Material Editor:** інструмент для створення матеріалів із використанням вузлової структури. Він підтримує створення як простих текстур, так і складних шейдерів.

3. **Cascade і Niagara:** системи частинок для створення спецефектів, таких як дим, вогонь, вода, вибухи.

Unreal Engine 4 (UE4) є не лише потужним інструментом для рендерингу, а й надає розробникам широкий набір інструментів, що значно спрощують процес створення 3D-середовищ та інтерактивних проектів. Ці інструменти розроблені для того, щоб охопити всі етапи роботи над проектом — від початкового моделювання до фінального тестування готової гри. Їх інтеграція в межах однієї екосистеми дозволяє працювати з високою продуктивністю, зменшуючи необхідність використання сторонніх рішень.

Одним із найбільш популярних інструментів UE4 є *Blueprints Visual Scripting*. Це система візуального програмування, яка дозволяє створювати складну логіку без необхідності писати код вручну. Завдяки графічному інтерфейсу, розробники можуть з'єднувати вузли (nodes), які представляють різні функції та події. Такий підхід має декілька переваг: він знижує вхідний бар'єр для новачків, які не володіють знаннями мов програмування, і водночас пришвидшує процес створення навіть для досвідчених спеціалістів. Наприклад, за допомогою Blueprints можна створити систему керування персонажем, обробку подій взаємодії з об'єктами або навіть складні фізичні симуляції. Це також чудовий

інструмент для швидкого прототипування, що дозволяє тестувати ідеї без значних витрат часу.

Матеріали і текстури грають ключову роль у створенні реалістичних 3D-сцен, і Unreal Engine 4 має для цього потужний *Material Editor*. Цей інструмент працює на основі вузлової структури, що дозволяє створювати як прості, так і складні матеріали, використовуючи інтуїтивний графічний інтерфейс. У *Material Editor* розробники можуть легко налаштовувати властивості, такі як відбивання, прозорість, розсіювання світла, та додавати карти нормалей для створення глибини на плоских текстурах. Матеріали можуть бути адаптивними: наприклад, змінювати свої властивості залежно від умов освітлення, кута огляду або взаємодії з іншими об'єктами. Результат можна в реальному часі переглядати у вікні попереднього перегляду, що значно спрощує процес розробки.

Для створення реалістичних спецефектів Unreal Engine пропонує системи частинок — *Cascade* та *Niagara*. *Cascade* — це класична система для генерації ефектів, таких як дим, вогонь, іскри, вода чи вибухи. Вона дозволяє задавати траєкторію руху частинок, змінювати їх розмір, прозорість і колір залежно від часу. *Niagara* є більш сучасним і функціональним аналогом, який надає розширені можливості, зокрема глибоку інтеграцію з іншими системами UE4. *Niagara* дає змогу створювати не лише візуально приголомшливі ефекти, але й складні симуляції, які реагують на фізичні закони, такі як гравітація, вітер чи взаємодія з об'єктами. Наприклад, у грі можна реалізувати реалістичний дим, що змінює свою форму залежно від руху персонажа, або воду, яка розбризкується при взаємодії з об'єктами.

UE4 також підтримує повну інтеграцію з іншими програмними продуктами, що дозволяє використовувати сторонні інструменти для створення моделей, текстур чи анімацій, але при цьому мати все необхідне для доопрацювання прямо у рушії. Наприклад, розробники можуть імпортувати 3D-моделі, створені в Blender або Maya, і з легкістю працювати з ними, використовуючи інструменти Unreal Engine. Це скорочує час на перенесення та сумісність між різними платформами, забезпечуючи ефективний робочий процес.

Ще однією перевагою Unreal Engine 4 є його гнучкість та розширюваність. Завдяки доступу до вихідного коду рушія розробники можуть налаштовувати його під свої потреби, створюючи унікальні інструменти чи модифікуючи існуючі. Це особливо корисно для великих студій, які розробляють проекти з нестандартною механікою або вимагають специфічних рішень.

Всі ці інструменти створені з урахуванням вимог індустрії, що робить Unreal Engine 4 універсальним рушієм для проектів різного масштабу. Незалежно від того, чи йдеться про створення мобільної гри, чи про розробку масштабного AAA-проекту, UE4 надає все необхідне для досягнення високоякісного результату. Розробники можуть фокусуватися на творчості, маючи в своєму розпорядженні надійний набір технологічних засобів, які економлять час і зусилля.

Система фізики в UE4 базується на PhysX від NVIDIA, що забезпечує високу точність симуляцій. Unreal Engine дозволяє реалізовувати:

- Динамічну взаємодію об'єктів у середовищі.
- Системи руйнувань (Destruction System), які створюють реалістичні сценарії руйнувань будівель або об'єктів.
- Симуляцію рідин та твердих тіл.

Unreal Engine 4 має потужну інтеграцію з іншими програмними продуктами, такими як Blender, ZBrush, Substance Painter. Це дозволяє легко імпортувати моделі, текстури та інші активи, зберігаючи їх якість та функціональність.

Unreal Engine 4 активно використовується для створення проектів у віртуальній (VR) та доповненій реальності (AR). Завдяки спеціальним інструментам, таким як VR Mode, розробники можуть проєктувати інтерактивні середовища, адаптовані для пристроїв, таких як Oculus Rift, HTC Vive або Microsoft HoloLens.

UE4 оснащений інструментами профілювання, такими як GPU Profiler, які дозволяють відслідковувати навантаження на графічний процесор і оптимізувати продуктивність проєкту. Також движок підтримує рівні деталізації (LOD), що знижує кількість полігонів для віддалених об'єктів.

Unreal Engine 4 підтримує широкий спектр платформ, включаючи Windows, macOS, Linux, iOS, Android, PlayStation, Xbox та HTML5. Це робить його універсальним рішенням для розробки ігор будь-якого масштабу.

Unreal Engine 4 — це інноваційний інструмент, який надає розробникам величезний набір можливостей для створення інтерактивних 3D-середовищ. Завдяки його технологічним особливостям, таким як високоякісний рендеринг, розширені системи фізики, інтеграція з іншими програмними продуктами та підтримка VR/AR, UE4 залишається одним із провідних движків у сфері розробки відеоігор.

1.3 Використання інших програмних інструментів: Blender, Substance Painter, ZBrush

Для створення інтерактивних 3D-середовищ у відеоіграх важливо використовувати не лише движок Unreal Engine 4, але й інші програмні інструменти, що доповнюють функціонал UE4. Ці інструменти дозволяють моделювати об'єкти, створювати текстури, матеріали та забезпечувати високий рівень деталізації. Найпопулярнішими серед таких інструментів є Blender, Substance Painter і ZBrush.

Blender — це безкоштовний інструмент для 3D-моделювання з відкритим кодом. Його можливості дозволяють створювати тривимірні моделі, анімації, симуляції фізики та рендеринг.

Основні функції Blender:

1. Полігональне моделювання: Blender забезпечує інтуїтивний інтерфейс для створення геометрії, підтримує як низькополігональні (low-poly), так і високополігональні (high-poly) моделі.

2. Текстурування: завдяки вбудованому UV-розгортанню користувачі можуть створювати точні текстури для моделей.

3. Скульптинг: Blender пропонує інструменти для роботи над деталізацією моделей, дозволяючи додавати складні елементи, такі як зморшки, тріщини або текстуру шкіри.

4. Фізичні симуляції: ідеально підходить для створення реалістичних симуляцій рідин, диму, тканин та інших матеріалів.

Blender інтегрується з Unreal Engine 4 завдяки формату FBX, що дозволяє без проблем експортувати моделі з усіма текстурами та анімаціями.

Substance Painter від Adobe — це спеціалізоване програмне забезпечення для текстуровання і створення матеріалів. Воно є невід'ємним інструментом для надання реалістичного вигляду 3D-моделям.

Ключові можливості Substance Painter:

1. Процедурне текстуровання: інструмент дозволяє створювати текстури за допомогою процедурних масок і шаблонів.

2. Інструменти для малювання: Substance Painter підтримує малювання текстур безпосередньо на моделі, що значно підвищує точність роботи.

3. Підтримка PBR: завдяки Physically Based Rendering текстури виглядають реалістично у різних умовах освітлення.

4. Ефекти в реальному часі: під час створення текстур можна застосовувати ефекти, такі як подряпини, бруд, пошкодження.

Substance Painter має потужну інтеграцію з Unreal Engine 4, дозволяючи експортувати готові текстури без втрати якості та автоматично налаштовувати матеріали у движку.

ZBrush — професійне програмне забезпечення для цифрового скульптингу. Воно використовується для створення високополігональних моделей з високим рівнем деталізації, які пізніше можна оптимізувати для використання у відеоіграх.

Основні функції ZBrush:

1. Скульптинг: ZBrush дозволяє створювати складні моделі завдяки інструментам для додавання деталей, таких як шкіра, волосся чи текстура каменю.

2. Динамічні підрозділи (Dynamesh): ця функція дозволяє змінювати топологію моделі без втрати її форми, що зручно для ранніх етапів розробки.

3. Ретопологія: Після створення моделі з високим числом полігонів її можна спростити для використання у відеоіграх.

4. Текстурування: ZBrush підтримує роботу з PolyPaint, що дозволяє малювати текстури без необхідності створення UV-розгортки.

ZBrush зазвичай використовується для створення персонажів або складних органічних об'єктів, таких як рослини, скелі чи текстури землі. Готові моделі можуть бути імпортовані в Unreal Engine 4 для подальшої роботи.

Поєднання інструментів Blender, Substance Painter та ZBrush дозволяє створювати високоякісні 3D-об'єкти для інтерактивних середовищ. Кожен із цих інструментів має унікальні переваги, які доповнюють функціонал Unreal Engine 4. Використання цих програм у комплексі забезпечує ефективний робочий процес, починаючи від базового моделювання і закінчуючи фінальним текстуруванням та оптимізацією для відеоігор.

1.4 Аналіз ринку інтерактивних 3D-середовищ для відеоігор

Інтерактивні 3D-середовища є одним із ключових елементів сучасних відеоігор. Зростання їх популярності обумовлене постійним розвитком технологій, зростанням потужностей апаратного забезпечення та попитом на більш реалістичний і захопливий ігровий досвід. У цьому розділі розглянемо сучасні тенденції ринку, основних гравців, сфери застосування інтерактивних середовищ та фактори, які впливають на розвиток цієї галузі.

Тенденції ринку інтерактивних 3D-середовищ

1. Зростання популярності open-world ігор. Відкриті світи, які надають гравцям свободу дій, стали одним із провідних напрямків у розробці відеоігор. Інтерактивні 3D-середовища для таких проєктів, як The Legend of Zelda: Breath of the Wild, Red Dead Redemption 2 або Elden Ring, стають усе більшими і деталізованішими. Це вимагає використання передових 3D-двигків, таких як Unreal Engine 4, Unity або Frostbite.

2. Розвиток VR та AR. Технології віртуальної та доповненої реальності сприяють появі нових ігрових жанрів, у яких інтерактивність середовища виходить на перший план. Наприклад, проєкти, як *Half-Life: Alyx*, використовують інтерактивні об'єкти, фізику та реалістичні симуляції, створені завдяки інтеграції сучасних движків із VR-платформами.

3. Фокус на реалістичності та деталізації. Завдяки підтримці технологій, таких як Ray Tracing, PBR (Physically Based Rendering), DLSS від NVIDIA, графіка в сучасних 3D-середовищах досягає кінематографічного рівня якості. Ця тенденція особливо актуальна для AAA-ігор.

4. Попит на процедурно згенеровані середовища. Процедурна генерація дозволяє створювати великі світи з мінімальними затратами ресурсів. Прикладами таких ігор є *No Man's Sky* та *Minecraft*. Вони демонструють, як інтерактивність і масштаб середовища впливають на ігровий процес.

Тенденції ринку інтерактивних 3D-середовищ у відеоіграх визначають сучасний напрямок розвитку індустрії, орієнтуючись на зростаючі очікування гравців і технологічні можливості. Одним із ключових факторів, що впливають на популярність ігор, є інтерактивність і реалістичність середовищ, які дозволяють гравцям занурюватися у віртуальні світи з максимальним відчуттям присутності.

Однією з найпомітніших тенденцій є зростання популярності ігор із відкритими світами (*open-world games*). Цей жанр став домінуючим завдяки своїй здатності надавати гравцям свободу дій і можливість дослідження великих, деталізованих просторів. Проєкти, такі як *The Legend of Zelda: Breath of the Wild*, *Red Dead Redemption 2* та *Elden Ring*, вражають не лише масштабом, а й рівнем інтерактивності. У таких іграх середовище часто є більше, ніж фоном — воно стає активним учасником ігрового процесу. Наприклад, у *Breath of the Wild* погода, фізика і поведінка об'єктів створюють додаткові механіки для взаємодії, тоді як у *Red Dead Redemption 2* увага до деталей забезпечує реалістичність світу. Реалізація подібних проєктів потребує використання потужних 3D-движків, таких як *Unreal Engine 4* або *Unity*, які забезпечують не тільки графіку високої якості, а й можливість створення комплексної симуляції.

Розвиток технологій віртуальної реальності (VR) і доповненої реальності (AR) також значно впливає на ринок інтерактивних 3D-середовищ. VR та AR дозволяють створювати абсолютно нові жанри, у яких гравці взаємодіють із середовищем у фізично природний спосіб. Наприклад, у грі *Half-Life: Alyx* інтерактивність досягається за допомогою реалістичної фізики, складних механік взаємодії з об'єктами та візуальної правдоподібності. Сучасні 3D-движки інтегруються з платформами VR, надаючи розробникам можливість реалізовувати проєкти, які відзначаються високим рівнем занурення та інтерактивності. Гравці можуть буквально "торкатися" об'єктів, переміщатися у віртуальному просторі та впливати на нього, що відкриває нові горизонти для ігор, симуляцій і тренувальних програм.

Ще однією важливою тенденцією є зростання уваги до реалістичності й деталізації 3D-середовищ. Завдяки сучасним технологіям, таким як *Ray Tracing* (відстеження променів) і *Physically Based Rendering* (PBR), графіка у відеоіграх досягла кінематографічного рівня. Технології від *NVIDIA*, такі як *DLSS* (Deep Learning Super Sampling), дозволяють підвищувати якість зображення без значного навантаження на обладнання. Ці досягнення особливо важливі для AAA-ігор, де деталізація кожного елемента світу — від текстури поверхонь до складних тіней і відблисків — є частиною загального ігрового досвіду. Реалістичність середовищ не тільки сприяє зануренню, а й відкриває нові можливості для ігрової механіки. Наприклад, у грі з реалістичним освітленням гравець може використовувати тіні для приховування або орієнтуватися у просторі за відблисками світла.

Процедурна генерація стає важливим інструментом у створенні масштабних і динамічних 3D-середовищ. Цей підхід дозволяє автоматично створювати ландшафти, будівлі та навіть цілі світи за допомогою алгоритмів, що економить час і ресурси розробників. Наприклад, гра *No Man's Sky* використовує процедурну генерацію для створення унікальних планет, що дає змогу гравцям досліджувати майже нескінченний космос. Подібний підхід застосовується і в *Minecraft*, де середовище генерується в реальному часі, забезпечуючи унікальний досвід для

кожного гравця. Процедурна генерація має особливу цінність у контексті інтерактивності, оскільки дозволяє створювати світи, які реагують на дії гравця або змінюються залежно від певних умов.

Таким чином, сучасний ринок інтерактивних 3D-середовищ розвивається у напрямках, що поєднують технологічний прогрес із зростаючими вимогами гравців до якості та реалістичності. Інновації в галузі рендерингу, процедурної генерації, VR/AR і open-world ігор трансформують підходи до розробки ігор, роблячи їх дедалі складнішими й захопливішими. Ці тенденції не лише визначають майбутнє відеоігор, а й сприяють створенню інтерактивних середовищ для інших сфер, таких як освіта, медицина чи архітектура.

Ринок інтерактивних 3D-середовищ представлений кількома категоріями учасників:

1. Розробники движків
 - Unreal Engine (Epic Games): забезпечує масштабованість, високу графічну якість і підтримку VR/AR.
 - Unity: орієнтований на інди-розробників і мобільні платформи.
 - CryEngine (Crytek): відомий своїми передовими візуальними ефектами.
2. Розробники ігор
 - CD Projekt Red: відома інтерактивними світами в серії The Witcher і Cyberpunk 2077.
 - Rockstar Games: лідер у створенні open-world середовищ (Grand Theft Auto, Red Dead Redemption).
 - Ubisoft: застосовує інтерактивні світи у франшизах Assassin's Creed, Far Cry.
3. Технологічні компанії
 - NVIDIA та AMD: розробляють графічні процесори та інструменти для роботи з 3D-графікою (Ray Tracing, DLSS).
 - Autodesk: пропонує програмне забезпечення для моделювання та анімації (Maya, 3ds Max).

