

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Економічний факультет
Кафедра економіко-математичного моделювання та інформаційних техно-
логій

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавра

на тему: **«Створення інтерактивного віртуального середовища
НаУОА (екстер'єр адмінбудівлі та прилеглих територій)»**

Виконав: студент 4-го курсу, групи КН-41
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Лаптев Богдан Валерійович

Керівник: *Місай В.В.*, викладач кафедри ЕММІТ

Рецензент: *кандидат технічних наук, доцент, доцент
кафедри прикладної математики та кібербезпеки До-*
нецького національного університету імені Василя
Стуса

Загоруйко Любов Василівна

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри економіко-математичного моделювання та інфор-
маційних технологій _____ (проф., д.е.н. Кривицька О.Р.)

Протокол №11 від «30» травня 2024 р.

Острог, 2024

Міністерство освіти і науки України
Національний університет «Острозька академія»

Факультет: економічний

Кафедра: економіко-математичного моделювання та інформаційних технологій

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри ЕММІТ
Ольга КРИВИЦЬКА

« ____ » _____ 20__ р.

ЗАВДАННЯ
на кваліфікаційну роботу студента

Лаптева Богдана Валерійовича

1. Тема роботи: Створення інтерактивного віртуального середовища НаУОА (екстер'єр адмінбудівлі та прилеглих територій)

керівник роботи: Місай В.В., викладач кафедри ЕММІТ

Затверджено наказом ректора НаУОА від 03.11.2023 р., № 98.

2. Термін здачі студентом закінченої роботи: 31 травня 2024 року.

3. Вихідні дані до роботи: для створення проекту використовувались Blender(для моделювання всіх об'єктів та будівель з елементами екстер'єру) Unreal Engine 5 (для створення всіх будівель і конструктивних елементів екстер'єру віртуального середовища) Adobe Photoshop, Adobe Illustrator.

4. Перелік завдань, які належить виконати: створення всіх будівель та прилеглої території, взведення усіх конструктивних елементів, розробка екстер'єру адміністративної будівлі нового корпусу НаУОА , розробка генерації снігу, створення зимового оточення, створення звуків , реалізація механіки міні мапи, створення віртуальних вказівників.

5. Перелік графічного матеріалу: рисунки, таблиці, порівняння, відеоматеріали.

6. Консультанти розділів роботи:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| 1 | Місай В. В. | 01.12.2023 | 01.12.2023 |
| 2 | Місай В. В. | 01.12.2023 | 01.12.2023 |

7. Дата видачі завдання: 01.12.2023 р..

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів | Примітка |
|-------|--|------------------------|----------|
| 1 | Затвердження теми кваліфікаційної роботи | до 31.10.2023 р. | |
| 2 | Постановка технічного завдання | до 01.12.2023 р. | |
| 3 | Ознайомлення з документацією | до 10.12.2023 р. | |
| 4 | Написання розділу 1 | до 01.02.2024 р. | |
| 5 | Написання розділу 2 | до 01.04.2024 р. | |
| 6 | Тестування системи | до 20.04.2024 р. | |
| 7 | Виправлення помилок | до 01.05.2024 р. | |
| 8 | Попередній захист та перевірка на рівень унікальності кваліфікаційної роботи/проекту | до 31.05.2024 р. | |
| 9 | Здача кваліфікаційної роботи/проекту на кафедрі | 31.05.2024 р. | |

Студент: _____ **Лаптев Богдан**
(підпис)

Керівник кваліфікаційної роботи: _____ **Володимир МІСАЙ**
(підпис)

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

Тема: Створення інтерактивного віртуального середовища НаУОА (екстер'єр адмінбудівлі та прилеглих територій)

Автор: Лаптев Богдан Валерійович

Науковий керівник: Місай В.В., викладач кафедри ЕММІТ.

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: 92 с., 70 рис., 1 додаток, 31 джерел.

Ключові слова: інтерактивність, віртуалізація, екстер'єр, Blender, Unreal Engine 5, моделювання, компілювання, створення, конструювання.

Короткий зміст праці:

Завданням кваліфікаційної роботи, було створення екстер'єру адмін будівлі нового корпусу НаУОА та прилеглих територій. Це інтерактивне середовище може служити засобом представлення загального враження для вступників, які мають можливість віртуально оглянути новий корпус. Використання Blender дозволило створити деталізований екстер'єр, а Unreal Engine 5 забезпечив високоякісний рендеринг та інтерактивність у цьому середовищі. Даний проект відкриває нові можливості використання технологій у навчальному процесі та підкреслює потенціал віртуальної реальності для просування університетів та полегшення сприйняття майбутніми абітурієнтами.

Keywords: interactivity, virtualization, exterior, Blender, Unreal Engine 5, modeling, compilation, creation, construction.

The task of the qualification work was to create the exterior of the administration building of the new building of the National Academy of Arts and Sciences and the surrounding area. This interactive environment can serve as a means of presenting a general impression to the participants who have the opportunity to virtually explore the new building. Using Blender made it possible to create a detailed exterior, and Unreal Engine 5 provided high-quality rendering and interactivity in this environment. This project opens up new opportunities for using technology in the educational process and highlights the potential of virtual reality to promote universities and facilitate acceptance by future applicants.

Зміст

| | |
|--|----|
| ВСТУП | 4 |
| РОЗДІЛ 1.Теоретична частина.Створення інтерактивного віртуального середовища НаУ ОА (екстер'єр адмінбудівлі та прилеглих територій) та вибір програмного забезпечення Blender та ігрового рушія Unreal Engine 5 | 5 |
| 1.1. Постановка проблеми створення інтерактивного віртуального середовища НаУ ОА (екстер'єр адмінбудівлі та прилеглих територій) | 5 |
| 1.2. Опис базових понять у виконанні проекту у програмі Blender..... | 6 |
| 1.3. Опис базових понять у виконанні проекту у рушії Unreal Engine 5 | 8 |
| 1.4. Аналіз аналогічних розробок | 13 |
| РОЗДІЛ 2.Створення інтерактивного віртуального середовища НаУ ОА (екстер'єр адмінбудівлі та прилеглих територій) у програмному середовищі Blender та у ігровому рушії Unreal Engine 5 | 15 |
| 2.1. Організація управління та планування проекту за допомогою jira | 15 |
| 2.2. Опис технологічного стеку | 18 |
| 2.3. Збір інформації та її підготовка | 19 |
| 2.4 Створення 3D моделей в Blender | 22 |
| 2.5 Інтеграція 3D моделей в Unreal Engine | 33 |
| 2.6 Створення інтерактивного віртуального середовища в Unreal Engine 5..... | 42 |
| ВИСНОВКИ | 83 |
| Список використаних джерел | 85 |
| ДОДАТКИ | 87 |

ВСТУП

У сучасному світі, ігрова індустрія швидко розвивається, і розробники постійно шукають нові способи привернути увагу гравців. Використання віртуальної реальності та взаємодії стає важливим для створення цікавих ігор. Також важливо розуміти, що задоволення від ігрового процесу є ключовим для успіху проекту.

Тема даного проекту: Створення інтерактивного віртуального середовища НаУОА (включаючи зовнішній вигляд адмінбудівлі та прилеглі території) у ігровому рушії Unreal Engine 5. Ця тема обрана для розробки оптимальних та функціональних компонентів у проекті. Інтерактивність є ключовим аспектом, який зробить віртуальне середовище більш реалістичним та привабливим для гравців.

Мета кваліфікаційної роботи: Розробка та втілення інтерактивного віртуального середовища для Національного університету "Острозька академія", що відображатиме зовнішність адміністративної будівлі та прилеглих територій. Основна мета полягає в створенні реалістичного та функціонального віртуального простору, який дозволить користувачам взаємодіяти з ним, вивчати його та отримувати інформацію про архітектурні особливості та ландшафтний дизайн об'єкта. Робота спрямована на застосування передових технологій віртуалізації з метою покращення якості освітнього процесу та створення зручного середовища для навчання та досліджень університетської спільноти.

Головне завдання: Створення і впровадження інтерактивного віртуального середовища, яке точно відтворює зовнішність адміністративної будівлі та прилеглих територій Національного університету "Острозька академія". Це включає розробку віртуальних моделей будівель і ландшафтного оформлення, а також створення функціоналу для взаємодії користувачів з цим середовищем, його дослідження та отримання необхідної інформації.

Предмет дослідження: Аспекти створення віртуального середовища для відтворення зовнішнього вигляду та інтерактивної взаємодії з середовищем Національного університету "Острозька академія" включають архітектурні

деталі будівель, елементи ландшафту на території та можливості реалізації інтерактивних функцій для поліпшення враження користувачів.

Завдяки цим заходам та використанню базових функцій Blender та Unreal Engine 5, планується створення інтерактивного віртуального середовища, яке відповідатиме всім вимогам та планам, що ставляться перед університетом у реальному житті.

Для розробки проекту з використанням програм Blender та Unreal Engine 5 планується не тільки моделювання та візуалізація адміністративної будівлі та прилеглих територій, але й проведення аналізу різних аспектів, що впливають на створення віртуального та інтерактивного середовища університету.

Під час роботи враховуватимуться технічні особливості програмних засобів, їх можливості та обмеження з метою досягнення максимально реалістичного результату. Розробка також включатиме формулювання та впровадження практичних рекомендацій, спрямованих на оптимізацію та покращення якості віртуального середовища.

РОЗДІЛ 1

ТЕОРЕТИЧНА ЧАСТИНА. СТВОРЕННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА НАУ ОА (ЕКСТЕР'ЄР АДМІНБУДІВЛІ ТА ПРИЛЕГЛИХ ТЕРИТОРІЙ) ТА ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ BLENDER ТА ІГРОВОГО РУШІЯ UNREAL ENGINE 5

1.1. Постановка проблеми створення інтерактивного віртуального середовища Нау ОА (екстер'єр адмінбудівлі та прилеглих територій)

Під час створення інтерактивного віртуального середовища Нау ОА (екстер'єр адмінбудівлі та прилеглих територій) потрібно враховувати різні аспекти у створенні даного середовища. У контексті створення екстер'єру адміністративної будівлі та прилеглих територій виникають деякі важливі аспекти, які варто розглянути:

1. Візуальна естетика та функціональність: З головних поставлених проблем є забезпечення гармонії між візуальним виглядом та функціональністю екстер'єру . Потрібно правильно зробити архітектуру щоб вона була привабливою для перегляду, але потрібно враховувати правильність розташування різних елементів та відповідало реальності також.
2. Зручність: Створення екстер'єру повинно сприяти створенню комфортного та зручного простору для користувачів. Які елементи та рішення дизайну сприятимуть покращенню робочого середовища та загального комфорту?
3. Інтеграція з навколишнім середовищем: Потрібно правильно об'єднати адмінбудівлю та прилеглі території корпусу щоб воно мало естетичний вигляд.
4. Технологічні рішення: Розвиток технологій дає змогу отримувати велику кількість інструментів для полегшення роботи у створенні різних продуктів.

5. Інтерактивна взаємодія з віртуальним середовищем: Передбачає можливість користувача взаємодіяти з об'єктами та елементами в цьому середовищі за допомогою різних методів інтерфейсу.

1.2. Опис базових понять у виконанні проекту у програмі Blender

Це важливий етап який дає розуміння для чого і як можна використовувати дану програму у проекті. Дане програмне забезпечення є багатофункціональне, тому важко описати все про дану програму і до того ж у проекті всі функції використати фізично неможливо.

Blender - це потужний та відкритий програмний продукт для 3D-моделювання, анімації, рендерингу, композитингу, редагування відео та багатьох інших завдань.

Ось кілька понять які застосовуються у проекті:

- Моделювання (Modeling): Процес створення геометричних форм та об'єктів у тривимірному просторі. У Blender це може бути виконано за допомогою різних інструментів, таких як ручне моделювання, скульптування, чи використання готових примітивів.
- Матеріали та текстури (Materials and Textures): Визначення вигляду та поведінки поверхонь об'єктів. Матеріали визначають властивості, такі як колір, блиск, прозорість, тоді як текстури додають деталі та реалізм.
- Освітлення (Lighting): Створення та налаштування світла в сцені для досягнення потрібного ефекту. Blender дозволяє використовувати різні типи джерел світла, такі як сонце, точкові джерела світла, або навіть освітлення від HDRI-зображень.
- Рендеринг (Rendering): Процес створення фінального, готового до перегляду, зображення з тривимірної сцени. Blender має вбудовані двигуни рендерингу, такі як Cycles та Eevee.

- Сцена (Scene): Область, де розташовуються всі об'єкти, камери, світло та інші елементи проекту. Сцена визначає простір, на якому моделюються та візуалізуються об'єкти .
- Управління камерою (Camera Control): Визначення точки перегляду та налаштування параметрів камери для створення відповідного ракурсу перегляду сцени

Переваги Blender над іншими програмами:

- безкоштовність та відкритий код: Blender є повністю безкоштовним та доступним для використання без обмежень. Його відкритий код означає, що користувачі можуть переглядати, змінювати та розповсюджувати програму згідно з умовами ліцензії GNU GPL.
- гнучкість та настільні можливості: Blender має гнучкий та настроюваний інтерфейс, який дозволяє користувачам налаштовувати робочий простір під свої потреби. Це особливо корисно для користувачів з різними вимогами та рівнями досвіду.
- широкі можливості: Blender підтримує широкий спектр завдань від 3D-моделювання та анімації до відеоредагування, композитингу, рендерингу та багатьох інших. Це робить його важливим інструментом для великої кількості креативних завдань.
- спільнота та ресурси: Blender має велику та активну спільноту користувачів, що сприяє обміну знаннями, допомогою та взаємодією. Існує багато онлайн-ресурсів, таких як уроки, форуми та відео, які допомагають користувачам в освоєнні програми.
- можливості анімації: Blender надає широкі можливості для створення анімацій, включаючи скелетну анімацію, різні техніки кадрової анімації, анімацію частинок та інші ефекти.
- підтримка різних платформ: Blender доступний для використання на різних операційних системах, включаючи Windows, macOS та Linux, що дозволяє користувачам використовувати його на різних пристроях.

- інтеграція з різними програмами та рушіями: Blender добре працює з різними програмами разом і чудово працює інтеграція моделей в інші програми чи рушії.
- продовження розвитку: Blender постійно оновлюється та розвивається завдяки активній розробці та внесенню внесків спільнотою. Оновлення та нововведення регулярно надаються користувачам.

1.3. Опис базових понять у виконанні проекту у рушії Unreal Engine 5

Unreal Engine5 (UE5) - це потужне інтегроване середовище розробки та ігровий рушій, розроблений Epic Games. Він призначений для створення ігрових додатків високого графічного рівня, віртуальних середовищ, симуляторів, архітектурних візуалізацій і багатьох інших добре відомих додатків, і не тільки в індустрії. Під час оновлення в 2022/1 UE5 був доступний лише як частина попереднього доступу.

Для виконання проекту також ми використовуємо даний рушій для створення та реалізації віртуального середовища .

Ось стислий перелік термінів, що застосовуються у проекті для досягнення конкретного результату:

- **Актори (Actors):** В основі Unreal Engine 5 лежить концепція акторів, які є основними об'єктами в ігровому світі. Актори можуть представляти персонажів, об'єкти, світло, камери, та інше.
- **Блюпринти (Blueprints):** Графічні скрипти, що дозволяють створювати функціональність та логіку гри без програмування. Блюпринти можуть визначати взаємодію акторів, анімації, фізику та багато іншого.[31]
- **Матеріали та Текстури (Materials and Textures):** У Unreal Engine 5, аналогічно до Blender, використовуються матеріали та текстури для визначення вигляду об'єктів. Матеріали можуть включати текстури, шейдери та інші ефекти.