Хоча основним напрямком використання інтерактивних 3D-середовищ залишається ігрова індустрія, їх застосовують і в інших галузях:

- Кінематограф: приклади включають серіал *The Mandalorian*, де використовувався Unreal Engine для створення інтерактивних фонів.
- Архітектурна візуалізація: інтерактивні тури в будинках або офісах.
- Освіта: симуляції для навчання, наприклад, у медицині або авіації.
- Військові тренажери: реалістичне відтворення бойових сценаріїв.

Інтерактивні 3D-середовища, хоча й асоціюються переважно з ігровою індустрією, знаходять усе ширше застосування в інших галузях, де візуалізація, інтерактивність і реалістичність відіграють ключову роль. Завдяки можливостям сучасних 3D-движків, таких як Unreal Engine, Unity та інші, створення інтерактивних віртуальних середовищ стає доступним для вирішення завдань у різноманітних професійних і творчих сферах.

Кінематограф є одним із яскравих прикладів нетрадиційного використання інтерактивних 3D-середовищ. У таких проєктах, як серіал *The Mandalorian*, технології, що зазвичай застосовуються у відеоіграх, використовувалися для створення віртуальних фонів у реальному часі. Завдяки Unreal Engine розробники змогли створити віртуальні сцени, які синхронізувалися з камерою і акторською грою, забезпечуючи природність зображення та інтерактивність у процесі зйомки. Цей підхід відкриває нові горизонти для кінематографа, дозволяючи режисерам експериментувати зі складними візуальними ефектами без необхідності великих затрат на побудову декорацій чи тривалу постобробку.

В архітектурній візуалізації інтерактивні 3D-середовища стали потужним інструментом для презентації проєктів. Раніше статичні рендери будівель і інтер'єрів сьогодні поступаються місцем інтерактивним турам, які дозволяють клієнтам або інвесторам буквально "прогулятися" майбутнім простором. Завдяки VR або простим десктопним платформам, користувачі можуть змінювати параметри освітлення, кольори стін, розташування меблів і навіть оцінювати акустику приміщення в реальному часі. Це не тільки покращує комунікацію між архітекторами та замовниками, а й значно скорочує час на погодження проєктів.

Сфера освіти також активно інтегрує технології інтерактивних 3D-середовищ для створення ефективніших навчальних програм. У медицині, наприклад, симуляції хірургічних операцій дозволяють студентам та лікарям-практикантам відточувати свої навички у віртуальному просторі, не ризикуючи життям пацієнтів. Подібні симуляції забезпечують високу деталізацію анатомії людини, моделюють реальні сценарії ускладнень і дозволяють адаптуватися до непередбачуваних ситуацій. В авіації інтерактивні тренажери дозволяють пілотам проходити підготовку на симуляторах, які точно імітують поведінку літака в різних умовах, включаючи екстремальні погодні умови або технічні несправності.

Військова справа також використовує інтерактивні 3D-середовища для тренувань солдатів і відпрацювання бойових сценаріїв. Завдяки віртуальним симуляторам військові можуть відтворювати реалістичні бойові ситуації, координувати дії груп, розробляти стратегії та адаптуватися до різних умов на полі бою. Застосування таких тренажерів знижує ризики під час реальних тренувань і дає можливість економити ресурси, необхідні для організації польових навчань.

Крім того, інтерактивні 3D-середовища знаходять своє місце у маркетингу, дозволяючи компаніям створювати вражаючі презентації своїх продуктів. Наприклад, автомобільні концерни використовують віртуальні шоуруми, де потенційні клієнти можуть взаємодіяти з моделями автомобілів, змінювати їхні характеристики та навіть тестувати віртуальний водійський досвід. У сфері нерухомості інтерактивні тури дозволяють покупцям оглядати будинки чи квартири, не залишаючи своїх осель.

Розширення застосування інтерактивних 3D-середовищ також стимулює розвиток суміжних технологій, таких як штучний інтелект, машинне навчання та обробка великих даних. Наприклад, інтеграція 3D-движків із штучним інтелектом дозволяє створювати складніші симуляції, де віртуальні персонажі демонструють реалістичну поведінку, адаптуючись до дій користувача.

Таким чином, інтерактивні 3D-середовища стали універсальним інструментом, який знаходить застосування у дедалі ширшому колі галузей. Їхнє

використання не лише оптимізує процеси та підвищує ефективність, але й відкриває нові можливості для творчості, навчання та розвитку. Інтерактивні 3D-середовища стають невід'ємною частиною сучасного світу, сприяючи інтеграції цифрових технологій у повсякденне життя та професійну діяльність.

Чинники, що впливають на розвиток ринку

1. Технічний прогрес. Розвиток апаратного забезпечення, зокрема графічних процесорів, дозволяє створювати дедалі складніші 3D-середовища. Інтеграція з технологіями машинного навчання також відкриває нові можливості для автоматизації процесів.
2. Економічні фактори. Зростання витрат на розробку AAA-ігор змушує студії вкладати більше ресурсів у оптимізацію процесів створення 3D-середовищ.
3. Попит з боку користувачів. Геймери очікують вищої якості графіки та глибшої інтерактивності, що стимулює інновації в галузі.

Проблеми та виклики:

1. Висока вартість виробництва. Створення реалістичних 3D-середовищ потребує значних фінансових і часових ресурсів.
2. Оптимізація для різних платформ. Важливо забезпечити баланс між якістю графіки та продуктивністю для різних платформ, зокрема консолей і мобільних пристроїв.
3. Екологічний вплив. Використання великих серверів і обчислювальних ресурсів під час створення і рендерингу 3D-середовищ має значний вуглецевий слід.

Ринок інтерактивних 3D-середовищ для відеоігор динамічно розвивається, забезпечуючи користувачів усе більш реалістичними та інтерактивними світи. Інноваційні технології, зокрема Ray Tracing, процедурна генерація та VR/AR, формують майбутнє цієї галузі. Однак розробникам доводиться вирішувати численні виклики, пов'язані з високою вартістю виробництва та технічною складністю інтеграції. Незважаючи на це, зростаючий попит і можливості технологічного прогресу сприятимуть подальшому розширенню ринку та появі нових інновацій.

2 МЕТОДОЛОГІЯ СТВОРЕННЯ ІНТЕРАКТИВНИХ 3D-СЕРЕДОВИЩ НА ОСНОВІ UNREAL ENGINE 4

2.1 Основи проєктування і розробки 3D-середовищ

Методологія включає низку етапів, таких як планування, створення та оптимізація моделей, налаштування матеріалів, інтеграція фізики та освітлення, а також програмування інтерактивних елементів середовища.

На початковому етапі розробки 3D-середовища необхідно визначити основні вимоги до середовища, такі як стиль, масштаб, структура та інтерактивні елементи. Також важливо скласти загальну концепцію рівня, зокрема сценарій взаємодії користувача з середовищем, щоб проєктування було послідовним і ефективним.

Підготовка технічного завдання: визначення функціональних та графічних вимог, плану інтерактивності.

Розробка концепт-арту: створення базових ескізів об'єктів, основних локацій та загальної естетики проєкту.

Планування етапів розробки: підготовка часових меж для виконання окремих етапів, таких як створення моделей, текстуровання, програмування тощо.

Етап моделювання включає створення 3D-об'єктів, які будуть розміщені у віртуальному середовищі. Для цього зазвичай використовуються зовнішні програми, такі як Blender або Autodesk Maya, після чого моделі імпортуються в Unreal Engine 4.



Рис. 2.1 Концепція локації

Створення низькополігональних моделей для оптимізації: розробка моделей із низькою кількістю полігонів для забезпечення продуктивності.

Деталізація ключових об'єктів: використання скульптингу та текстурівання для створення високодеталізованих моделей.

Оптимізація та ретопологія: зменшення кількості полігонів на об'єктах, які не є ключовими для забезпечення плавності рендерингу.

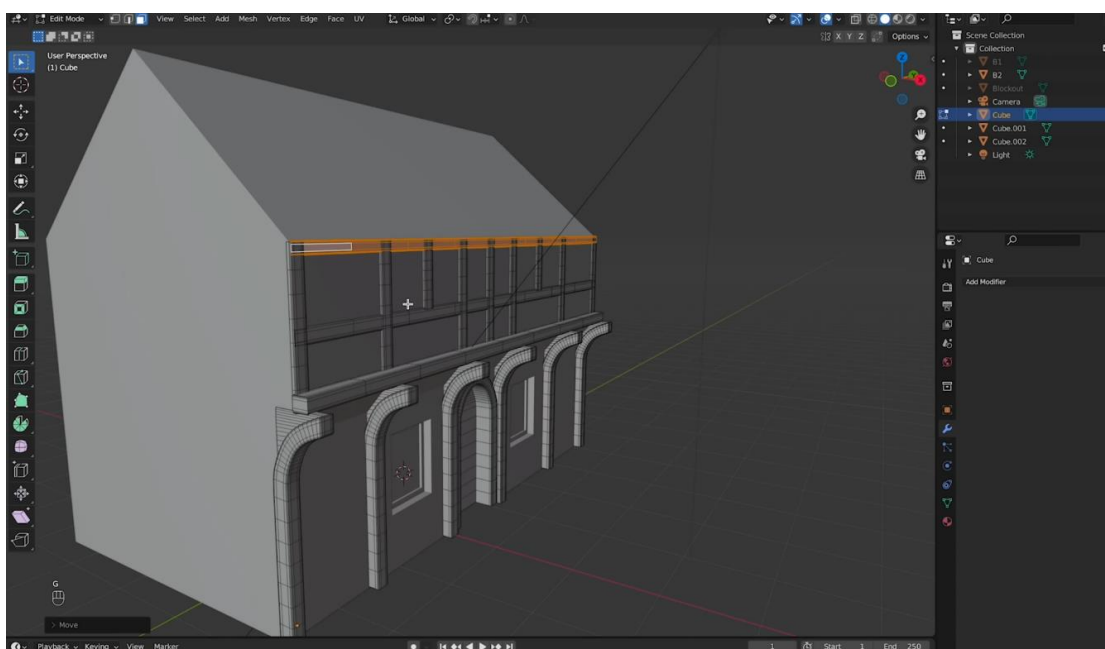


Рис. 2.2 Моделювання будинку

На цьому етапі розробляються матеріали та текстури, які додають реалізму 3D-середовищу. UE4 надає можливість налаштування матеріалів за допомогою візуального редактора, що дозволяє створювати різноманітні ефекти, наприклад, блискучі або матові поверхні, ефект зносу, прозорість тощо.

Використання PBR (Physically Based Rendering) для реалізму: підхід до створення матеріалів, який дозволяє відтворити властивості різних поверхонь.

Створення карт нормалей та текстур високої роздільної здатності: для додання глибини об'єктам та симуляції деталей.

Візуальні ефекти (VFX): додавання ефектів, таких як дим, вода, вогонь, що підсилюють інтерактивність.

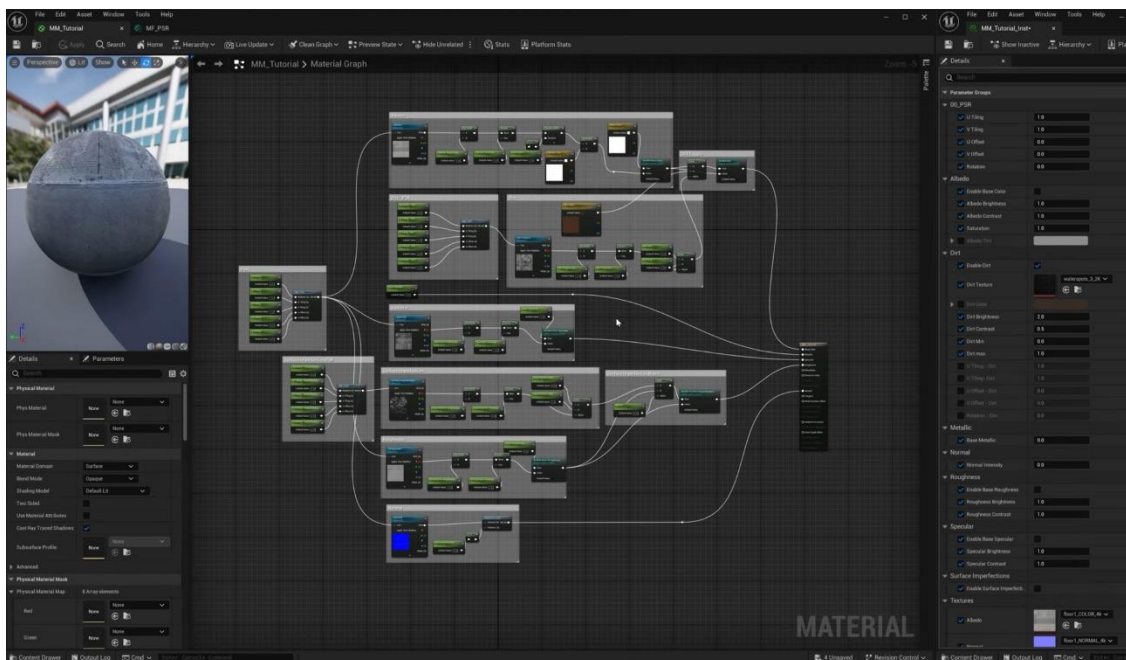


Рис. 2.3 Приклад текстури

Правильне освітлення є важливим для створення атмосфери та передачі відчуття глибини. Unreal Engine 4 підтримує динамічне та статичне освітлення, а також постобробні ефекти, що дозволяють додати реалізму до сцен.

Налаштування статичного і динамічного освітлення: використання різних типів джерел світла (точкове, направлене, навколишнє світло) для формування об'ємності сцени.

Постобробка зображення: налаштування ефектів, таких як глибина різкості, віньєтування, хроматичні аберації для досягнення кіношного вигляду.

Глобальне освітлення (Global Illumination): налаштування, яке імітує природне відбиття світла для більшої правдоподібності сцени.

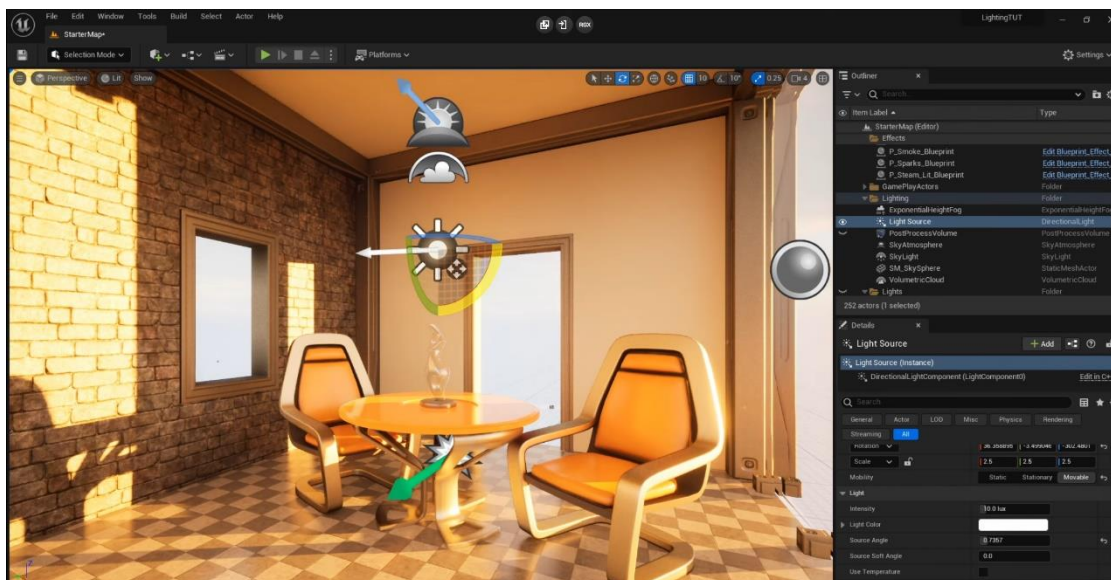


Рис. 2.4 Приклад освітлення

Для створення інтерактивного середовища в UE4 використовується система Blueprints, яка дозволяє створювати ігрову логіку без написання коду. Також можливо використовувати мову програмування C++ для більш глибокої кастомізації та розширення функціональності.

Налаштування тригерів і подій: використання Blueprints для активації подій при взаємодії з об'єктами.

Скрипти поведінки: створення поведінки для рухомих об'єктів, анімацій, ефектів та інших інтерактивних елементів.

Взаємодія з гравцем: програмування елементів управління, систем зворотного зв'язку (ефекти при натисканні, вібрація контролерів тощо).