- Рівні (Levels): Графічні та просторові компоненти ігрового світу, які можуть бути завантажені та вивантажені окремо. У реальному проекті рівні можуть представляти окремі локації чи рівні складності гри.
- Світло та Тіні (Lighting and Shadows): Unreal Engine 5 використовує потужні алгоритми обчислення світла і тіней для створення реалістичного освітлення в грі. Global Illumination (GI) та Lumen є деякі з нових функцій, які поліпшують якість освітлення.
- Анімація (Animation): Система анімації в Unreal Engine 5 дозволяє створювати реалістичні та плавні анімації для персонажів, об'єктів, та інших акторів.
- Проведення Фізики (Physics Simulation): Unreal Engine 5 має вбудовану систему фізики, яка дозволяє моделювати реалістичні фізичні властивості об'єктів, їх взаємодію та реакцію на середовище.
- Віртуальна Реальність (VR): Unreal Engine 5 підтримує розробку VR-проектів, що дозволяє створювати імерсивні ігрові досвіди для віртуальної реальності.
- Рендеринг (Rendering): Високоякісний рендеринг у реальному часі забезпечує створення реалістичних зображень та вражаючих графічних ефектів.
- Mesh: Використовується для опису геометричних об'єктів у тривимірному просторі, які можуть бути відображені в грі або інтерактивній програмі. Mesh може бути будь-якого форми та складності: від простих геометричних фігур, таких як куби або сфери, до складних моделей персонажів, транспортних засобів або будівель.
- Dynamic Mesh: Надає можливість створювати, модифікувати та відображати меші безпосередньо у грі з високою продуктивністю. Це дозволяє реалізувати різноманітні динамічні ефекти, такі як динамічні об'єкти, динамічне руйнування, анімовані персонажі та багато іншого.
- Widget: Використовується для створення інтерфейсу користувача (UI) в іграх та інтерактивних додатках, розроблених з використанням Unreal Engine 5. Віджети дозволяють розміщувати елементи інтерфейсу, такі як

кнопки, текстові поля, зображення, списки тощо, на екрані гри або додатка. Вони можуть взаємодіяти з гравцем або користувачем і виконувати різноманітні функції, такі як відображення інформації, управління грою, введення даних тощо.

- Raycasts: використовуються для взаємодії з об'єктами в грі або в інтерактивній сцені.
- Bounds: Обмежувальні прямокутники або об'ємні оболонки, які визначають межі або розміри об'єктів у тривимірному просторі.
- UV: Дозволяє текстурі правильно проєкціюватися на поверхню моделі, надаючи їй реалістичний вигляд з деталізованими текстурами, кольорними ефектами та іншими деталями.
- Smoothing: Використовується для створення більш природного або реалістичного вигляду поверхонь моделей
- Dynamic Mesh Actor: Об'єкт у ігровому двигуні, який представляє собою комбінацію динамічної сітки.
- BVH: Дозволяє розбити простір на області з мінімальною кількістю об'єктів, що дозволяє швидше визначити, які об'єкти можуть перетинатися або колізіонувати між собою.
- Sprite: Двовірний графічний об'єкт, який може бути використаний для створення елементів інтерфейсу користувача, візуальних ефектів, анімації та інших декоративних або інформаційних елементів у віртуальному середовищі.
- Camera: Використовується для визначення точки огляду в грі. Камери визначають положення та орієнтацію глядача у віртуальному просторі гри.
- Geometry: Відноситься до візуальних складових ігрового світу, таких як об'єкти, поверхні, та їхні форми та структури. Геометрія визначає форму та структуру об'єктів у грі і включає в себе вершини, ребра та полігони.

Unreal Engine (UE) є одним з провідних ігрових рушіїв у світі, і він має кілька значущих переваг, які роблять його популярним серед розробників ігор та інших інтерактивних додатків.

Переваги Unreal над іншими програмами:

- потужний графічний двигун: Unreal Engine відомий своєю потужною графічною системою, яка дозволяє створювати вражаючі візуальні ефекти та фотореалістичні зображення у реальному часі.
- система моделювання та анімації: UE має вбудовані інструменти для 3D-моделювання та анімації, що спрощує створення та імпорт об'єктів без необхідності використання інших програм.
- розширена фізична система: Unreal Engine включає потужну фізичну систему, що дозволяє моделювати реалістичні фізичні властивості об'єктів, взаємодію та реакцію на середовище.
- велика бібліотека готових активів: UE надає широкий вибір готових активів, таких як моделі, текстури та анімації, які можна використовувати для швидкого розпочатку роботи.
- мультиплатформенність: Ігри та додатки, розроблені у Unreal Engine, можуть бути легко виведені на різні платформи, такі як PC, консолі, мобільні пристрої та навіть віртуальна реальність.
- безкоштовний для проектів з високим рівнем рентабельності: Unreal Engine безкоштовний для використання в проектах, які генерують невеликий обсяг доходу, і лише здатний до отримання ліцензійних витрат, якщо проект стає високорентабельним.
- активна спільнота та документація: Unreal Engine має велику та активну спільноту користувачів, а також обширну документацію та різноманітні уроки, що допомагають розробникам вирішувати завдання та розвиватися в освоєнні рушію.
- візуалізація та реалізм: UE5 надає потужні інструменти для створення вражаючих ігрових світів з високою деталізацією та реалізмом. Завдяки функціям, таким як Nanite Virtualized Geometry та Lumen Global Illumination, розробники можуть створювати величезні, деталізовані середовища з реалістичним освітленням та тінями.

- інтерактивність та динаміка: UE5 дозволяє реалізувати різноманітні інтерактивні елементи та динамічні ефекти, такі як руйнування, фізика та взаємодія з оточенням, що сприяє створенню емоційно насичених ігрових досвідів.
- швидкість та продуктивність: UE5 пропонує покращену продуктивність та оптимізацію завдяки новим технологіям, таким як World Partition і Nanite, що дозволяють працювати з великими об'ємами даних без втрати продуктивності.
- розширені можливості редагування: Засоби редагування в UE5, такі як редактор матеріалів та редактор світла, надають розробникам багато можливостей для створення високоякісних ігрових зображень та ефектів.
- зручний інтерфейс користувача: Інтерфейс користувача UE5 є дружнім та інтуїтивно зрозумілим, що дозволяє розробникам швидко оволодіти платформою та максимально ефективно використовувати її потужності.
- великий потенціал для реалістичних візуальних ефектів: Unreal Engine 5 надає широкі можливості для створення вражаючих візуальних ефектів, таких як реалістичне освітлення, тіні, атмосферний ефект, рух інтерактивних об'єктів тощо.
- швидкий процес розробки: Завдяки потужним інструментам редактора та багатофункціональному інтерфейсу, Unreal Engine 5 дозволяє розробникам швидко створювати, тестувати і вдосконалювати віртуальні середовища.
- потужність та масштабованість: Unreal Engine 5 може легко масштабуватись для роботи над проектами будь-якої складності і масштабу, починаючи від невеликих інді-ігор до великих AAA-проектів.
- гнучкість та модульність: Редактор Unreal Engine 5 дозволяє розробникам створювати власні розширення, плагіни та модулі, що робить його гнучким і адаптованим для різноманітних потреб розробки.
- маркетплейс: Unreal Engine має власний маркетплейс, де розробники можуть придбати або продавати готові активи, матеріали та розширення для своїх проєктів, що спрощує розробку та знижує витрати на створення контенту.
- графічна якість: Unreal Engine відомий своєю вражаючою графікою та візуальними ефектами, що дозволяє створювати надзвичайно реалістичні та естетично привабливі віртуальні середовища.

- візуальне програмування: Використання системи вузлів дозволяє програмістам та дизайнерам створювати логіку та функціональність за допомогою графічного інтерфейсу, що спрощує процес розробки та полегшує спільну роботу команди.
- спрощена інтеграція з C++: Система вузлів в Unreal Engine легко інтегрується з кодом на C++, що дозволяє поєднувати вузловий та кодовий підходи до розробки та використовувати переваги обох підходів.

1.4. Аналіз аналогічних розробок

При пошуку аналогічних робіт нашого проекту виявилось що не так багато є робіт які навіть приблизно схожі на наш проект. Це все тому що для цієї роботи потрібно багато ресурсів та часу що сильно ускладнює процес розробки і це сповільнює загалом роботу по проекту.

Провівши пошуки в інтернет мережі ми знайшли 3D-модель НаУОА від Skeiron (Рис. 1.1) , яка представляє скановану модель головного корпусу НаУОА . Однак, ця модель не може порівнюватися з нашим проектом через низьку якість та відсутність можливості віртуальної прогулянки по території університету.

Крім того, модель не має інтерактивності та взаємодії із середовищем, тому вона використовується лише для загального огляду будівлі та прилеглої території.