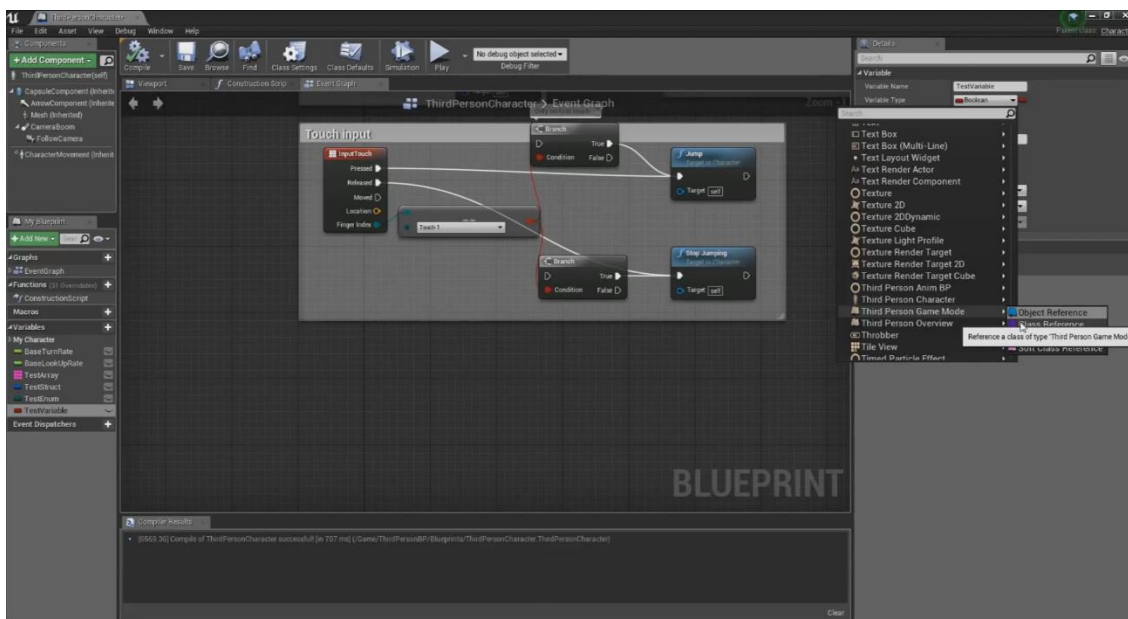


Рис. 2.5 Приклад Blueprints

Оптимізація є важливим етапом, який допомагає забезпечити стабільну роботу гри на різних платформах і уникнути зниження продуктивності. Unreal Engine 4 надає інструменти для аналізу продуктивності, які дозволяють знаходити вузькі місця і покращувати їх.

Ліміт полігонів та LOD (Level of Detail): налаштування автоматичного зменшення кількості полігонів для віддалених об'єктів.

Оптимізація текстур і матеріалів: використання оптимізованих текстурних карт і простіших матеріалів для забезпечення стабільної роботи.

Профілювання та тестування продуктивності: використання інструментів Unreal Engine для оцінки FPS, завантаження процесора та відеокarti.

Однією з найважливіших особливостей Unreal Engine 4 є його здатність працювати з великими відкритими світами, забезпечуючи при цьому високу продуктивність. Система Level Streaming дозволяє підвантажувати лише необхідні ділянки карти, що суттєво знижує навантаження на систему.

Для забезпечення візуальної реалістичності й оптимізації рушій використовує технологію Level of Detail (LOD). Об'єкти на сцені автоматично змінюють рівень деталізації залежно від відстані до гравця, що дозволяє зберігати високу частоту кадрів навіть у насичених сценах.

Однією з найбільш перспективних функцій Blender є Geometry Nodes, яка дозволяє створювати складні об'єкти процедурним способом. Це особливо корисно для створення природних середовищ, таких як ландшафти або лісові масиви, де кожен елемент має випадкові варіації, але залишається частиною загальної структури. Geometry Nodes також дозволяють автоматизувати рутинні процеси, наприклад, розташування трави на ландшафті або розподіл каменів уздовж доріг.

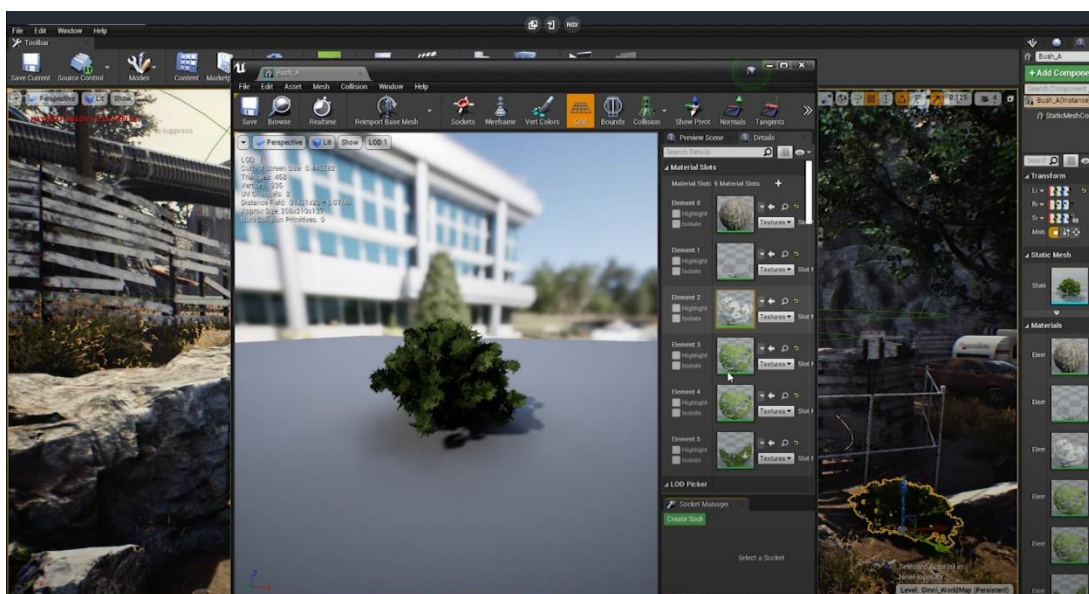


Рис. 2.6 Приклад оптимізації

Останнім етапом є тестування готового середовища на цільових платформах (ПК, консолі, мобільні пристрої). Unreal Engine 4 дозволяє тестувати проєкт на різних пристроях для виявлення помилок і забезпечення стабільної роботи.

Підготовка до розгортання на платформі: налаштування параметрів, специфічних для обраної платформи.

Усунення багів та недоліків: виправлення помилок та тестування на продуктивність.

Тестування інтерактивності та відповідності вимогам: перевірка зручності взаємодії користувача з середовищем та відповідності проєкту технічному завданню.

Unreal Engine 4 часто використовується в галузях поза ігровою індустрією:

- Архітектурна візуалізація: створення інтерактивних віртуальних турів для демонстрації дизайну інтер'єрів або екстер'єрів. Висока якість графіки й підтримка реалістичного освітлення дозволяють клієнтам візуалізувати майбутні проєкти в деталях.

- Освітні симуляції: університети та навчальні заклади використовують Unreal Engine для створення віртуальних лабораторій, де студенти можуть експериментувати без ризику для життя чи майна.

- Віртуальна реальність (VR): використовується для навчання медичних працівників або пілотів у симуляторах, що відтворюють реалістичні сценарії.

Blender часто використовується в індустрії VFX через свою інтеграцію з фізичними симуляціями. Наприклад, система Mantaflow дозволяє створювати реалістичні вибухи, дим або потоки рідин. Завдяки рушію Cycles і підтримці GPU-рендерингу, Blender може створювати якісні кінематографічні ефекти, які використовуються в короткометражних фільмах і анімації.

2.2 Робота з системою Blueprints у Unreal Engine 4

Система Blueprints в Unreal Engine 4 є однією з ключових особливостей цього движка. Вона дозволяє створювати складну ігрову логіку без використання текстового програмування, спрощуючи процес розробки та роблячи його доступним навіть для розробників без ґрунтовних знань у програмуванні. У цьому розділі розглянемо основні принципи роботи з Blueprints, їх можливості та роль у створенні інтерактивних 3D-середовищ.

Blueprints — це візуальна скриптова мова, заснована на архітектурі об'єктно-орієнтованого програмування. Кожен Blueprint є певним класом, який може містити властивості (variables), функції (functions), події (events) та графічні елементи для створення взаємодії.

Типи Blueprints у Unreal Engine 4:

1. Level Blueprints. Використовуються для створення логіки, яка пов'язана з рівнем. Наприклад, запуск катсцени, зміна освітлення або інтерактивні події, пов'язані з місцем.

2. Class Blueprints. Дозволяють створювати логіку для окремих об'єктів (актори, персонажі, транспорт).

3. Widget Blueprints. Використовуються для створення елементів інтерфейсу користувача (UI).

4. Animation Blueprints. Призначені для управління анімаціями персонажів, таких як рух, дихання або атака.

У Unreal Engine 4 система Blueprints є потужним інструментом для створення ігрової логіки без необхідності використання традиційного програмування. Вона дозволяє розробникам зосередитися на креативних аспектах проекту, водночас забезпечуючи можливості для побудови складних і динамічних сценаріїв. Різні типи Blueprints допомагають структурувати процес розробки, забезпечуючи розділення логіки залежно від її призначення.

Level Blueprints відіграють ключову роль у створенні логіки, яка пов'язана безпосередньо з рівнем. Вони дозволяють задавати події, що відбуваються у конкретному ігровому просторі. Наприклад, за допомогою Level Blueprints можна реалізувати такі механіки, як активація катсцени при досягненні певної точки на карті, зміна освітлення залежно від часу доби, або запуск пастки, якщо гравець наступає на певний тригер. Також цей тип Blueprint може управляти специфічними подіями, що залежать від взаємодії гравця з елементами навколишнього середовища, наприклад відкривання дверей чи активація механізмів.

Class Blueprints надають розробникам можливість створювати логіку для окремих об'єктів, таких як персонажі, транспорт або предмети. Цей тип Blueprint дозволяє задавати поведінку об'єкта, включаючи взаємодії з іншими елементами гри. Наприклад, можна створити персонажа, який реагує на дії гравця: змінює свій стан, виконує анімацію або запускає певний сценарій. Для транспортних засобів Class Blueprints можуть реалізувати фізику руху, керування та взаємодію з

довкіллям. Завдяки широким можливостям кастомізації Class Blueprints стають основним інструментом для створення ігрових механік і наповнення світу динамічними елементами.

Widget Blueprints спрямовані на розробку інтерфейсу користувача (UI), що є одним із критичних аспектів будь-якої гри. Інтуїтивно зрозумілий і візуально привабливий інтерфейс значно покращує досвід гравця. Widget Blueprints дозволяють створювати кнопки, шкали здоров'я, меню, діалогові вікна, карти та інші елементи UI. Інтеграція цих елементів з ігровою логікою здійснюється легко, оскільки Widget Blueprints можуть обмінюватися даними з іншими Blueprint-системами. Наприклад, шкала здоров'я може динамічно змінюватися відповідно до поточного стану персонажа, або індикатор завдань оновлюватись у реальному часі залежно від прогресу гравця.

Animation Blueprints використовуються для управління анімаціями персонажів і об'єктів. Вони дозволяють задавати послідовність дій, перемикатися між різними станами анімації та синхронізувати їх із ігровою логікою. Наприклад, Animation Blueprints можуть забезпечувати плавність переходу між ходьбою, бігом і стрибком персонажа. Крім того, цей тип Blueprint дозволяє реагувати на різні ігрові події: анімація атаки може запускатися під час натискання кнопки, або персонаж змінює позицію залежно від кута огляду камери. Animation Blueprints також підтримують інверсну кінематику, що забезпечує реалістичність взаємодії персонажа з оточенням, наприклад, адаптацію рухів до поверхонь різної висоти чи нахилу.

Важливо зазначити, що всі ці типи Blueprints можуть взаємодіяти один з одним, утворюючи комплексні системи. Наприклад, Animation Blueprints можуть отримувати дані від Class Blueprints для запуску анімації, а Widget Blueprints можуть передавати інформацію до Level Blueprints для відображення прогресу гравця на рівні. Такий підхід дозволяє підтримувати модульність у проєкті, що полегшує обслуговування та розширення функціоналу гри.

Узагальнюючи, система Blueprints у Unreal Engine 4 є надзвичайно гнучким інструментом, що дозволяє створювати інтерактивну ігрову логіку на будь-якому

рівні деталізації. Різноманіття типів Blueprints допомагає оптимізувати робочий процес, розділяючи логіку на відповідні категорії, що спрощує розробку навіть для великих команд. Завдяки цьому інструменту Unreal Engine 4 залишається однією з провідних платформ для розробки відеоігор та інтерактивних 3D-додатків.

Система Blueprints базується на графічному інтерфейсі, де основними компонентами є вузли (nodes), які з'єднуються між собою, утворюючи логічні блоки.

Основні типи вузлів:

1. Event Nodes — ініціюють виконання логіки. Наприклад, *Event Begin Play* запускається, коли об'єкт з'являється у грі.
2. Execution Nodes — виконують конкретні дії, такі як переміщення об'єкта, відтворення звуку або виклик функції.
3. Variable Nodes — зберігають дані (позиція, швидкість, кількість життів).
4. Flow Control Nodes — управляють потоком виконання через умовні оператори (*Branch, For Loop, While Loop*).

Створення інтерактивності за допомогою Blueprints

1. Обробка подій. Blueprints дозволяє створювати події, які активуються взаємодією гравця. Наприклад, якщо персонаж підходить до дверей, вони можуть автоматично відчинитися. Це реалізується через *Event Overlap*, який перевіряє, чи увійшов об'єкт до певної зони тригера.
2. Взаємодія з об'єктами. Для інтерактивності у 3D-середовищах важливо створити можливість взаємодії з об'єктами. Наприклад, коли гравець натискає клавішу «Е», система Blueprints може запустити анімацію підняття предмета.
3. Скрипти для ігрових механік. Можна налаштувати фізику руху персонажа, створити механіку бою або систему інвентаря, використовуючи Blueprint Functions та Variables.

4. Динамічна зміна середовища. Blueprints дозволяє створювати середовища, які змінюються залежно від дій гравця. Наприклад, якщо гравець активує важіль, це може змінити розташування об'єктів або запустити новий сценарій.

Приклади використання Blueprints у 3D-середовищах

1. Система дверей із ключами. Для створення інтерактивних дверей у грі можна використати тригери та змінні:

- Перевірити, чи має гравець ключ.
- Якщо так, виконати анімацію відчинення дверей через *Timeline Node*.

2. Динамічне освітлення. Blueprints дозволяє змінювати інтенсивність освітлення в реальному часі залежно від часу доби або дій гравця (наприклад, активація ліхтаря).

3. Система погодних ефектів. Використовуючи *Event Begin Play*, можна створити ефект дощу або снігу, який змінюється залежно від ігрових умов.

Переваги використання Blueprints:

1. Простота у використанні. Blueprints значно знижує поріг входу у розробку. Для роботи з ними не потрібно знати текстові мови програмування, що робить їх ідеальним вибором для початківців.

2. Швидкий прототипінг. Завдяки візуальній природі Blueprints можна швидко створювати і тестувати ігрові механіки без витрат на написання коду.

3. Гнучкість. Незважаючи на свою простоту, Blueprints дозволяють реалізовувати складні ігрові сценарії завдяки підтримці функцій, масивів та циклів.

4. Інтеграція з текстовим кодом. Blueprints можуть працювати разом із C++ для створення складніших систем, що поєднує візуальну простоту з можливостями програмування.

Система Blueprints у Unreal Engine 4 є одним із найпотужніших інструментів для розробки ігрових механік, що поєднує інтуїтивно зрозумілий інтерфейс із широкими можливостями. Однією з її головних переваг є простота у використанні. Цей підхід особливо цінний для початківців, які тільки

розпочинають свій шлях у розробці відеоігор. Завдяки візуальній природі Blueprints користувачі можуть створювати складну логіку, не володіючи знаннями в текстових мовах програмування. Це дозволяє значно знизити поріг входу в галузь і стимулює більшу кількість розробників долучатися до проєктів, використовуючи цей інструмент.

Ще однією перевагою Blueprints є можливість швидкого прототипування. У традиційному програмуванні розробка і тестування нових ігрових механік можуть займати значний час, оскільки вимагають написання коду, його налагодження та інтеграції. Blueprints, завдяки своїй візуальній природі, дозволяє створювати базові версії ігрових механік за лічені хвилини. Наприклад, додавання тригера для відкриття дверей або запуску катсцени може бути реалізовано безпосередньо в редакторі за допомогою простих вузлів. Це забезпечує не лише економію часу, але й дає змогу експериментувати з ідеями, швидко відсіваючи ті, що не відповідають очікуванням.