Рис.1.1. Вигляд сканованої моделі НаУОА розробленої Skeiron

Джерело: Google Arts & Culture

Наш проект, на відміну від моделі , яку розробили Skeiron , передбачає високоякісну візуалізацію, інтерактивність та можливість віртуального туру. Це дозволяє користувачам детально ознайомитися з усіма аспектами університетського комплексу, взаємодіяти з різними елементами та отримати більш глибоке розуміння простору.

Крім того, наш проект включає додаткові функції, такі як інформаційні підказки та інтеграція з іншими системами для покращення користувацького досвіду.

Ці додаткові функції дозволяють користувачам отримати не лише повний обсяг інформації про університетський комплекс, але й забезпечать зручний та цікавий спосіб взаємодії з цим простором. Наприклад, інформаційні підказки можуть надати корисну інформацію про окремі елементи або місця в університеті, такі як аудиторії, бібліотека, спортивні майданчики тощо. Інтеграція з іншими системами може підвищити функціональність проекту.

РОЗДІЛ 2

СТВОРЕННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА НАУ ОА (ЕКСТЕР'ЄР АДМІНБУДІВЛІ ТА ПРИЛЕГЛИХ ТЕРИТОРІЙ) У ПРОГРАМНОМУ СЕРЕДОВИЩІ BLENDER ТА У ІГРОВОМУ РУШІ UNREAL ENGINE 5

2.1. Організація управління та планування проекту за допомогою jira

2.1.1. Розподіл завдань у команді

Перед початком розробки над проектом і для кращого розуміння хто і що робить у проекті , ми з командою вирішили розподілити завдання між собою за допомогою Jira Software[20]. Кожний учасник команди мав власні спринти з відповідними завданнями для виконання проекту.

Це дало краще розуміння які завдання виконує учасник і Jira Software дає можливість контролювати прогрес виконання і скільки часу було витрачено на виконання завдань.

Крім того, використання Jira Software дозволило нам ефективно спілкуватися щодо проекту, обмінюватися інформацією та вирішувати будь-які питання або проблеми, що виникали під час розробки. Також ми мали змогу створювати звіти і аналізувати продуктивність кожного учасника команди, що сприяло вдосконаленню наших процесів розробки.



Рис.2.1. Завдання у Jira Software

Джерело: розроблено автором

Також для кращого візуального вигляду ми використали Scrum який структурно дав змогу розподілити завдання на етапи :

TO DO: Цей етап відображає завдання, які потрібно виконати у майбутньому.

IN PROGRESS: В цій колонці знаходяться завдання, які вже у процесі виконання.

CHECKING: Тут перевіряються завдання після їх виконання.

TESTING: На цьому етапі відбувається тестування усіх механік для виявлення багів і подальшого вдосконалення.

DONE: У цій колонці знаходяться завдання, які успішно пройшли всі попередні етапи та вважаються виконаними.

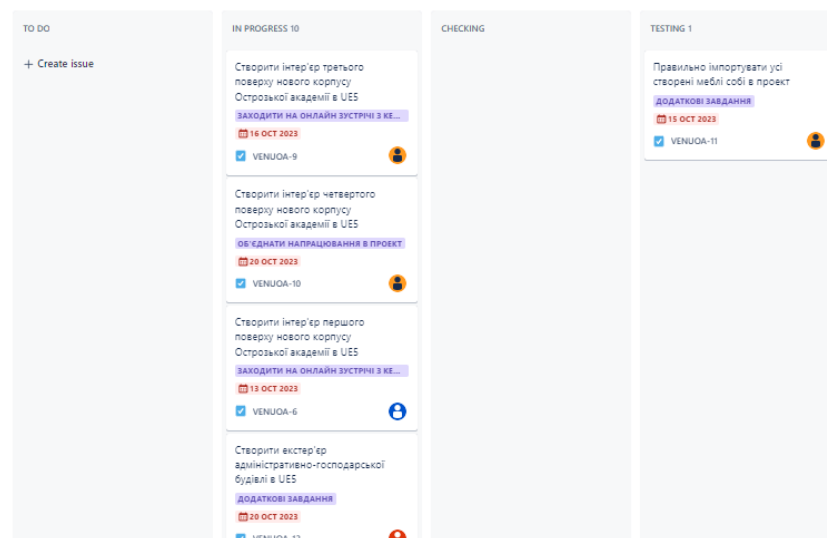


Рис.2.2. Демонстрація дошки Scrum

Джерело: розроблено автором

Наступним етапом була оптимізація робочого процесу через налаштування Workflow.

Це дозволило нам підвищити ефективність та контроль над кожним етапом процесу розробки, а також спростило спілкування та співпрацю всієї команди під час виконання завдань.

Оптимізація робочого процесу у Jira Software дозволила нам також легко відстежувати прогрес кожного завдання, контролювати строкові обмеження та уникати затримок у виконанні проєктів. Такий підхід сприяв підвищенню продуктивності та якості наших розробок.

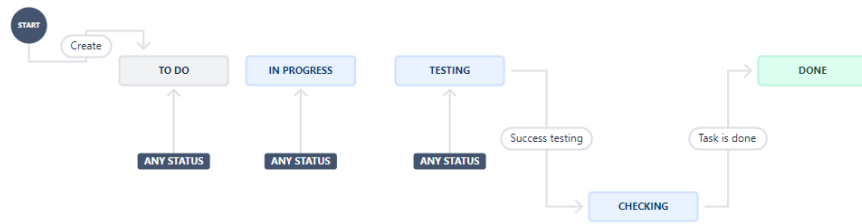


Рис.2.3. Налаштований Workflow згідно вимог команди

Джерело: розроблено автором

На фото відображено, як кожне завдання пройшло через певні кроки, щоб досягти статусу «DONE». Завдання не може бути завершеним повністю не пройшовши всі етапи розробки, тобто завдання має пройти як процес його виконання так і процес тестування, а потім вже перевірка та завершення.

Ця структура дала змогу частіше і якісніше робити перевірку завдання на його правильність та загальну функціональність.

Також кожне завдання у процесі розробки і перевірки ми позначали різними типами:

- **Epic** - Епіки відстежують колекції пов'язаних помилок, історій і завдань.
- **Bug** - Помилки відстежують проблеми чи помилки.
- **Story** - Історії відстежують функціональність або функції, виражені як цілі користувача.
- **Task** - Завдання відстежують невеликі окремі частини роботи.
- **Subtask** - Підзавдання відстежують невеликі фрагменти роботи, які є частиною більшого

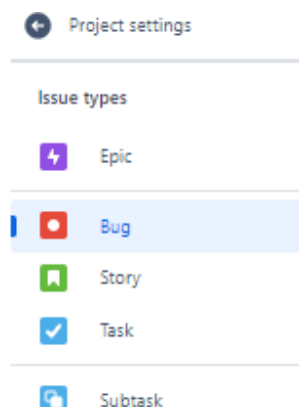


Рис.2.4. Демонстрація різних типів для завдань у проекті

Джерело: розроблено автором

2.2. Опис технологічного стеку

Перед початком роботи над проектом ми вибрали для використання такі програми як :

- Photoshop-це графічний редактор, призначений для редагування та обробки растрових зображень. Програма надає широкий набір інструментів для творчого дизайну, ретуші, маніпулювання зображеннями та створення графічних проектів.[5]
- Illustrator- це векторний графічний редактор, створений для створення та редагування векторних зображень. Програма призначена для професійного дизайну та має ряд функцій, спрямованих на роботу із векторною графікою..[6]
- Blender-це вільний та відкритий графічний редактор і програма для 3D-моделювання, анімації, рендерингу, композитингу, редагування відео та інших творчих задач.[13]

Також було використано багато доповнень для Blender такі як :

- Blender KIT-це онлайн-ресурс та додаток, які надають користувачам Blender доступ до різноманітних ресурсів, таких як моделі, текстури, матеріали, HDR-зображення, та інші елементи, які можна використовувати в їхніх проектах.[4]
- Booltool-може надавати різні додаткові функції для булевих операцій, такі як автоматичне видалення внутрішніх граней після об'єднання об'єктів, автоматичне коректне замкнення отворів, виправлення топології та інші зручності для моделювання.[12]
- Node Wrangler-це розширення спрощує роботу з вузлами, включаючи роботу з текстурами та зображеннями в графічному вузловому редакторі Blender.

Для розробки проекту ми обрали програмне середовище Unreal Engine 5

Unreal Engine 5- це ігровий рушій призначений для створення високоякісних ігор та візуалізаційних проектів і вирізняється численними технологічними інноваціями.

Також було використано декілька доповнень для Unreal Engine 5, такі як :

- Quixel bridge- це інструмент візуалізації та перегляду контенту для текстур і 3D-моделей, який інтегрований із продуктами Quixel, такими як Quixel Megascans та Quixel Mixer. Цей інструмент від Quixel, який був незалежним продуктом і до певного часу був доступний безкоштовно.[10]
- Realsky - це інструмент призначений для створення та візуалізації реалістичного неба у сцені проекту.
- Geometry Script - це плагін Unreal Engine, що містить набір бібліотек функцій, які надають можливість створювати та редагувати геометрію сітки за допомогою Blueprints.[16]
- Blueprint - це система візуального програмування, яка дозволяє створювати складну логіку та інтерактивні елементи гри без необхідності в програмуванні на мовах, таких як C++ або Python. У Blueprint ви можете створювати і керувати об'єктами, анімацією, звуками, фізикою, штучним інтелектом та багато іншого. Це інтуїтивний інтерфейс, який дозволяє розробникам швидко та ефективно створювати складну логіку гри, переглядати та налагоджувати її у реальному часі.[31]

2.3. Збір інформації та її підготовка

Хід роботи:

1. Опрацювати інформацію про поставлену задачу .
2. Пошук схем Адмінбудівлі та прилеглої території
3. Зробити фото для візуального розуміння як все має виглядати.
4. Підготовка до початкового етапу розробки.

Опис виконання:

В даному розділі ми зайнялися пошуком схем будівель Нового корпусу, щоб потім використовувати для подальшої розробки. Схеми також потрібні для дотримання розмірів та правильності побудови будівель.

Для нас були потрібні схеми адміністративної будівлі які дали мені загальне розуміння розмірів будівлі.

Схеми адмінбудівлі (Рис 2.5) можуть вказувати на плани або схеми планування будівлі. Такі схеми можуть містити інформацію про розташування кабінетів, робочих зон, приймальних, конференц-залів, туалетів, кухонь, ліфтів та інших приміщень у будівлі.

Аналіз планування будівлі дозволяє забезпечити оптимальне розміщення компонентів нашого проекту, враховуючи внутрішню логістику та потреби користувачів приміщень. Ми можемо врахувати розташування робочих просторів, зон відпочинку, зон для спілкування та інші аспекти, які сприятимуть ефективній роботі та зручності використання простору.

Крім того, наявність схем допомагає у візуалізації та розумінні структури будівлі, що може бути корисним при комунікації з іншими учасниками проекту, замовниками та іншими зацікавленими сторонами. Це сприяє точній інтерпретації та дотриманню розмірів та правильності побудови будівлі на ранніх етапах розробки.

Крім того, наявність схем допомагає у візуалізації та розумінні структури будівлі, що може бути корисним при комунікації з іншими учасниками проекту, замовниками та іншими зацікавленими сторонами. Це сприяє точній інтерпретації та дотриманню розмірів та правильності побудови будівлі на ранніх етапах розробки.

Детальний аналіз планування дозволяє врахувати не тільки функціональні, але й естетичні аспекти будівлі. Оптимальне розміщення вікон для забезпечення природного освітлення, використання матеріалів, які відповідають екологічним стандартам, та забезпечення належного рівня вентиляції та звукоізоляції – все це є важливими складовими успішного проекту.

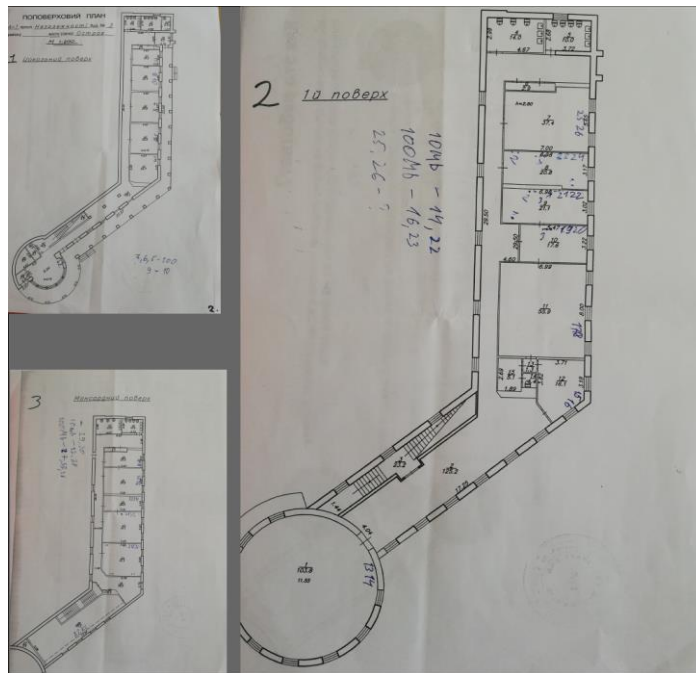


Рис.2.5. Схеми поверхів адмінбудівлі НаУОА.

Джерело: розроблено автором

Також, через кілька днів ми провели фотофіксацію всієї території нового корпусу для подальшого розуміння того, як саме має виглядати майбутній об'єкт. Цей процес фотозйомки дозволив нам отримати повний обсяг інформації щодо теперішнього стану та характеристик території, а також слугуватиме важливою вихідною точкою для подальшого проектування та вдосконалення простору (Рис 2.2).

Завдяки детальним фотографіям ми можемо краще оцінити умови на місцевості, такі як рельєф, наявність зелених насаджень та існуючі будівлі, що потребують інтеграції або обмеження. Це дозволяє знизити ризики помилок у плануванні та проектуванні, забезпечуючи більш точне та реалістичне уявлення про майбутній об'єкт.

Фотографії також слугують надійною документацією, яка може використовуватися для подальшого моніторингу та порівняння з поточним станом території під час і після завершення будівництва. Це допомагає виявляти можливі відхилення від плану та оперативно вживати коригувальні заходи.



Рис.2.6. Фото адмінбудівлі та прилеглої території НаУОА.

Джерело: розроблено автором

2.4. Створення 3D моделей в Blender

2.4.1. Налаштування програмного середовища

Хід роботи:

1. Налаштування програмного середовища **Blender** для зручності виконання роботи
2. Завантаження додаткових інструментів для полегшення роботи.

Опис виконання:

Щоб робота з Blender стала ще більш привабливою та зручною, ми налаштували інтерфейс (Рис.2.3) так, щоб кожен користувач бачив, що відбувається на екрані та як це впливає на кінцевий результат. Маючи можливість спостерігати за власною роботою у режимі реального часу, ми надаємо користувачам можливість миттєво бачити, як їх ідеї перетворюються у віртуальну реальність.

Поza тим, Blender має безліч додаткових плагінів і інструментів, що роблять роботу ще більш гнучкою та розширюють її можливості. Багато талановитих користувачів розробляють свої власні додатки та розширення, щоб спрос-

тити конкретні завдання або надати нові можливості. Такий підхід відкриває безмежні горизонти для креативності та інновацій, сприяючи взаємодії та обміну ідеями в спільноті Blender.

Крім того, важливо відзначити, що Blender є інструментом, який активно розвивається завдяки великій та дружній спільноті користувачів. Кожен може приєднатися до цієї спільноти, де досвідчені користувачі діляться своїм досвідом, порадами та творчими підходами. Це робить Blender не просто інструментом для роботи, але і платформою для навчання та обміну знаннями.

Додатково, багатофункціональність Blender дозволяє використовувати його для різноманітних проектів, починаючи від створення ігор і анімаційних фільмів до дизайну об'єктів та архітектурного моделювання. Такий універсальний підхід робить Blender популярним вибором як серед початківців, так і серед досвідчених професіоналів.