Попри свою простоту, Blueprints забезпечують високу гнучкість у роботі. Цей інструмент підтримує базові програмні конструкції, такі як функції, масиви та цикли. Це дозволяє реалізовувати складні ігрові сценарії, які враховують множинні взаємодії між об'єктами, умови та стани гри. Наприклад, можна створити систему, яка перевіряє кілька умов для активації певного сценарію, таких як досягнення гравцем певного рівня, наявність у нього необхідних ресурсів чи виконання певної дії. Завдяки цьому Blueprints стають ефективним інструментом навіть для досвідчених розробників, які прагнуть спростити свою роботу, не втрачаючи функціональності.

Інтеграція Blueprints із текстовим кодом на мові C++ відкриває нові можливості для розробників. Хоча Blueprints чудово підходять для більшості завдань, є ситуації, коли потрібна висока продуктивність або доступ до низькорівневих функцій движка. У таких випадках можна комбінувати Blueprints із C++, використовуючи візуальний скриптинг для швидкого прототипування та розробки базової логіки, а текстовий код — для створення оптимізованих і складних систем. Наприклад, складний штучний інтелект або фізична симуляція

можуть бути написані на C++, тоді як їхня інтеграція з іншими аспектами гри здійснюється через Blueprints.

Важливою перевагою Blueprints є їхня роль у командній роботі. Візуальний підхід до створення логіки дозволяє учасникам команди, які не мають технічного досвіду, краще розуміти структуру проєкту та пропонувати свої ідеї. Художники, дизайнери рівнів або навіть менеджери можуть легко переглядати Blueprints, щоб зрозуміти, як працює той чи інший механізм. Це значно спрощує комунікацію всередині команди та робить процес розробки більш прозорим.

Окрім цього, Blueprints ідеально підходять для навчання новачків. Візуальна структура дозволяє швидко опанувати базові принципи програмування, такі як умовні оператори, цикли та функції. Це створює міцний фундамент для переходу до більш складних мов програмування, таких як C++ чи Python. Багато навчальних матеріалів, доступних для Unreal Engine 4, орієнтовані саме на Blueprints, що робить процес навчання доступним для широкої аудиторії.

Таким чином, використання Blueprints у Unreal Engine 4 забезпечує широкий спектр переваг для розробників будь-якого рівня досвіду. Простота у використанні, швидкість прототипування, гнучкість, інтеграція з текстовим кодом і можливість командної роботи роблять цей інструмент незамінним у сучасній ігровій індустрії. Завдяки Blueprints розробники отримують унікальну можливість поєднати творчий процес із технологічними інноваціями, створюючи захоплюючі ігрові світи.

Хоча система Blueprints має багато переваг, існують і певні обмеження:

1. Менша продуктивність у порівнянні з C++ у складних проєктах.
2. Труднощі в організації великого проєкту — коли проєкт містить багато вузлів, граф стає складним для розуміння.

Система Blueprints у Unreal Engine 4 є потужним інструментом для розробки інтерактивних 3D-середовищ. Вона дозволяє створювати ігрову логіку швидко і без складного програмування, що робить її важливим елементом у розробці відеоігор. Незважаючи на деякі обмеження, Blueprints ідеально підходить як для

швидкого прототипування, так і для створення повноцінних ігрових механік, забезпечуючи гнучкість і доступність для розробників різного рівня.

2.3 Використання фізичних симуляцій і реалістичного рендерингу

У сучасних відеоіграх фізичні симуляції та реалістичний рендеринг відіграють ключову роль, адже вони забезпечують максимальне занурення гравця у віртуальний світ. Unreal Engine 4 (UE4) надає розробникам широкий набір інструментів для створення реалістичних ефектів, фізики взаємодії об'єктів та високоякісного рендерингу, що дозволяє досягти високого рівня візуалізації.

Фізичні симуляції забезпечують реалістичну поведінку об'єктів у 3D-середовищі, використовуючи закони фізики. UE4 має вбудований фізичний рушій, який дозволяє відтворювати різні фізичні явища, такі як гравітація, зіткнення, деформації та динаміка рідин.

Unreal Engine 4 пропонує широкий набір інструментів для симуляції фізичних процесів, що надає інтерактивним 3D-середовищам реалістичності та динамічності. Завдяки вбудованим фізичним системам розробники можуть створювати ігрові світи, які відповідають законам фізики, а також моделювати різноманітні взаємодії між об'єктами.

Одним із ключових аспектів фізики в UE4 є система Rigid Body Physics, що забезпечує симуляцію руху твердих об'єктів. У цій системі кожен об'єкт має свої фізичні параметри, такі як маса, тертя та пружність. Наприклад, об'єкт із великою масою буде падати швидше або завдавати більшої шкоди при зіткненні, тоді як низький коефіцієнт тертя дозволить йому легко ковзати поверхнею. Фізичні матеріали, які можна застосувати до об'єктів, додають додаткового реалізму. Наприклад, дерево матиме високий коефіцієнт тертя, тоді як метал буде більш ковзким, а камінь — міцнішим. Завдяки цьому взаємодії між об'єктами виглядають природно.

Ще однією важливою складовою є Cloth Simulation — система, яка дозволяє моделювати поведінку тканин. Це може бути одяг персонажів, рухомі прапори

або навіть штори, що коливаються на вітрі. Система підтримує широкий набір параметрів для налаштування, включаючи жорсткість, гнучкість і взаємодію з іншими об'єктами чи силами, наприклад із вітром. Наприклад, у грі можна створити персонажа з плащем, який реалістично рухається при бігу або змінює форму під впливом вітру. Крім того, використання "скелетів" дозволяє прив'язати тканину до конкретних частин тіла персонажа, що забезпечує гармонійність рухів.

Симуляція рідин та частинок також є важливим компонентом Unreal Engine 4. Інструмент Cascade Particle Editor дозволяє створювати візуальні ефекти, такі як дим, вогонь, дощ або пил. Наприклад, у сцені битви можна додати вогонь із реалістичними язиками полум'я або густий дим, який змінює форму залежно від напрямку вітру. Для більш складних і динамічних ефектів використовується система Niagara, яка є сучасною альтернативою Cascade. Niagara дозволяє створювати ефекти з інтерактивною поведінкою, наприклад, частинки, що реагують на рух персонажа або змінюють форму під час зіткнення з об'єктами.

Система руйнувань, або Chaos Destruction System, є ще однією унікальною можливістю Unreal Engine 4. Ця технологія дозволяє створювати реалістичне руйнування об'єктів у реальному часі. Наприклад, у грі з бойовими сценами гравець може зруйнувати стіну, побачивши, як її уламки падають і взаємодіють із землею чи іншими об'єктами. Розробник має змогу налаштувати рівень деталізації руйнувань: від великих фрагментів до дрібного пилу, залежно від потреб проекту та продуктивності апаратного забезпечення.

Ще однією важливою функцією є Physics Constraints, які використовуються для створення механізмів і складних систем руху. Це може бути модель дверей, що обертаються на петлях, або підвісний міст, який коливається під вагою персонажа. Constraints також застосовуються для створення важелів, ланцюгів або навіть автомобільних механізмів, що робить їх корисним інструментом для розробки інженерних чи пригодницьких ігор. Наприклад, у грі можна створити головоломку, де важіль приводить у рух механізм, відкриваючи шлях для гравця.

Усі ці фізичні системи дозволяють розробникам створювати більш захоплюючі ігрові світи, які виглядають і поведуться реалістично. Завдяки

гнучкості налаштувань і інтеграції з іншими інструментами Unreal Engine 4 стає потужним засобом для реалізації найсміливіших ідей. Незалежно від того, чи це сцена епічної битви з масштабними руйнуваннями, чи детальний симулятор одягу, фізичні можливості UE4 забезпечують необхідну глибину ігрового процесу.

Реалістичний рендеринг у Unreal Engine 4 (UE4) є ключовою складовою для створення графіки кінематографічного рівня, яка може задовольнити найвищі вимоги сучасної ігрової індустрії. Завдяки використанню Physically Based Rendering (PBR) і сучасних алгоритмів освітлення UE4 забезпечує точну імітацію фізичних властивостей світла і його взаємодії з матеріалами віртуального світу.

Технологія PBR (Physically Based Rendering) є основою системи рендерингу в UE4. Вона дозволяє створювати матеріали, які реалістично реагують на освітлення незалежно від умов сцени. Основним підходом у PBR є використання текстурних карт для моделювання властивостей матеріалу. Наприклад, карта Base Color визначає базовий колір поверхні, тоді як Roughness відповідає за шорсткість поверхні, що впливає на те, наскільки розсіяними будуть відбиття світла. Карта Metallic задає металеві властивості, імітуючи, як поверхня відбиває світло, властиве металам. Завдяки цим картам розробники можуть створювати реалістичні матеріали, такі як метал, дерево, тканина чи скло, які автоматично адаптуються до різних умов освітлення.

Одним із найбільших досягнень UE4 є підтримка динамічного освітлення. Ця функція дозволяє світловим джерелам змінювати свій вплив на сцену залежно від руху об'єктів чи зміни положення камери. Розробники мають у своєму розпорядженні декілька типів джерел світла, таких як Directional Light для імітації сонця, Point Light для точкових джерел (наприклад, лампочки), і Spot Light для спрямованого світла, як у прожектора. Для створення драматичних сцен із промінням світла у повітрі використовується Volumetric Lighting, який додає об'єму освітленню, забезпечуючи ефект реальних світлових променів.

Ще однією важливою складовою рендерингу є глобальне освітлення (Global Illumination). Цей підхід моделює, як світло відбивається від поверхонь,

освітлюючи інші об'єкти в сцені. Unreal Engine 4 використовує технологію Lightmass, яка попередньо обчислює глобальне освітлення, щоб забезпечити точну імітацію відбиттів світла. Наприклад, якщо у сцені світло падає на стіну, то його відбиття буде впливати на кольори і освітлення сусідніх об'єктів, додаючи сцени глибини та реалістичності.

Screen Space Reflections (SSR) є ще однією потужною технологією, яка використовується в UE4. Вона забезпечує створення реалістичних відбиттів на поверхнях, таких як вода, скло чи металеві предмети. Завдяки SSR навіть невеликі деталі, такі як краплі дощу на вікні чи відблиски на шоломі персонажа, додають реалістичності ігровим сценам. Хоча SSR вимагає порівняно небагато ресурсів, воно має певні обмеження — наприклад, працює лише для об'єктів, видимих у кадрі.

Для створення графіки найвищого рівня Unreal Engine 4 підтримує технологію Ray Tracing, яка є проривом у симуляції освітлення. Трасування променів дозволяє обчислювати, як світло рухається у просторі, враховуючи його відбиття, заломлення та розсіювання. Це забезпечує максимально реалістичну імітацію взаємодії світла з об'єктами. Наприклад, прозорі матеріали, такі як скло, будуть коректно заломлювати світло, створюючи ефекти, схожі на ті, що ми спостерігаємо в реальному житті. Ray Tracing також дозволяє створювати високоякісні тіні, які точно відображають складну геометрію сцени.

Комбінація цих технологій забезпечує потужний набір інструментів для досягнення вражаючої графіки. Наприклад, у сценах із міським пейзажем, освітленим вуличними ліхтарями, можна використовувати динамічне освітлення для створення ефекту рухомих тіней, глобальне освітлення для імітації відбиттів від фасадів будівель і Ray Tracing для деталізації відбитків у калюжах на дорозі. Усе це працює в унісон, забезпечуючи максимально реалістичний візуальний досвід.

Таким чином, реалістичний рендеринг в Unreal Engine 4 — це не просто набір окремих технологій, а інтегрована система, яка дозволяє розробникам створювати віртуальні світи з неперевершеною деталізацією і реалістичністю.

Завдяки таким функціям, як PBR, глобальне освітлення, SSR і Ray Tracing, Unreal Engine 4 стає вибором номер один для розробників, які прагнуть досягти кінематографічної якості графіки.

Поєднання фізичних симуляцій із реалістичним рендерингом дає змогу створювати захопливі ігрові сцени. Наприклад:

- Падіння об'єктів із реалістичним рухом і тіннями.
- Відбиття світла від розбитого скла або калюж.
- Симуляція погоди, такої як дощ із фізичними краплями та реалістичним освітленням.

Переваги використання Unreal Engine 4:

1. Продуктивність. Навіть складні симуляції та рендеринг добре оптимізовані для сучасного обладнання.
2. Інтеграція технологій. UE4 підтримує всі етапи: від фізичних ефектів до рендерингу, забезпечуючи безперервний робочий процес.
3. Кросплатформеність. Фізичні симуляції та візуальні ефекти можуть працювати на різних платформах — ПК, консолях та мобільних пристроях.
4. Інтерактивність. Усі ефекти можуть реагувати на дії гравця, що підвищує рівень занурення.

Unreal Engine 4 надає потужні інструменти для реалізації фізичних симуляцій і реалістичного рендерингу. Ці технології забезпечують глибину, реалістичність та інтерактивність сучасних 3D-середовищ у відеоіграх. Поєднання фізики та PBR-матеріалів робить Unreal Engine 4 одним із найкращих виборів для розробників, які прагнуть створювати унікальний ігровий досвід.

2.4 Підхід до оптимізації 3D-середовищ

Оптимізація є ключовим етапом розробки 3D-середовищ у відеоіграх. Вона забезпечує плавну роботу гри на різних платформах, знижує споживання ресурсів і гарантує високу продуктивність навіть у великих, складних сценах. Unreal

Engine 4 (UE4) надає розробникам широкий набір інструментів та методів для досягнення оптимального балансу між якістю графіки та продуктивністю.

Оптимізація 3D-сцен у процесі створення інтерактивних середовищ є ключовим аспектом, що дозволяє забезпечити високу продуктивність і якість графіки. Unreal Engine 4 надає широкий набір інструментів та методів для оптимізації, які допомагають знизити навантаження на апаратне забезпечення без втрати візуальної якості. Одним із найважливіших аспектів є зменшення кількості полігонів у моделях, адже їх надмірна кількість може суттєво вплинути на продуктивність. Полігони формують основу будь-якого 3D-об'єкта, тому оптимізація полігонального навантаження є базовою задачею для розробників. Застосування технології Levels of Detail (LOD) дозволяє створювати кілька версій моделі з різною кількістю полігонів. Це дає змогу відображати об'єкти з високою деталізацією, коли вони знаходяться близько до камери, і використовувати спрощені моделі на віддалених ділянках сцени. Інструмент Simplygon, інтегрований у UE4, автоматизує процес генерації LOD, що суттєво скорочує час, необхідний для підготовки активів.

Ще одним важливим кроком в оптимізації є зменшення ваги і роздільної здатності текстур, адже текстури займають значну частину пам'яті графічного процесора та впливають на час завантаження сцени. Завдяки технології Texture Streaming, Unreal Engine 4 завантажує лише ті текстури, які потрапляють у поле зору камери. Це знижує кількість ресурсів, що використовуються одночасно. Для далеких об'єктів розробники можуть застосовувати текстури із пониженим розширенням, зберігаючи при цьому візуальну привабливість. Компресія текстур із використанням сучасних форматів, таких як DXT або ASTC, зменшує розмір файлів, не погіршуючи якість зображення, що значно полегшує завантаження великих сцен.

Матеріали, які визначають зовнішній вигляд поверхонь об'єктів, також впливають на продуктивність. У кожній моделі може бути кілька матеріалів, і кожен із них вимагає окремого виклику до графічного процесора, що збільшує навантаження на систему. Щоб уникнути цього, розробники можуть

використовувати універсальні матеріали, які дозволяють налаштовувати параметри для різних об'єктів, зменшуючи кількість унікальних матеріалів. Ще один метод — об'єднання текстур у атласи текстур (texture atlases), що дозволяє зберігати кілька текстур в одному файлі. Це зменшує кількість викликів і підвищує продуктивність, особливо в складних сценах із великою кількістю дрібних об'єктів.

Реалістичне освітлення та тіні є важливими елементами, але вони також можуть бути надзвичайно ресурсомісткими. UE4 пропонує різні підходи для оптимізації цих процесів. Наприклад, для нерухомих об'єктів освітлення може бути розраховане заздалегідь за допомогою Static Lighting, що значно знижує навантаження на систему під час гри. Для рухомих об'єктів можна налаштувати якість і радіус динамічних тіней, що допомагає знайти баланс між реалістичністю та продуктивністю. Використання Lightmass дозволяє попередньо обчислювати глобальне освітлення, створюючи м'які й природні тіні без потреби в реальному часі обробляти складні світлові взаємодії.