У цьому динамічному середовищі кожен користувач має можливість вдосконалювати свої навички, впроваджувати нові ідеї та брати участь у творчому процесі, що робить користування Blender захопливим і заохочує до постійного вдосконалення своєї майстерності.[11]

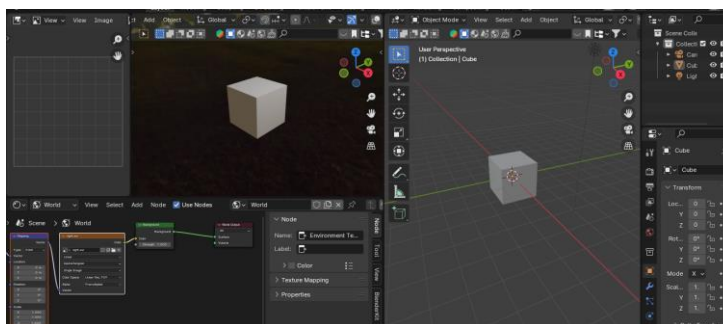


Рис.2.7. Візуальний вигляд інтерфейсу програмного середовища Blender.

Джерело: розроблено автором

2.4.2. Створення загальний вигляд фасаду з дотриманням зовнішнього вигляду та розмірів

Хід роботи:

1. Створення загального вигляду фасаду.
2. Додавання конструктивних елементів.
3. Розробка додатних елементів.

4. Дотримуючись розмірів всіх частин фасаду розбити вагомі частини фасаду для подальшого застосування у розробці.

Щоб створити загальний фасад, я використовував схеми, щоб врахувати розмір і геометрію будівлі. Я також додав багато структурних елементів, які раніше спроектував сам (вікна (кілька типів), вхідні двері, колони та арки).

Ми також розробили додаткові елементи для подальшого використання (сходи та квітник). Закінчивши деталізацію фасаду на цьому етапі, я припинив розробку фасаду і почав розробку паркової зони.

Після деталізації фасаду адмінбудівлі (Рис.2.8) та внесення деяких правок, до фасаду будівлі було додані вікна кількох видів, також було додано елементи які розташовані на даху і були внесені деякі конструктивні елементи.

Були додані різні виступи також додано арки, різні решітки, додавання деяких виступів на фасаді та розробки деяких додаткових елементів, після проведення цих робіт результат був наступним:

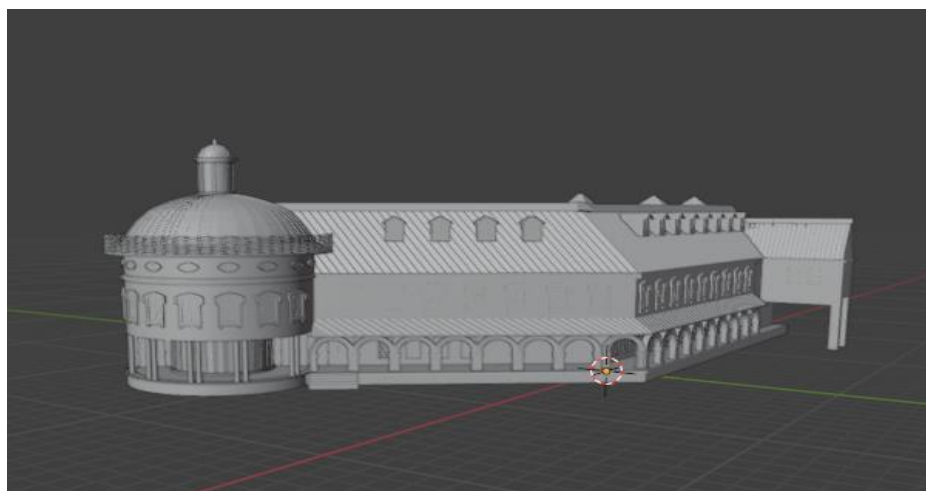


Рис.2.8. Результат розробки фасаду адмінбудівлі НаУОА в Blender.

Джерело: розроблено автором

2.4.3. Деталізація фасаду будівлі та створення паркових зон та дворової частини в Blender

Хід роботи:

1. Деталізація фасаду та внесення змін у конструкцію.
2. Створення паркової зони
3. Створення задньої дворової частини.

4. Створення різних елементів зовнішньої частини двору.
5. Імпортування різних додаткових елементів до фасаду будівлі.
6. Розробка різних дрібних деталей для застосування до фасаду.

Опис роботи :

Ми зосередилися на розробці та деталізації фасаду будівлі та прилеглого двору. Основне завдання полягало в тому, щоб створити високоякісні візуалізації архітектурних елементів та оточуючого середовища, надаючи їм реалістичний та привабливий вигляд.

Дизайн фасаду був значно змінений через невідповідність розмірів і відмінності між реальним фасадом і фасадом, який я спроектував.

Для заднього двору я використовував фотографії, зроблені раніше. На основі цих фотографій я відтворив все, що знаходиться в глибині саду. Там була побудована господарська будівля (котельня) і деякі декоративні елементи навколо будівлі. Крім того, було зроблено мінімальну розмітку для паркування автомобілів.

Для зовнішнього вигляду було додано кілька елементів (бар'єри, стовпчики та тротуарна розмітка), що прилягають до адмінбудівлі.

На фасаді також були імпортовані додаткові елементи, які значно збільшили естетичний вигляд будівлі.

Ці зміни не лише покращили зовнішній вигляд будівлі, але й створили комфортніше та привітніше середовище навколо неї. Тепер будівля виглядає не лише як споруда, але і як вираз мистецтва, завдяки уважній роботі над деталями та вибіркового використанню декоративних елементів.

Велика увага також приділялася освітленню та тінюванню, щоб кожна деталь виглядала натурально в будь-якому освітленні. Це дозволило створити не лише зовнішній вигляд будівлі, але і відтворити атмосферу реального середовища.

Завдяки використанню 3D-моделей, ви зможете віртуально досліджувати кожен елемент фасаду з будь-якого кута зору, дозволяючи отримати повний об-

сяг його краси та дизайну. Такий підхід до моделювання не лише поліпшує візуальний ефект, але і дозволяє архітекторам та дизайнерам максимально точно відтворити задуману концепцію в реальному просторі.

Наступним етапом було створення заготовок для додаткових дрібних деталей, які будуть використовуватися в майбутньому. Цей підхід дозволяє нам забезпечити високий рівень консистентності та деталізації у всьому проекті.

Заготовки для дрібних деталей включають в себе елементи, такі як (Рис.2.6) декоративні лавочки, шлагбаум, забор, клумби, двері стилізовані підвіконня та інші архітектурні акценти. Вони ретельно працюються в 3D-середовищі, щоб забезпечити їхню сумісність та гармонійне вплитання зі загальним дизайном фасаду.

Ці заготовки створюють основу для подальших додатків і деталей, дозволяючи легко внести зміни та адаптувати дизайн у майбутньому. Крім того, вони служать як платформа для творчості, надаючи можливість експериментувати з різними елементами та стилями перед їх остаточним впровадженням у проект.

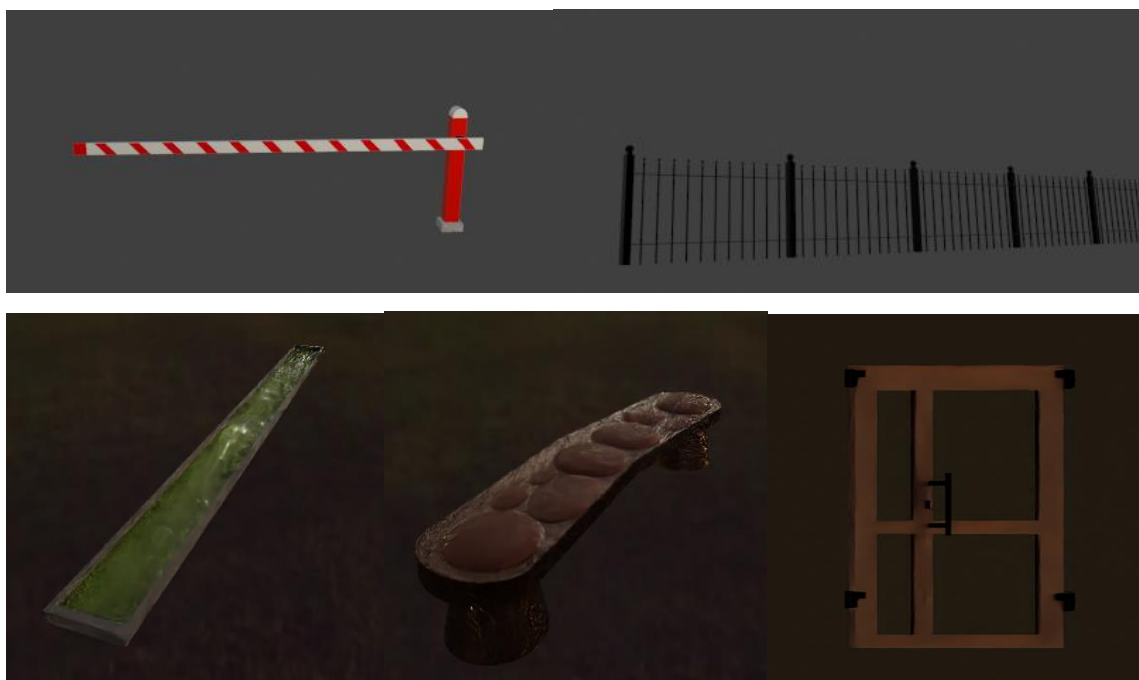


Рис.2.9. Зовнішній вигляд створених об'єктів таких як: шлагбаум, паркан, клумба , лавочка , двері, які розташовані на території НаУОА

Джерело: розроблено автором

Цей підхід до створення заготовок для дрібних деталей відкриває широкі можливості для індивідуалізації та персоналізації дизайну. Завдяки використанню заготовок, ми можемо легко експериментувати з різними варіантами інтер'єрів та екстер'єрів, додаючи або модифікуючі деталі за потребою.

Окрім того, цей етап готує нас до можливих змін та розвитку проекту в майбутньому. Нові ідеї, що можуть виникнути під час роботи над іншими частинами будівлі, легко інтегруються завдяки гнучкості заготовок. Це важливо для того, щоб забезпечити консистентність та високу якість дизайну на всій площині будівлі.

2.4.4. Створення матеріалів та шейдерів та застосування до моделей

Хід роботи:

1. Створення перших варіантів матеріалів .
2. Розробка та пошук правильних текстур для подальшого застосування.
3. Вибір способу реалізації для створення застосування матеріалів.
4. Внесення певних правок.
5. Налаштування матеріалів під певне середовище(кольорова гамма та відтінки).

Опис роботи:

Для створення матеріалів мені довелося переглянути фотографії кольорової гами фасаду будівлі та прилеглих частин двору, а також те, як виглядають різні елементи поруч.

Побачивши, які текстури де використовуються, я почав підбирати відповідний набір матеріалів і способи їх застосування.

У Blender матеріали та шейдери застосовуються через систему вузлів, що трохи полегшує реалізацію. Для розробки текстур також були використані додаткові сервіси (різні сервіси з готовими прикладами текстур). Крім того, текстури для дерева, бруківки та трави були отримані з інтернету.

Вузлова система спрощує роботу в реалізації та надає розширені можливості при внесенні змін. Кількість реалізованих матеріалів досить велика, тому

ми не будемо перераховувати їх всі, але для прикладу наведемо схему для одного матеріалу (Рис.2.10). У цьому прикладі це бруківка:

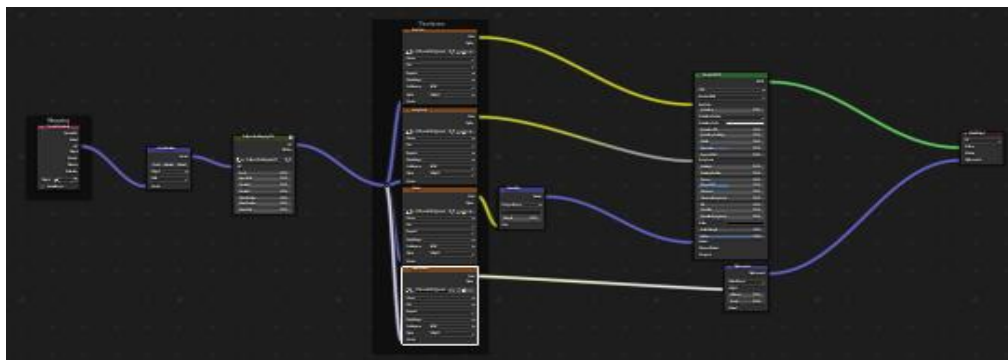


Рис.2.10. node матеріалу бруківки.

Джерело: розроблено автором

Після того, як всі матеріали були розроблені, ми почали застосовувати їх до моделі та різних елементів. Спочатку я створив фон для середовища, щоб покращити зовнішній вигляд матеріалів, а також налаштував основний рушій для рендерингу, який називається Cycles- це вбудований рендерер у програмі Blender. Він використовується для створення реалістичних зображень з тривимірних сцен, які ви створюєте у Blender.

Цей рушій має кілька переваг над іншими:

- реалістичність світла та тіней: Cycles використовує метод трасування променів (path tracing), що дозволяє наблизитися до реалістичної моделі поведінки світла. Це дозволяє отримати відмінну відтворення тіней, відблисків і освітлення.[14]
- гнучкість матеріалів: Cycles підтримує широкий спектр матеріалів, включаючи різноманітні текстури, прозорі матеріали, відблиски і відтінки.
- можливості в роботі з вузлами: Cycles надає багато можливостей для використання вузлів (node-based), що дозволяє складати різноманітні налаштування для матеріалів, текстур та освітлення.[14]
- підтримка GPU: Cycles підтримує використання графічних процесорів (GPU), що може значно прискорити час рендерингу, особливо на потужних відеокартах.
- інтеграція з Blender: Оскільки Cycles є частиною Blender, він інтегрований із іншими інструментами та можливостями програми, що робить роботу з ним зручною та продуктивною.[14]

Побачивши початкові результати після застосування текстур, я почав налаштовувати відтінки матеріалів і різні ефекти, щоб досягти бажаного результату в зовнішньому вигляді моделі і різних елементів. Використовуючи вузловий редактор у Blender, я експериментував із параметрами, такими як колір, блиск, прозорість та інші, для створення виразного і реалістичного образу.

Особливу увагу я приділяв взаємодії світла з поверхнею об'єктів. Змінюючи налаштування відбиття та розсіювання світла, я досягав натурального вигляду тіней та відблисків. Використання карт високої деталізації та нормалей також сприяло створенню деталізованих текстур, що підсилює візуальний ефект.

Крім того, я тестував різні типи освітлення, щоб підібрати той, який найкраще відтворює атмосферу, яку я хотів досягти. Експериментуючи з напрямками та інтенсивністю світла, я створював варіанти освітлення для різних частин моделі, підкреслюючи їхні особливості.

Весь цей процес налаштувань та експериментів допоміг мені досягти бажаного рівня реалізму та естетики у візуальному вигляді моєї 3D-моделі. Щоразу, коли адаптував параметри матеріалів та освітлення, я відкривав нові можливості для покращення деталей і забезпечення виразного відображення концепції.

Особливу увагу я приділяв дрібницям, таким як мікро-текстури та нюанси кольорів, щоб забезпечити реалістичний вигляд кожної поверхні. Ретельно налаштовані відтінки, вигляд витікання світла та відтворення матеріалів створюють відчуття взаємодії з об'єктом у віртуальному просторі.

Кожна зміна в параметрах впливала на загальний вигляд і атмосферу сцени, дозволяючи мені долучити особистий та художній підхід до створення 3D-моделі. Цей процес не лише вдосконалив мою технічну майстерність, але й розвивав мою творчість та здатність виражати свої ідеї через віртуальний простір.

Загалом, розробивши матеріали та застосувавши їх до моделей, отримав наступний проміжний результат: вірно передані властивості поверхні, відтінки та текстури створюють враження високої деталізації та реалістичності. Кожен

елемент виглядає не лише як частина моделі, але й як окремий об'єкт з власною текстурою та характером.

Матеріали вдало підібрані так, що вони гармонійно взаємодіють між собою та створюють виразний зовнішній вигляд об'єкта. Більшість текстур передають деталізованість, що підкреслює роботу над реалізмом. Відтінки матеріалів виразно передають глибину та структуру поверхонь.