Зменшення кількості викликів до графічного процесора (Draw Calls) є ще одним критичним аспектом оптимізації. Кожен об'єкт сцени вимагає окремого виклику для відображення, тому об'єднання сусідніх об'єктів у єдині групи може суттєво скоротити кількість таких викликів. Для великих сцен із повторюваними елементами, такими як дерева, кущі чи будівельні блоки, використовується Instanced Static Meshes, що дозволяє створювати копії об'єктів із мінімальними витратами ресурсів. Інструменти, такі як Frustum Culling, автоматично вимикають об'єкти, які знаходяться поза полем зору камери, забезпечуючи, що ресурси витрачаються лише на те, що бачить гравець.

Оптимізація є невіддільною частиною процесу створення 3D-середовищ, адже навіть найкрасивіша сцена втрачає свою цінність, якщо вона погано працює на кінцевих пристроях. Грамотне використання полігонів, текстур, матеріалів і освітлення допомагає досягти найвищої продуктивності без жодних помітних втрат у якості. Завдяки потужним інструментам і методам Unreal Engine 4

розробники мають усе необхідне для створення високоякісних, оптимізованих сцен, які працюють плавно навіть на пристроях із обмеженими ресурсами.

Оптимізація фізичних симуляцій у середовищі Unreal Engine 4 є ключовим завданням для створення продуктивних ігор, що вимагають реалістичного моделювання поведінки об'єктів. Використання фізичних симуляцій дозволяє розробникам створювати динамічний і інтерактивний світ, однак ці симуляції можуть значно збільшувати навантаження на апаратне забезпечення, особливо якщо вони не оптимізовані. Одним із перших кроків у цьому напрямі є вимкнення фізики для об'єктів, які не потребують динамічної взаємодії. Наприклад, статичні об'єкти, які лише створюють декорації, не повинні оброблятися фізичним рушієм, що дозволяє звільнити ресурси для інших завдань. Unreal Engine надає Collision Presets, які допомагають налаштувати рівень фізичної взаємодії. Це дозволяє використовувати більш спрощені моделі зіткнень, наприклад, прості коробки або сфери, замість складних багатокутникових оболонок, які потребують значно більше обчислювальних потужностей.

Симуляції частинок, такі як вогонь, дим або вода, додають сцені візуальної привабливості, але можуть швидко перевантажити систему, особливо якщо на екрані відображається велика кількість частинок. Для їхньої оптимізації слід обмежувати кількість частинок, які відображаються одночасно, і зменшувати їхню тривалість у кадрі. Наприклад, для віддалених об'єктів можна використовувати менш деталізовані системи частинок, оскільки дрібні деталі все одно будуть непомітні. В Unreal Engine сучасна система Niagara дозволяє створювати і налаштовувати такі ефекти з високим ступенем контролю, забезпечуючи ідеальний баланс між якістю та продуктивністю. Cascade, як попередник Niagara, також залишається корисним інструментом для швидких і легких ефектів, але вимагає обережного налаштування параметрів, щоб уникнути перевантаження системи.

У багатьох ігрових проектах динамічна фізика не завжди потрібна для всіх об'єктів. Наприклад, якщо об'єкт не змінює своє положення або властивості в реальному часі, його фізична поведінка може бути попередньо розрахована. Цей

підхід дозволяє знижувати навантаження на процесор і графічний чіп, оскільки всі складні обчислення виконуються заздалегідь. Для таких об'єктів Unreal Engine використовує технології попереднього розрахунку фізики (Precomputed Physics), що є особливо корисним для великих сцен із численними нерухомими об'єктами. Це забезпечує баланс між реалістичністю та продуктивністю, дозволяючи зосередити ресурси на тих аспектах, які мають більший вплив на ігровий процес.

Ще одним ефективним інструментом оптимізації фізичних симуляцій є впровадження групування фізичних об'єктів. Якщо кілька об'єктів мають схожу поведінку, вони можуть бути об'єднані в одну симуляцію, що значно скорочує кількість обчислень. Наприклад, уламки, які залишаються після вибуху, можуть бути згруповані в єдиний фізичний об'єкт після певного часу. Це зменшує кількість індивідуальних об'єктів, що перебувають у симуляції, і, відповідно, знижує навантаження.

Оптимізація також охоплює фізичні матеріали, які визначають, як об'єкти взаємодіють один з одним. Замість того, щоб кожен об'єкт мав індивідуальний фізичний матеріал, можна використовувати загальні налаштування для об'єктів зі схожими властивостями. Це дозволяє зменшити кількість унікальних матеріалів, які потрібно обробляти рушієм, що позитивно впливає на продуктивність. Крім того, варто враховувати, що надмірна деталізація фізичних властивостей, таких як тертя або пружність, для об'єктів, які не беруть участі в ключових взаємодіях, є зайвою і може бути зведена до мінімуму.

Для великих ігор із відкритим світом, де фізичні взаємодії можуть включати сотні об'єктів, важливо застосовувати відсікання фізичних взаємодій за межами поля зору гравця. Цей метод дозволяє виключати з обробки об'єкти, які знаходяться далеко від камери або приховані іншими об'єктами. Це значно знижує обчислювальне навантаження і допомагає підтримувати стабільну частоту кадрів навіть у найбільш насичених сценах.

Розумне налаштування і використання доступних інструментів оптимізації в Unreal Engine 4 дозволяє досягти балансу між реалістичністю і продуктивністю фізичних симуляцій. Правильний підхід до обробки фізики не тільки покращує

візуальне сприйняття гри, але й забезпечує плавну роботу на різних платформах, від персональних комп'ютерів до консолей і мобільних пристроїв.

Використання профілювання

Unreal Engine 4 має вбудовані інструменти для аналізу продуктивності:

- Stat Unit — вимірювання часу на обробку кадру, рендеринг і оновлення ігрової логіки.
- GPU Profiler — аналіз навантаження на графічний процесор.
- Performance Insights — звіт про ресурси, які споживаються кожним компонентом сцени.

Оптимізація звуку та анімацій в Unreal Engine 4 є важливою складовою загальної продуктивності гри. Система звуку відіграє ключову роль у створенні реалістичного ігрового середовища, однак без належної оптимізації вона може стати джерелом значного навантаження на ресурси. Використання стиснутих аудіофайлів, таких як OGG або MP3, дозволяє суттєво зменшити розмір файлів без помітної втрати якості. Це особливо важливо для мобільних платформ, де обсяг пам'яті обмежений. Додатково слід активувати просторовий звук (spatial audio), щоб звук оброблявся лише для тих джерел, які знаходяться у безпосередній близькості від гравця. Цей підхід знижує кількість аудіоканалів, які одночасно використовуються, і створює більш реалістичну звукову картину.

Щодо анімацій, значна частина оптимізації зосереджена на зменшенні розміру файлів та обчислювальних ресурсів, необхідних для їхнього відтворення. Animation Compression є одним із головних інструментів Unreal Engine, що дозволяє зменшити обсяг анімаційних даних. Компресія забезпечує збереження основних деталей рухів, водночас суттєво скорочуючи розмір файлів. Також важливо уникати дублювання анімацій шляхом об'єднання схожих рухів. Це дозволяє не лише зменшити обсяг даних, але й спрощує процес їхнього управління. У разі великих ігрових проектів слід використовувати Animation Blueprints, які забезпечують більш ефективне використання ресурсів, дозволяючи одночасно відтворювати кілька анімацій для різних персонажів.

При розробці ігор, орієнтованих на різні платформи, оптимізація повинна враховувати специфіку кожної з них. На ПК і консолях, які зазвичай мають потужніші апаратні характеристики, можна дозволити використання високоякісних текстур і складних шейдерів. Це забезпечує вищий рівень деталізації і загальної візуальної якості. Крім того, підтримка багатопоточності дозволяє ефективніше використовувати всі доступні ядра процесора, прискорюючи обробку даних та відтворення складних сцен.

Для мобільних пристроїв, обмежених у ресурсах, підхід до оптимізації є більш консервативним. Наприклад, використання текстур із меншою роздільною здатністю дозволяє зекономити пам'ять і знизити навантаження на графічний процесор. Тіні є одним із найбільш ресурсомістких аспектів рендерингу, тому на мобільних платформах часто використовують прості тіні або повністю вимикають їх для об'єктів, які не є ключовими у сцені. Важливо також обмежити складні візуальні ефекти, такі як трасування променів, яке хоч і забезпечує високу якість освітлення і відбиттів, є надто ресурсомістким для більшості мобільних пристроїв.

У процесі оптимізації необхідно ретельно тестувати гру на цільових платформах, щоб виявити найбільш ресурсомісткі аспекти. Це дозволяє внести відповідні зміни і забезпечити стабільну частоту кадрів, навіть за умови використання обмеженого апаратного забезпечення. Unreal Engine надає широкий набір інструментів для профілювання продуктивності, які допомагають ідентифікувати «вузькі місця» у процесі розробки. Збалансований підхід до оптимізації звуку, анімацій і графіки дозволяє створювати ігри, що відповідають вимогам як високопродуктивних ПК, так і мобільних пристроїв.

Розробники багатьох успішних ігор, таких як Fortnite, PUBG та ARK: Survival Evolved, активно використовують методи оптимізації в Unreal Engine 4. Це дозволяє запускати масштабні багатокористувацькі ігри з високою якістю графіки та стабільною продуктивністю на різних платформах.

Оптимізація 3D-середовищ є важливим аспектом розробки відеоігор, що дозволяє досягти високої продуктивності без втрати якості візуалізації. Unreal

Engine 4 надає потужні інструменти для зниження навантаження на систему, що є ключовим для створення ігор із високими вимогами до графіки та фізики. Правильний підхід до оптимізації забезпечує кращий ігровий досвід і дозволяє залучити ширшу аудиторію гравців.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ ГРИ

3.1 Діаграма класів

Діаграма класів у UML слугує статичним представленням структури моделі. Вона показує декларативні елементи, такі як класи, типи даних, їхній вміст та зв'язки між ними.

На рис. 3.1 зображено діаграму класів.

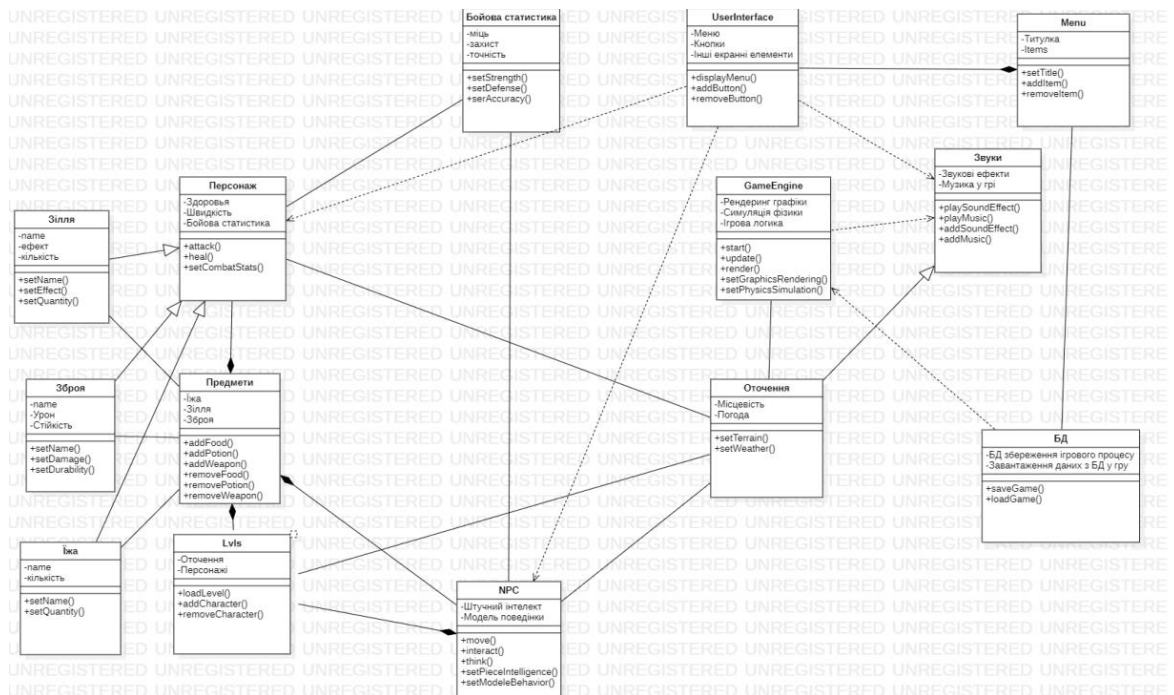


Рис. 3.1 Діаграма класів програмної реалізації

Клас Персонаж пов'язаний відношенням композиції з класом Предмети, оскільки персонаж гравця може мати в своєму інвентарі різноманітні предмети, наприклад їжу, зілля чи зброю.

Клас GameEngine має залежність від класу Звуки, оскільки рушій використовується для відтворення звукових ефектів чи музичного супроводу під час гри.

Клас UI знаходиться у відношенні композиції з класом Меню, адже інтерфейс містить різні кнопки, доступні для взаємодії гравця.

3.2 Опис функціоналу гри

Для створення головного меню в Unreal Engine 4 було використано інструмент "WB_MainMenu", створений за допомогою Unreal Editor. Цей інструмент дає змогу розробити користувацький інтерфейс для головного меню за допомогою Blueprint-інтерфейсу. Процес створення головного меню відображено на рис. 3.2.

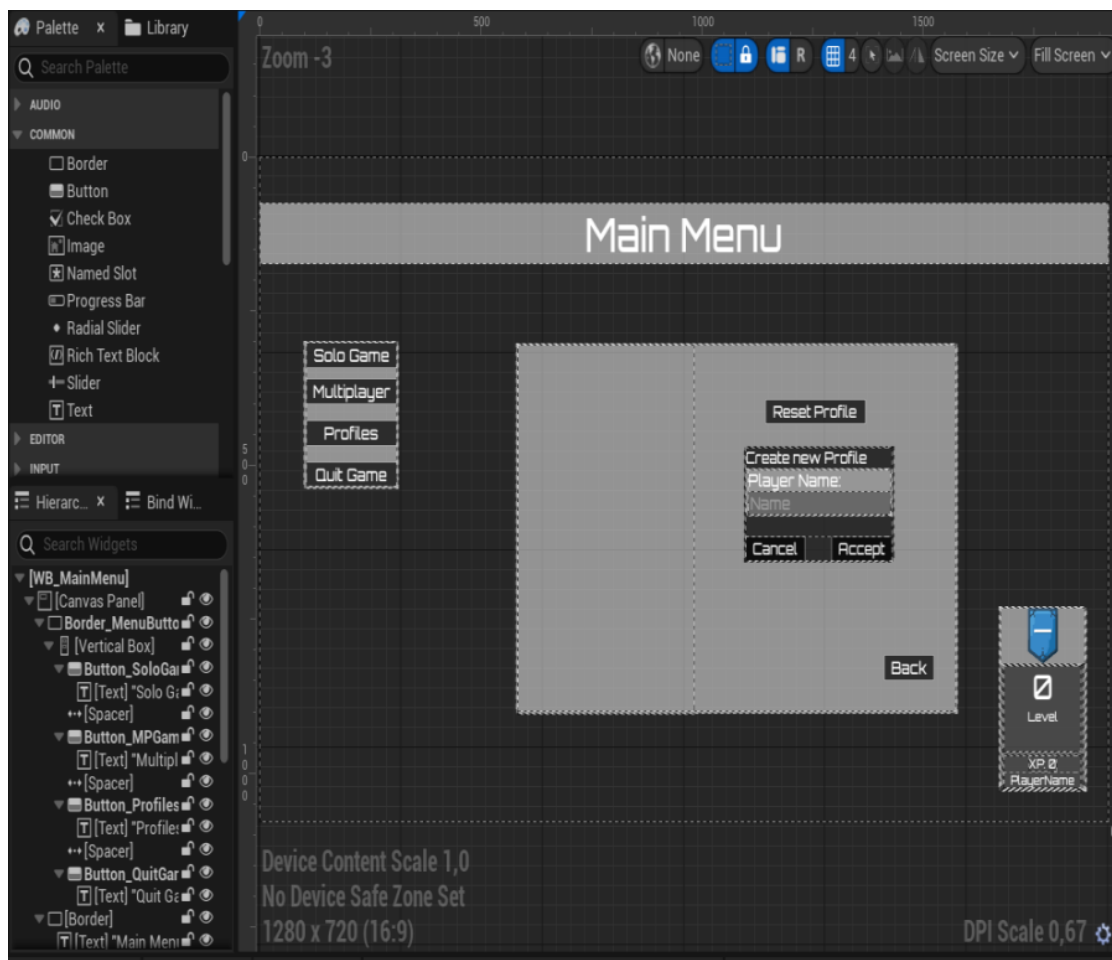


Рис. 3.2 Зображення процесу створення головного меню

Завдяки цьому класу стало можливим додавати й налаштовувати елементи інтерфейсу, такі як кнопки, текст, зображення та інші візуальні компоненти. На рис. 3.3 показано, як виглядає екран завантаження в контексті гри.



Рис. 3.3 Зображення готового інтерфейсу головного меню

Режим вибору гри створювався аналогічно до головного меню, але замість "WB_MainMenu" було застосовано "WB_GameMode", також створений через Unreal Editor. Екран завантаження для цього режиму відображено на рис. 3.4.

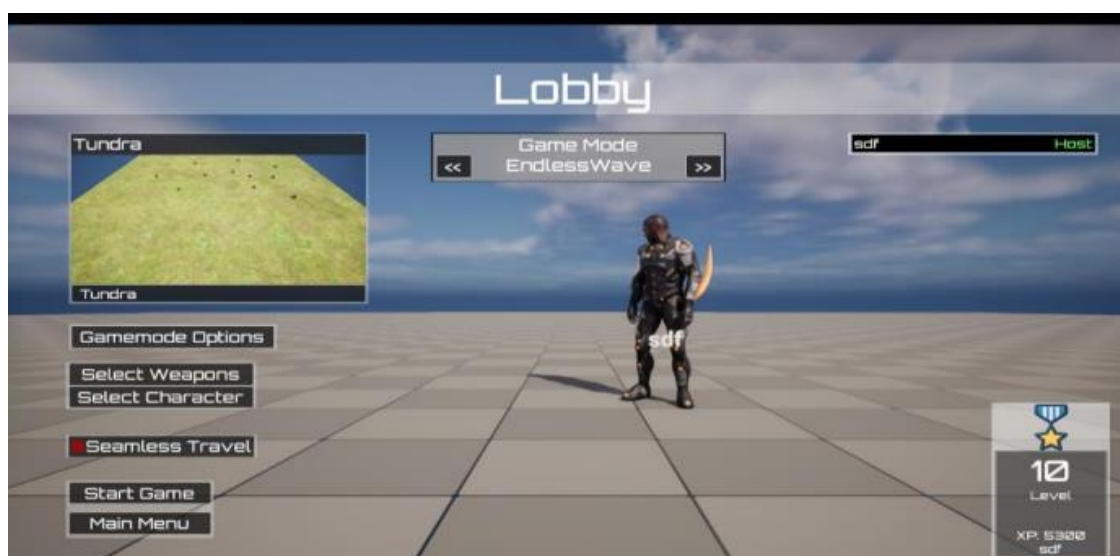


Рис. 3.4 Зображення екрану вибору ігрового режиму

Для реалізації екрану завантаження в Unreal Engine 4 було створено "WB_LoadScreen". Цей елемент дає змогу додати статичний екран завантаження без складних анімаційних ефектів, що позитивно впливає на швидкість завантаження гри. Розробку екрану завантаження представлено на рис. 3.5.

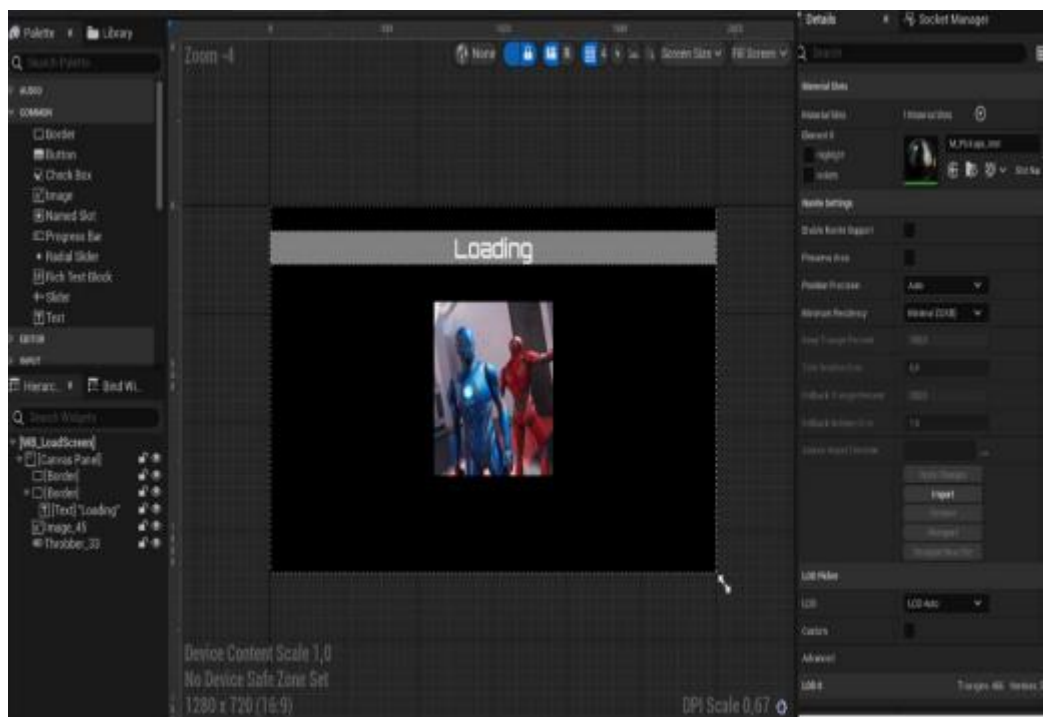


Рис. 3.5 Зображення процесу створення екрану завантаження

"WB_LoadScreen" дозволяє використовувати статичні зображення чи логотипи як екран завантаження, що забезпечує можливість брендування або передачі спеціального повідомлення для гравців. Завдяки цьому інструменту Unreal Engine 4 забезпечує оптимізовану швидкість завантаження, що є важливим аспектом для задоволення аудиторії. Використання таких функцій дозволяє поліпшити сприйняття гри шляхом оптимізації екранів завантаження.

Для створення ігрової місцевості в Unreal Engine 4 було використано інструменти Landscape Tool та Level Editor. Landscape Tool дає змогу швидко створювати рельєф та деталі поверхні, а Level Editor — розміщувати, масштабувати й налаштовувати об'єкти у грі. Приклад розробки місцевості показано на рис. 3.6.



Рис. 3.6 Зображення етапу розробки ігрової місцевості

Додатково Unreal Engine 4 пропонує широкі можливості для налаштування освітлення та створення атмосферних ефектів. Система динамічного освітлення дозволяє розробляти реалістичне освітлення з різними джерелами світла. На рис. 3.7 показано, як виглядає ігрова місцевість у фінальній версії гри.



Рис. 3.7 Зображення фінальної версії ігрової місцевості

Для створення ігрової зброї було використано готові моделі з бібліотеки Unreal Engine. Після імпорту зброї додано компонент Static Mesh Component, що дозволяє прив'язати модель до об'єкта й використовувати її у грі [5]. Зовнішній вигляд зброї представлено на рис. 3.8.



Рис. 3.8 Зображення моделі ігрової зброї

Схожий підхід застосовано для створення моделей боєзапасу. Імпортовані моделі було прив'язано до об'єктів боєприпасів за допомогою mesh-компонента. Як виглядає боєзапас у грі, зображено на рис. 3.9.



Рис. 3.9 Зображення процесу розробки боєзапасу

Механіка поповнення боєзапасу реалізована через програмування в Blueprint. Для цього створено атрибут, що відповідає за кількість набоїв або боєприпасів у гравця, а також передбачено їх початкову ініціалізацію.

3.3 Механіка пересування персонажа

Процес створення персонажа в Unreal Engine 4 включає кілька етапів, що дозволяють створити якісну та реалістичну модель. Основні стадії моделювання персонажа охоплюють:

Створення моделі. Першим кроком є створення тривимірної моделі персонажа за допомогою сторонніх 3D-редакторів, таких як Blender або 3ds Max. Модель має бути виконана з дотриманням реалістичних анатомічних пропорцій та деталізації.

UV-розгортка. Після створення моделі виконується розгортка UV-координат. Цей процес дозволяє коректно наносити текстури на поверхню моделі без спотворень, забезпечуючи точне відображення текстур на персонажі.

Текстурування. Після завершення розгортки створюються текстури, які додають моделі деталізації та реалістичності. Це можуть бути текстури кольору (алbedo), карти нормалей, матеріальні карти та інші.

Риггінг. На цьому етапі створюється скелетна структура персонажа, яка забезпечує можливість анімації. Скелет складається з кісток і суглобів, які дозволяють задавати рухи.

Анімація. Наступним кроком є створення анімацій. Для цього використовуються методи захоплення руху або створення ключових кадрів. Цей процес додає персонажу рухливості та динамічності.

Імпорт в Unreal Engine 4. Завершальним етапом є імпорт створеної моделі, текстур і анімацій в Unreal Engine 4. У середовищі Unreal Engine, зокрема за допомогою інструментів Persona або Animation Blueprint, розробники можуть налаштовувати анімації та інтерактивність персонажа.

Дотримуючись цих етапів, можна створити реалістичну модель персонажа з повноцінною анімацією для використання в ігрових проектах. На рис. 3.10 зображено процес редагування імпортованого персонажа в Unreal Engine 4.

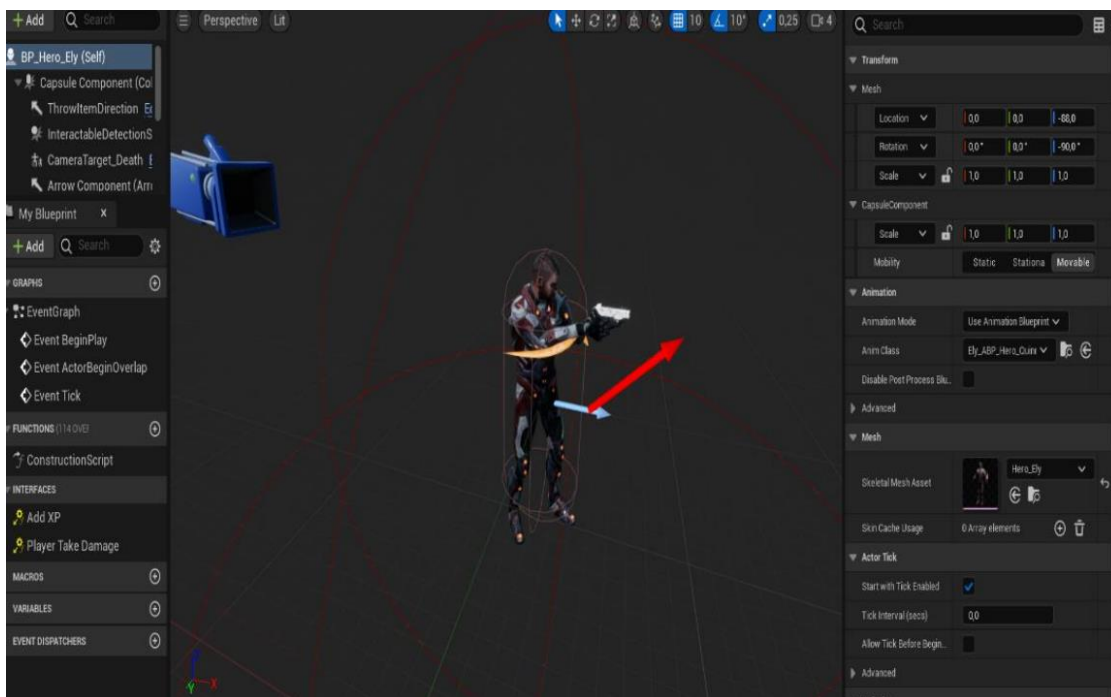


Рис. 3.10 Зображення персонажа в Unreal Engine 4

Розглянемо функції в Unreal Engine 4.

1. IA_Move. Функція відповідає за управління пересуванням персонажів із застосуванням штучного інтелекту (AI). Вона задає траєкторії, напрямки руху та швидкість пересування персонажа, що керується AI.

2. IA_Look. Ця функція використовується для визначення напрямку погляду AI-персонажа. Завдяки ній можна спрямувати персонажа на ціль, виявити супротивника або оглянути навколишнє середовище.

3. IA_ADS. Функція активує або вимикає режим прицілювання для AI-персонажа. У цьому режимі точність стрільби значно підвищується, що дозволяє більш ефективно вести бойові дії.

4. IA_Crouch. Дана функція дозволяє AI-персонажу приймати положення сидючи. Це корисно для маскування, зменшення видимості або отримання тактичної переваги в бою.

5. IA_Dash. Функція забезпечує швидкий ривок персонажа в заданому напрямку. Вона може бути використана для ухилення від атак супротивника або оперативного переміщення по ігровій карті.

6. IA_FireWeapon. Ця функція активує використання зброї AI-персонажем. Вона визначає напрямок та ціль для стрільби, забезпечуючи реалізацію бойових дій.

7. IA_Interact. Функція дозволяє взаємодіяти з об'єктами чи елементами середовища. AI може активувати механізми, використовувати важелі або виконувати інші дії.

8. IA_Jump. Ця функція дозволяє AI здійснювати стрибки. Вона застосовується для подолання перешкод або переходу на вищі рівні місцевості.

9. IA_Melee. Функція активує ближній бій, забезпечуючи атаку AI-персонажа в безпосередній близькості до супротивника.

10. IA_PauseGame. Функція призначена для призупинення гри. Її можна використовувати для виклику паузи або відображення меню налаштувань.

11. IA_Reload. Ця функція забезпечує перезарядку зброї AI-персонажа. Вона відновлює боезапаси та підготовлює зброю до наступного використання.

12. IA_SwitchShould. Функція дозволяє змінювати плечі, з яких персонаж веде стрільбу. Це впливає на комфортність управління та точність атаки.

13. IA_SwitchWeapon. Ця функція дозволяє AI-персонажу змінювати зброю. Завдяки їй можна перемикатися між доступними типами озброєння та боєприпасів.

14. IA_ThrowGrenade. Функція активує кидок гранати персонажем. Вона задає напрямок і дальність польоту гранати.

15. IA_ToggleInventory. Функція відкриває або закриває інвентар персонажа. Це дозволяє переглядати та використовувати предмети, а також керувати ресурсами.

16. IA_ToggleScoreBoard. Ця функція відповідає за відображення таблиці результатів. Вона дозволяє оцінювати статистику гри та аналізувати успішність.

На рис 3.11 представлено процес налаштування функцій для управління пересуванням.

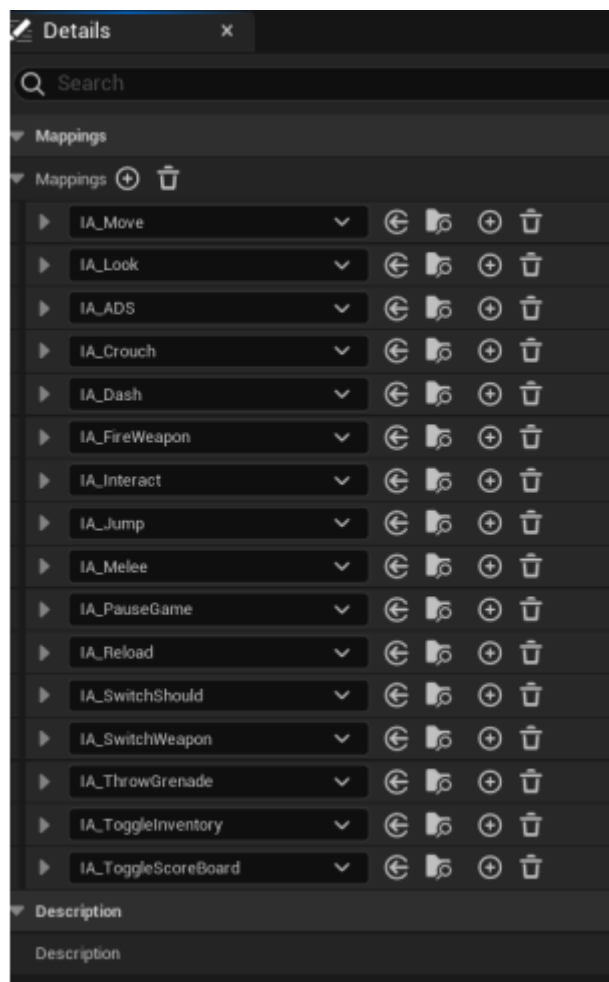


Рис. 3.11 Зображення розробки команд

Ці функції надають можливість ефективно контролювати та налаштовувати поведінку персонажа з елементами штучного інтелекту (AI) у грі. Завдяки їх використанню вдалося реалізувати складні й реалістичні механіки руху персонажа, включаючи нестандартні варіанти пересування, наприклад, у положенні сидячи .

На рис. 3.12 зображено етап розробки механіки пересування персонажа в режимі присядки.

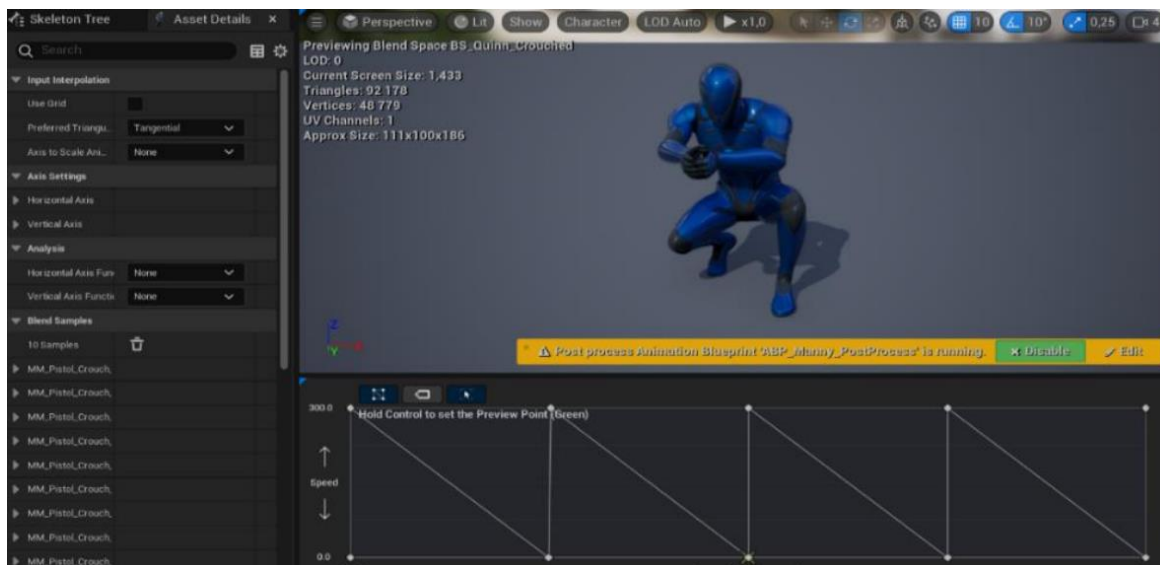


Рис. 3.12 Зображення персонажу у положенні присіду

3.4 Алгоритм пошуку шляху

Для реалізації функції пошуку локації гравця в грі, створеній на Unreal Engine 4, використовувалися скрипти Blueprint, зокрема компонент Structure. Завдяки цьому механізму гра автоматично визначає місце розташування персонажа за векторними координатами та переміщує його у відповідну позицію на екрані. Такий підхід значно спростив процес розробки, оскільки відпала необхідність вручну програмувати кожний крок руху персонажа. Розробнику достатньо лише задати нові координати, і гра самостійно виконує переміщення.

Цей метод також забезпечує плавність руху персонажа, використовуючи алгоритми інтерполяції для згладжування переходів між точками. У проекті було застосовано лінійну інтерполяцію для прямолінійного переміщення, а для зміни швидкості використовували квадратичну інтерполяцію.

Окрім переміщення персонажа, аналогічний механізм використовувався для керування рухом камери. У цьому випадку задаються координати місцезнаходження камери, після чого гра автоматично забезпечує її переміщення до заданої позиції.

3.5 Створення мобів

У Unreal Engine 4 створення мобів здійснюється через систему Blueprint, яка дозволяє задавати логіку їхньої поведінки та налаштовувати анімацію для гри.

Для моба у грі було розроблено анімації атак. У Blueprint Editor були додані анімаційні вузли для здійснення ударів правою та лівою руками. За допомогою інструменту AnimNotify було створено сповіщення AnimNotify_Sword_R_Start, яке сигналізує про початок анімації атаки правою рукою з використанням меча. Додаткова функція Set дозволяє налаштувати моба, встановлюючи зв'язок із персонажем гравця та активуючи можливість завдання шкоди. Аналогічно реалізовано атаку лівою рукою через AnimNotify_Sword_L_Start, яка також використовує функцію Set для нанесення шкоди та переслідування гравця.

Після завершення атаки правою рукою активується AnimNotify_Sword_R_End, що завершує дію. Для лівої руки реалізовано такий самий підхід. Щоб забезпечити можливість чергування атак руками, було додано анімаційне сповіщення AnimNotify_StartRagdoll. У разі її активації під час анімації об'єкт переходить у режим "Ragdoll", у якому фізична симуляція бере на себе контроль над об'єктом. Це забезпечує реалістичну фізичну поведінку моба, дозволяючи йому взаємодіяти з середовищем відповідно до фізичних сил.

На рис. 3.15 продемонстровано процес створення системи атак моба.

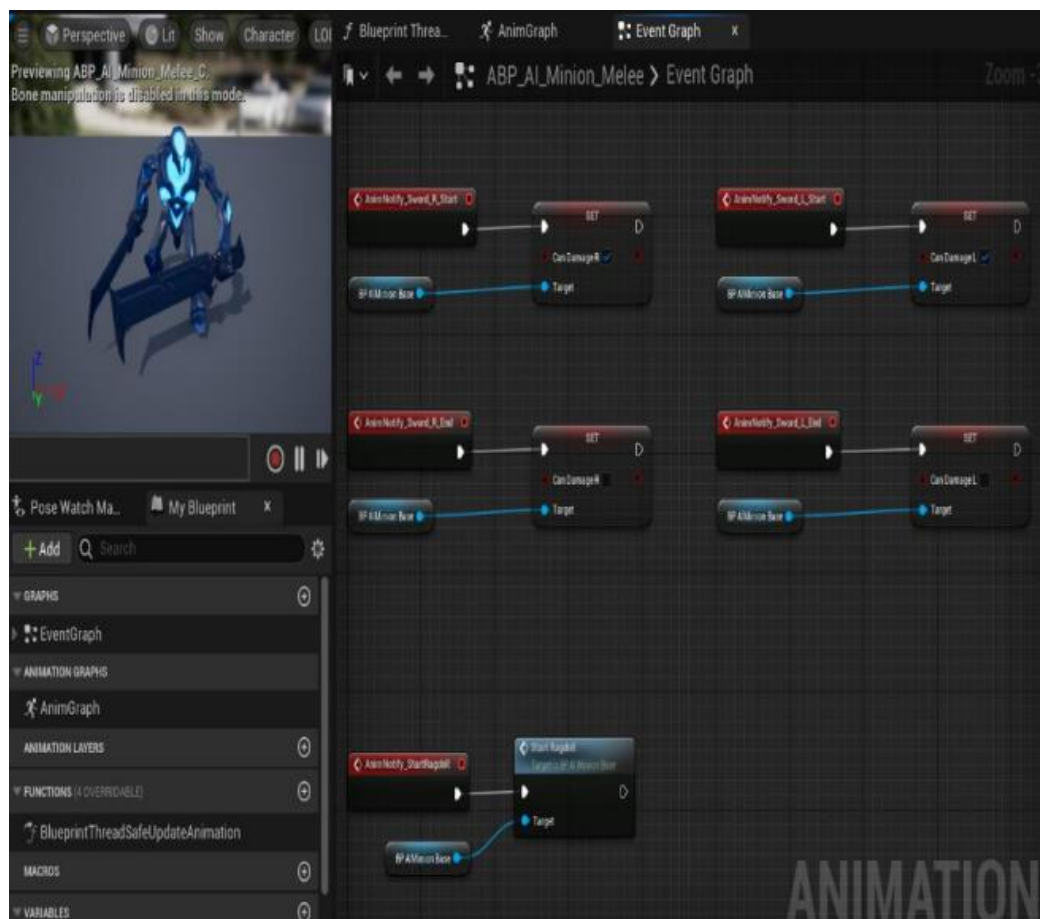


Рис. 3.15 Зображення роботи системи атака

Для створення анімацій у грі було використано систему Blueprint, яка дозволяє реалізовувати управління об'єктами та впроваджувати різноманітні механіки. Один із важливих аспектів процесу — забезпечення безпечного виконання коду за допомогою Thread Safe, що гарантує стабільність під час роботи кількох потоків одночасно, запобігаючи пошкодженню даних і конфліктам.

Для оновлення анімацій використовується механізм Update Animation, який відіграє ключову роль у створенні динаміки персонажів, об'єктів і середовища. Цей процес забезпечує зміну властивостей або станів анімацій з часом, що дозволяє створювати плавний рух і візуальні ефекти.

У Blueprint задіяно вузол Set, який встановлює значення швидкості (Speed), необхідне для контролю темпу відтворення анімації. Функція TryGetPawnOwner.GetVelocity використовується для визначення вектора

швидкості моба, що дозволяє враховувати напрямок і величину його руху. При цьому Vector Length XY обчислює довжину вектора, беручи до уваги лише компоненти X та Y, що дозволяє аналізувати горизонтальні переміщення.

На рис. 3.16 продемонстровано процес створення анімації моба.

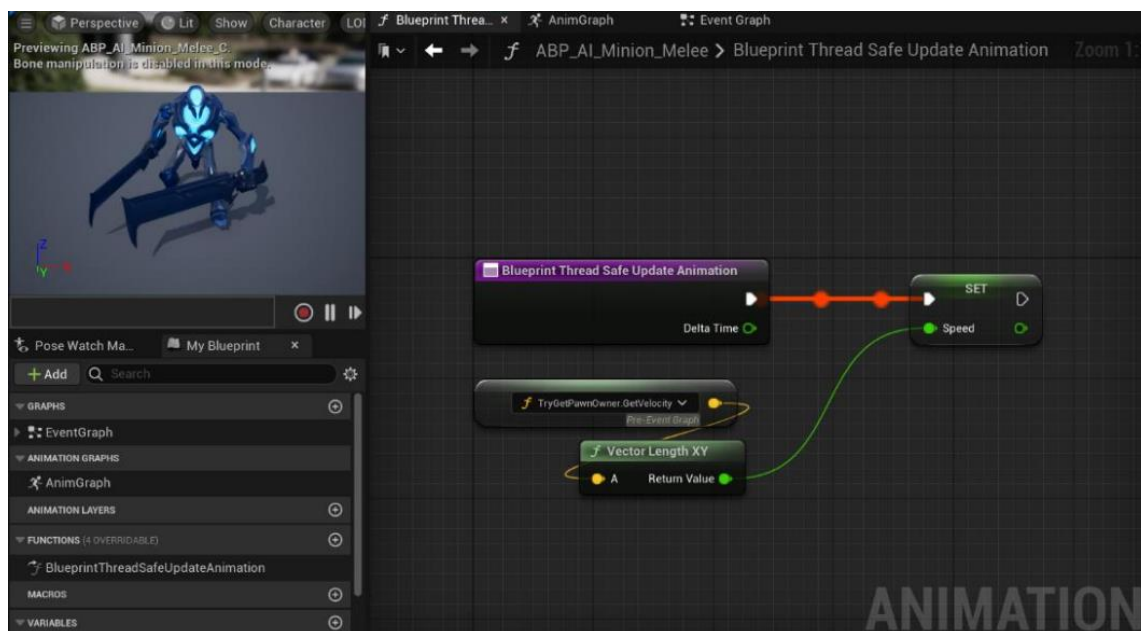


Рис. 3.16 Зображення розробки анімації

3.6 Розробка механік взаємодії

Health: Цей параметр дозволяє контролювати і регулювати рівень здоров'я персонажа. Його застосовують для відображення індикатора життя, виконання певних дій при зміні рівня здоров'я або обробки ситуацій, пов'язаних із критичним пошкодженням чи смертельними ударами.

Strength: Цей показник використовується для моніторингу та налаштування сили персонажа. Від нього залежить ефективність атак, здатність піднімати важкі предмети та виконувати інші фізичні дії, що вимагають значних зусиль.

Death_Front, Death_Back, Death_Left, Death_Right: Ці команди забезпечують анімацію для різних варіантів смерті персонажа залежно від напрямку удару. Вони включають падіння, відповідні рухи тіла та аудіоефекти, які підсилюють атмосферу гри.

Hit_Front, Hit_Back, Hit_Left, Hit_Right: Ці параметри відповідають за анімації реакції на удари з різних боків. Вони можуть включати рухи, що демонструють отримання пошкодження, і звукові ефекти, що підкреслюють цей момент.

CrouchEntry, CrouchExit: Ці команди керують анімаціями входу в режим присідання і виходу з нього. Вони забезпечують плавний перехід між положеннями, включаючи згинання ніг, зміну висоти персонажа та корекцію швидкості його руху.

Stamina: Ця команда контролює рівень витривалості персонажа. Вона визначає тривалість і інтенсивність фізичних дій, таких як біг, стрибки чи виконання складних маневрів, що потребують значних витрат енергії.

PistolEquipMontage: Ця команда активує анімацію екіпірування пістолета. Вона охоплює такі дії, як діставання пістолета з кобури та його підготовка до використання в грі.

RifleEquipMontage: Відповідає за анімацію екіпірування гвинтівки. Включає рухи, пов'язані з підняттям зброї з землі або стоячого положення, а також її підготовку до стрільби.

Система ModelsAndSkins використовується для управління зовнішнім виглядом персонажів у грі. Вона надає можливість змінювати вигляд героїв, використовуючи різні моделі та текстури. Гравці мають змогу обирати між різними стилями, образами та варіантами одягу для своїх персонажів за допомогою цієї функціональності. Це дозволяє кожному гравцеві створити унікальний образ свого героя, забезпечуючи індивідуальний підхід до налаштування персонажів.

Функція UnlockLevelCharacter реалізує систему підвищення рівня персонажа на основі знищення ворогів та здобуття медалей, що залежить від прогресу гравця в грі. Кожне знищення ворога приносить досвід, що сприяє підвищенню рівня персонажа. Медалі є винагородою за досягнення нових рівнів, які отримуються за виконання певних цілей.

Розробка механіки стрільби в Unreal Engine 4 базується на реалізації відповідної логіки. Перш за все, створюється вхідна подія для виконання стрільби, що активується натисканням "Лівої кнопки миші". У Blueprint додається Event Tick, після чого цей компонент з'єднується з подією, яка відповідає натисканню кнопки "Pressed" для "Лівої кнопки миші".

Далі розробляється вихідна подія для стрільби. До Blueprint додається компонент "Куля", який служить основним елементом для реалізації процесу стрільби. Вхід "Pressed", що відповідає "Лівій кнопці миші", з'єднується з входом "Fire", який запускає дію стрільби.

Після цього налаштовується вхідна подія стрільби. Event Tick інтегрується в Blueprint і підключається до входу "Fire". Наступним кроком створюється вихідна подія для стрільби.

Для визначення траєкторії кулі або снаряду встановлюється вектор швидкості. Обробляються події потрапляння кулі в ціль або зіткнення з об'єктами. Додатково реалізуються ефекти стрільби: звуковий супровід, візуалізація вогню з дула зброї та нарахування пошкоджень.

На рис. 3.17 продемонстровано основні етапи створення механіки стрільби.



Рис.3.17 Зображення розробки стрільби

3.7 Інструкція користувача гри

Посібник користувача до гри "3-D Game Shooter".

Вітаємо у світі захоплюючого науково-фантастичного тактичного екшену! Цей посібник ознайомить вас із ключовими функціями, елементами управління та механіками гри, щоб ви могли поринути у незабутні битви та пригоди.

Управління персонажем:

- Для переміщення використовуйте клавіші:

W – рух уперед,

A – вліво,

S – назад,

D – вправо.

- Миша використовується для орієнтації та огляду.
- Ліва кнопка миші дозволяє атакувати ворогів або взаємодіяти з об'єктами.
- Натискання клавіші Пробіл виконує стрибок або перехід у захисне укриття.

Бойова система:

- Для прицілювання затисніть праву кнопку миші.
- Використовуйте різні типи зброї, щоб знайти найкращу стратегію для перемоги над супротивниками.
- Слідкуйте за рівнем здоров'я персонажа: застосовуйте аптечки та інші предмети для його відновлення.

Взаємодія з ігровим світом

- Досліджуйте навколишнє середовище, взаємодіючи з персонажами та об'єктами.
- Збирайте корисні ресурси, які допоможуть у проходженні гри.

Менеджмент персонажа:

- Набирайте досвід і отримуйте нагороди, такі як медалі.

- Змінійте зброю та налаштовуйте зовнішній вигляд персонажа, щоб зробити гру максимально зручною та приємною.

Місії та завдання:

Ваше головне завдання – зупинити ворогів, які загрожують миру на вашій рідній планеті. Кожен супротивник – це виклик, що ставить під загрозу безпеку та гармонію вашого світу. Захистіть свою землю від вторгнення ворожих сил, які намагаються її підкорити.

Збереження прогресу:

- Використовуйте функції автоматичного або ручного збереження, щоб не втратити свій прогрес та мати змогу повернутися до гри в будь-який момент.

Нехай ваші пригоди у світі тактичного екшену принесуть багато яскравих вражень та перемог!

ВИСНОВКИ

У процесі виконання цієї магістерської роботи були детально розглянуті різні аспекти створення інтерактивних 3D-середовищ на основі Unreal Engine 4, а також методи, інструменти та можливості, що пропонує ця платформа для розробників. В результаті проведеного дослідження були сформульовані основні висновки, які висвітлюють ключові моменти використання Unreal Engine 4 у розробці 3D-середовищ, переваги та обмеження цієї платформи, а також перспективи її розвитку.

Unreal Engine 4 є однією з найбільш потужних і гнучких платформ для розробки відеоігор та інтерактивних 3D-середовищ. Вона дозволяє створювати високоякісну графіку, реалістичну фізику, а також інтерактивні елементи, що робить її ідеальним інструментом для розробки ігор, віртуальних симуляцій, архітектурних візуалізацій і навіть віртуальної реальності. Високий рівень деталізації графіки, фізичні симуляції, освітлення, а також інтеграція з різноманітними системами зробили Unreal Engine 4 основною платформою для багатьох провідних розробників по всьому світу.

Однією з великих переваг Unreal Engine 4 є наявність великої кількості плагінів і додаткових інструментів, які дозволяють розширити функціональність платформи. Розробники можуть використовувати плагіни для інтеграції сторонніх технологій, таких як віртуальна реальність, додаткові фізичні моделі, а також інструменти для роботи з анімацією і моделюванням. Завдяки цьому Unreal Engine 4 стає ще більш гнучким і зручним для використання в різних сферах.

Розробка інтерактивних 3D-середовищ на основі Unreal Engine 4 дає змогу вирішувати численні завдання, починаючи від створення відеоігор до реалізації навчальних і симуляційних програм. Висока якість графіки, зручні інструменти для роботи з анімацією, фізичними симуляціями та освітленням дозволяють створювати продукти, які мають високий рівень реалізму та інтерактивності.

У кваліфікаційній роботі було детально розглянуто ключові етапи програмної реалізації комп'ютерної гри, створеної за допомогою інструментів

Unreal Engine 4. Зокрема, було проаналізовано архітектуру проєкту за допомогою діаграми класів, яка показала структуру взаємозв'язків між основними компонентами гри, такими як GameEngine, UI, Персонаж та Предмети. Це дозволило визначити логічну побудову гри та ефективний розподіл функцій між класами.

Важливим аспектом є розробка функціоналу гри, що включає створення головного меню, екранів завантаження та вибору режимів. За допомогою Blueprint-інтерфейсу вдалося реалізувати інтуїтивний і зручний користувацький інтерфейс із мінімальними витратами ресурсів. Також було використано Landscape Tool та Level Editor для розробки ігрової місцевості, що забезпечило реалістичне відображення рельєфу й об'єктів у середовищі гри.

У підрозділі механіки руху персонажа було застосовано комплексний підхід до моделювання та анімації. Основні етапи — створення моделі, UV-розгортка, текстурювання, риггінг та анімація — виконані із використанням зовнішніх 3D-редакторів (Blender, 3ds Max) та інструментів Unreal Engine. Налаштовані функції, такі як IA_Move, IA_Crouch, IA_FireWeapon тощо, забезпечили розширену інтерактивність і управління персонажем.

Таким чином, програмна реалізація гри демонструє комплексний і системний підхід до створення проєкту на базі Unreal Engine 4. Поєднання сучасних інструментів і методів дозволило створити оптимізовану, функціональну й інтерактивну гру з високою якістю графіки, анімації та управління.

Підсумовуючи, Unreal Engine 4 — це інструмент, який відкриває безліч можливостей для розробників у всьому світі. Він не лише дозволяє реалізовувати масштабні проєкти з винятковою якістю, але й сприяє розвитку індустрії інтерактивних технологій у цілому. Хоча рушій потребує певного рівня знань і ресурсів, його потужність і універсальність роблять його ідеальним вибором для тих, хто прагне створювати інноваційні та якісні продукти.

ПЕРЕЛІК ПОСИЛАНЬ

1. Unreal Engine 4 Documentation. Epic Games. [Електронний ресурс]. – Режим доступу: <https://docs.unrealengine.com>.
2. Miller, S. *Mastering Unreal Engine 4: The Art of Game Development*. – Packt Publishing, 2021.
3. Kim, T. *Architectural Visualization with Unreal Engine 4*. – CRC Press, 2020.
4. Cawood, J. *Creating Augmented and Virtual Realities with Unreal Engine*. – Sybex, 2020.
5. Шаронов, В. С. *Основы разработки игр на Unreal Engine 4*. – Москва: Эксмо, 2021.
6. Wagner, T. *Lighting and Rendering Techniques in Unreal Engine 4*. – O'Reilly Media, 2020.
7. Авраменко, Д. *Моделювання та оптимізація графічних середовищ для інтерактивних додатків*. – Київ: Наукова думка, 2020.
8. Blender 3D Official Documentation. Blender Foundation. [Електронний ресурс]. – Режим доступу: <https://www.blender.org/documentation>.
9. Unreal Engine Blog. Epic Games. [Електронний ресурс]. – Режим доступу: <https://www.unrealengine.com/blog>.
10. Weiss, R. *Procedural Content Generation in Unreal Engine 4*. – Packt Publishing, 2021.
11. Epic Games. *Unreal Engine 4 Documentation*. [Електронний ресурс]. – Режим доступу: <https://docs.unrealengine.com/>.
12. Gregory, J. *Game Engine Architecture*. – 3rd ed. – CRC Press, 2019. – 1024 p.
13. Szabo, Z., & Kiss, G. *Developing Interactive Environments Using Unreal Engine 4 // International Journal of Computer Science and Applications*. – 2020. – № 12(3). – С. 45-60.

14. Blender Foundation. *Blender Manual: 3D Modeling and Texturing*. [Електронний ресурс]. – Режим доступу: <https://docs.blender.org/>.
15. Мартинюк О.С. *Основи розробки інтерактивних середовищ у відеоіграх*. – Київ: Політехніка, 2021. – 256 с.
16. Павлов А.В., Климчук В.М. *Оптимізація 3D-графіки та ігрових середовищ у реальному часі*. – Харків: Техноцентр, 2020. – 312 с.
17. Unreal Engine Community. *Best Practices for Performance Optimization in UE4*. [Електронний ресурс]. – Режим доступу: <https://forums.unrealengine.com/>.
18. Watt, A. *3D Computer Graphics: A Mathematical Introduction with OpenGL*. – Addison-Wesley, 2016. – 611 p.
19. Petrovic, J., & Martin, K. *Lighting and Rendering Techniques in Unreal Engine 4*. // *Journal of Game Development Technologies*. – 2021. – № 8(4). – С. 35-50.
20. Licea-Kane, B. *Direct3D and OpenGL in Real-Time Graphics Applications*. – New York: Springer, 2018. – 436 p.
21. Cook, T., & Grant, S. *Unreal Engine 4 for Beginners: Creating Your First Game Environment*. – Packt Publishing, 2019. – 376 p.
22. Schell, J. *The Art of Game Design: A Book of Lenses*. – 3rd ed. – CRC Press, 2019. – 600 p.
23. Unreal Engine Marketplace. *High-Quality Assets for Game Development*. [Електронний ресурс]. – Режим доступу: <https://www.unrealengine.com/marketplace>.
24. Badawi, A. *Performance Optimization and Level Design in Unreal Engine 4*. – Packt Publishing, 2020. – 290 p.
25. *IEEE Transactions on Visualization and Computer Graphics*. [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=2945>.
26. Unreal Engine. *Blueprints Visual Scripting System Overview*. [Електронний ресурс]. – Режим доступу: <https://docs.unrealengine.com/en-US/Programming/Blueprints/>.
27. Як працює Unreal Engine [Електронний ресурс]. – Режим доступу: <https://skvot.io/uk/blog/ne-soromno-zapitati-yak-pracyuye-unreal-engine>

28. Рекомендоване обладнання для Unreal Engine Engine [Електронний ресурс]. – Режим доступу: <https://artline.ua/uk/news/rekomenduемое-oborudovanie-dlya-unreal-engine>

29. Знайомство з Unreal Engine для початківців [Електронний ресурс]. – Режим доступу: <https://uk.sharpcoderblog.com/blog/introduction-to-unreal-engine-for-beginners>

30. Стилiзоване оточення в Unreal Engine 4 [Електронний ресурс]. – Режим доступу: <http://3das.com.ua/stilizovane-otochennya-v-unreal-engine-4/>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

Державний університет інформаційно-комунікаційних технологій
Кафедра інформаційних систем та технологій

МАГІСТЕРСЬКА РОБОТА

на тему:

“Платформа розробки інтерактивних 3D-середовищ для відеоігор на основі Unreal Engine 4”

Виконав: студент групи ІСДМ-61
Крушеніцький Марк

Керівник: к.т.н., доцент кафедри
ІСТ
Оксана ТКАЛЕНКО

Київ - 2024

• **Актуальність теми:**

- За останні роки значно зросла популярність інтерактивних 3D-середовищ, які активно використовуються у відеоіграх, віртуальній реальності, архітектурній візуалізації та навчальних симуляціях. Завдяки розвитку інструментів, таких як Unreal Engine 4, стало можливим створювати реалістичні, деталізовані та інтерактивні середовища з високою графічною якістю. Попит на такі технології зростає через збільшення потреби у реалістичному візуальному контенті та інтерактивних рішеннях. Дослідження можливостей і функціональних особливостей Unreal Engine 4 для створення інтерактивних середовищ є актуальним для розвитку ігрової індустрії, віртуальної реальності та суміжних галузей.

• **Мета роботи:**

- Розробка інтерактивного 3D-середовища з використанням Unreal Engine 4 та оцінка його функціональних можливостей у створенні високоякісних графічних рішень.

• **Завдання дослідження:**

- Проаналізувати сучасні рушії для створення інтерактивних 3D-середовищ.
- Дослідити функціональні можливості Unreal Engine 4, зокрема систему Blueprints та технології рендерингу.
- Розробити прототип інтерактивного 3D-середовища, що демонструє можливості Unreal Engine 4.
- Провести тестування та оптимізацію середовища для забезпечення стабільної роботи.
- Розробити рекомендації щодо використання Unreal Engine 4 для інтерактивних 3D-проектів.

• **Об'єкт дослідження:**

- Процес створення інтерактивних 3D-середовищ.

Огляд платформ для 3D-середовищ

Unreal Engine 4: Висока якість рендерингу, система Blueprints, фізика.

Unity: Кросплатформність, зручний інтерфейс для 2D/3D-проектів.

CryEngine: Реалістичне освітлення, підтримка великих відкритих світів.



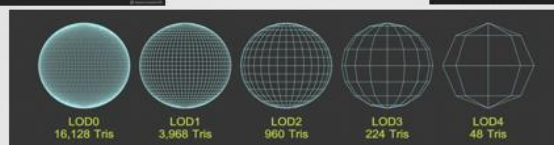
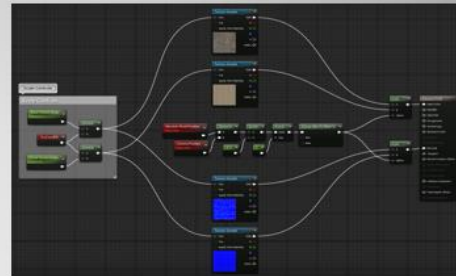
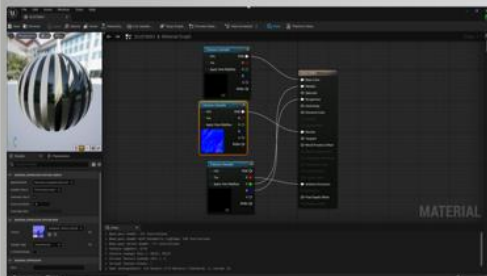
			
Features	Unity	Unreal Engine	CryEngine
Languages	C#	C++	Lua, C++, C#
VPL	Unity ECS	Blueprints	Flow Graph
Community	123k+	42.6k	10.2k+
Source Modify	Yes - Paid	Yes - Royalties	Yes - Royalties
Platforms	PC, GC, WEB, MD, XR	PC, GC, WEB, MD, XR	PC, GC, WEB, MD, XR
Assets Store	AssetStore	MarketPlace	MarketPlace
License	EULA	EULA	EULA
Developer's Year	Unity Technologies 08.06.2005	Epic Games 28.04.1997	Crytek 02.05.2002

Технологічні можливості Unreal Engine 4

Система рендерингу: Реалістичне освітлення, текстурзування.

Blueprints: Візуальне програмування ігрової логіки.

Оптимізація: Рівні деталізації (LOD), система Level Streaming.



Інструменти моделювання

Blender: Моделювання, текстурування, скульптинг.

ZBrush: Високополігональні моделі з деталізацією.

Substance Painter: Текстурування з підтримкою PBR.



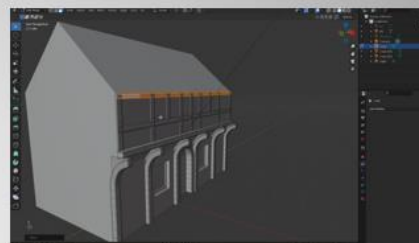
Етапи дослідження

Концепція: Аналіз вимог до інтерактивного 3D-середовища.

Моделювання: Оцінка ефективності створення низько- та високополігональних моделей.

Освітлення: Дослідження методів налаштування динамічного та статичного освітлення.

Оптимізація: Аналіз підходів до зниження кількості полігонів і оптимізації текстур.

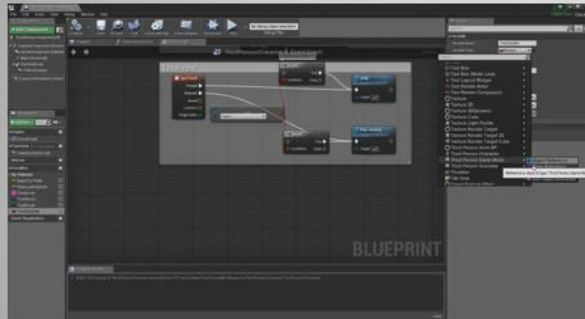


Використання Blueprints

Логіка взаємодії: Аналіз застосування тригерів та подій.

Поведінка об'єктів: Дослідження підходів до створення інтерактивних сценаріїв.

Прототипування: Вивчення можливостей швидкого створення ігрових механік.



Тестування та оптимізація

Платформи: Дослідження продуктивності на ПК, мобільних пристроях.

Методи: Вивчення інструментів профілювання та методів оптимізації.

Результат: Аналіз продуктивності для різних пристроїв.



Результати дослідження

Висновки: Unreal Engine 4 надає ефективні інструменти для створення інтерактивних середовищ.

Аналіз: Досліджено функції Blueprints, рендеринг і оптимізацію LOD.

Особливості: Виявлено переваги та обмеження використання Unreal Engine 4.

- Проведено комплексне дослідження можливостей Unreal Engine 4 для створення інтерактивних 3D-середовищ.
- Вивчено технології рендерингу, включаючи підтримку Physically Based Rendering (PBR) і систему глобального освітлення.
- Досліджено систему Blueprints, яка дозволяє створювати ігрову логіку без використання програмування, спрощуючи процес проектування.
- Встановлено ефективність рівнів деталізації (LOD) для оптимізації продуктивності великих середовищ.
- Проаналізовано методи інтеграції Unreal Engine 4 з іншими інструментами, такими як Blender і Substance Painter, для створення високоякісних 3D-моделей.
- Результати показали, що Unreal Engine 4 є ефективним інструментом для створення високоякісних середовищ із можливістю оптимізації для різних платформ.

Висновки

Unreal Engine 4 підтвердив свою ефективність для створення якісних інтерактивних середовищ.

Дослідження показало можливості оптимізації продуктивності.

Результати можуть бути застосовані у проєктах різних масштабів.

Мною підтверджено, що Unreal Engine 4 є ефективним інструментом для створення високоякісних інтерактивних середовищ із можливістю їхньої оптимізації.

Крушеніцький М.О. «Дослідження можливостей платформи Unreal Engine 4 для розробки інтерактивних 3D-середовищ».

Тези доповіді на I Всеукраїнській науково-технічній конференції «Інновації в інформаційних технологіях: сучасні виклики та перспективи». – Київ, 23 грудня 2024 року.

ДЯКУЮ ЗА УВАГУ!