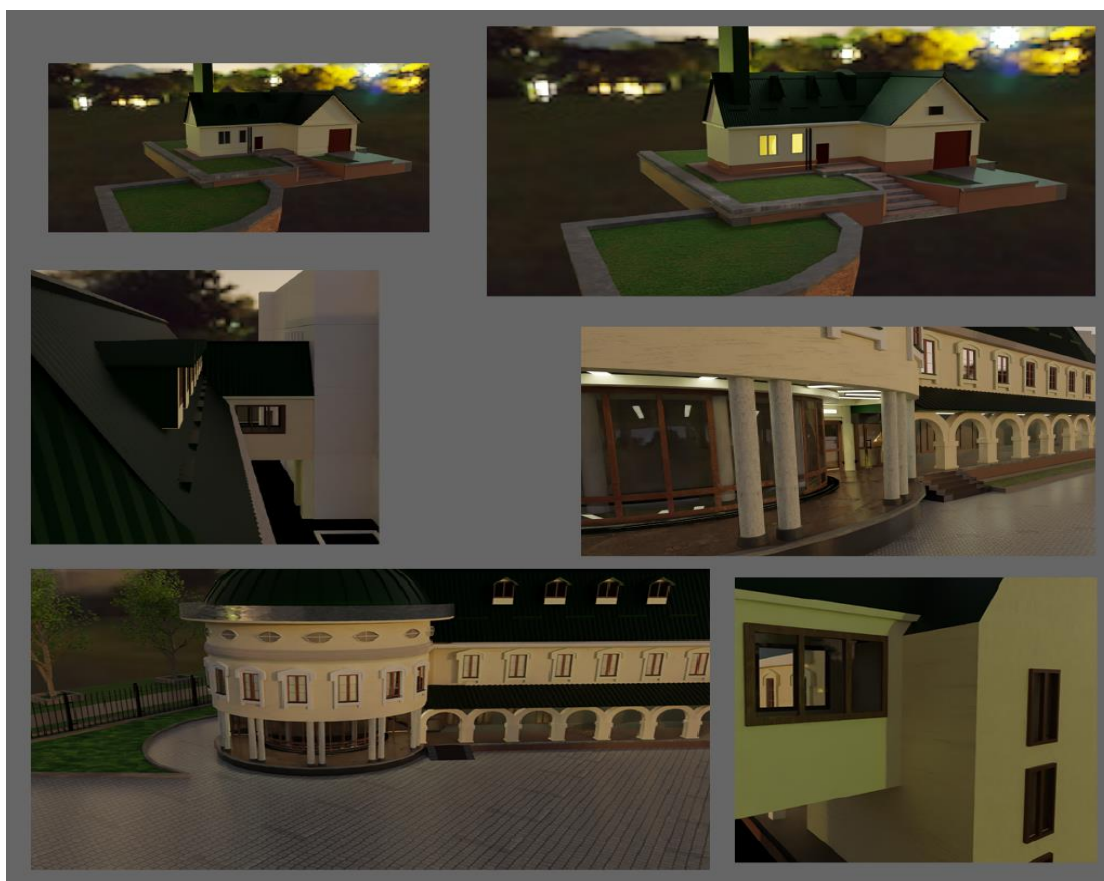


Рис.2.11. Результат текстурування адмінбудівлі НаУОА та прилеглої території.

Джерело: розроблено автором

Більшість використаних текстур передають деталізованість і структуру об'єкта, що додає відчуття реальності. Вони відображають відмінності в матеріалах, будь то мікроскопічні вибійки на текстурі каменю чи теплі відтінки на поверхні дерева.

Відтінки матеріалів, виважено підібрані, виразно передають глибину та структуру поверхонь. Колір взаємодіє з освітленням, змінюючи світлові відтінки та тіні в залежності від перспективи і напрямку світла, що додає динамічності об'єкту.

В результаті, деталізований підбір матеріалів і вдалий вибір текстур сприяли створенню 3D-моделі, яка не лише виглядає реалістично, але й вражає своєю естетикою та вишуканістю.

2.4.5. Внесення правок у роботу та додаткова розробка певних елементів
Хід роботи:

1. Розробка додаткових елементів для додавання додатково до моделі.
2. Внесення додаткових правок у проєкті

Опис роботи:

Загалом, у цьому розділі були розроблені додаткові елементи, які будуть використані в проєкті. Першими елементами були огорожа та перила для дворової території та нова лінія розмежування. По-друге, були спроектовані лавки та клумби. По-третє, було створено певну кількість статуй (ліхтарів) і додано невеликі ліхтарі, щоб краще демонструвати територію.

Також додали текстуру бруківці на головній площі та висадили зелені насадження, щоб доповнити картини.

За рекомендацією керівника ми також створили міську пішохідну доріжку та асфальтовану ділянку, а також додали дерева та бордюри в міській зоні.

Ці зміни та редагування покращили зовнішній вигляд та додали більше деталей до проєкту.

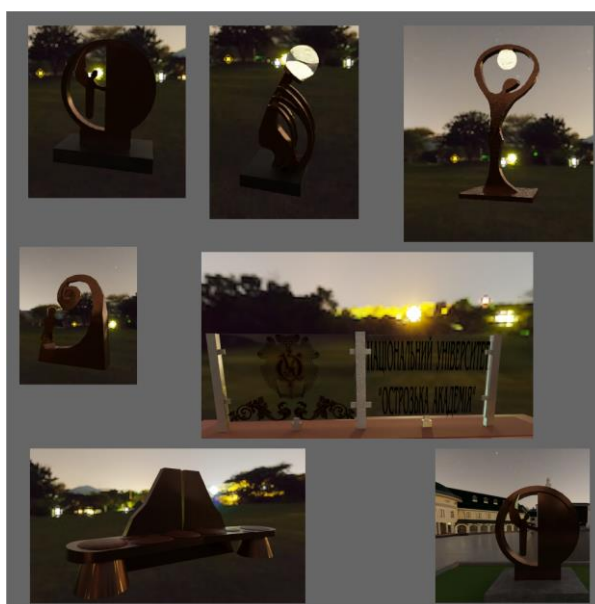


Рис.2.12. Результат правок та додаткових елементів адмінбудівлі НаУОА .

Джерело: розроблено автором

2.4.6. Підготовка до рендеру та налаштування сцени для фінального рендеру

Хід роботи:

1. Налаштувати сцену для фінального рендеру.
2. Підготувати освітлення та ракурси для рендеру.
3. Створити рендер сцени.

Опис роботи:

Після того, як ми завершили всі етапи розробки моделі та застосування текстур і матеріалів, ви можете приступити до дизайну та налаштування вашої сцени.

Сцена, яку я створив, пройшла кілька етапів налаштувань для досягнення бажаного вигляду. Спочатку я обрав фон для оточення, щоб забезпечити правильне та естетичне освітлення всіх об'єктів. Я вніс зміни до фону під час нанесення матеріалів, щоб покращити його візуальний вигляд.

Освітлення сцени виявилось ключовим етапом, оскільки воно визначало об'ємність зображення та реалістичність. Я використовував різні джерела світла, такі як ліхтарі, лампи та статуї з ліхтарями, для створення різноманітного і динамічного освітлення.

Після цього додав камеру та вибрав різні ракурси для створення різноманітних кутів зйомки. Вибір ракурсів був важливим кроком для створення цікавих та ефективних зображень.

Щоб підготувати сцену до рендерингу, я використовував рушій Cycles, встановлюючи необхідні налаштування, такі як кількість семплів для визначення якості рендерингу.

Після коригувань налаштувань я здійснив фінальний рендер для першого показу(Рис.2.13). Показавши результати керівнику, я отримав додаткові пропозиції та внесення змін для досягнення оптимального результату.



Рис.2.13. Зовнішній вигляд адмінбудівлі НаУОА та прилеглі території з різними додатковими елементами.

Джерело: розроблено автором

2.5. Інтеграція 3D моделей в Unreal Engine

2.5.1. Підготовка до інтеграції в Unreal Engine

Інтеграція 3D моделей в Unreal Engine є важливою частиною процесу розробки гри чи віртуальної реальності. Unreal Engine підтримує різні формати 3D моделей, такі як FBX, OBJ, і інші.[23]

При розробці виявилось, що основні елементи краще робити в ігровому рушії, а дрібні деталі або дуже деталізовані об'єкти розробляти у Blender, а також багато елементів було вже зроблено тому робити їх ще раз не логічно.

Коли розробляєте різні елементи у Blender, важливо враховувати поняття нормалізації, що визначає правильне розташування нормалей в об'єкті.

Нормалі - це вектори, які перпендикулярні до поверхні об'єкта і вказують напрямком, у якому світло повинно взаємодіяти з поверхнею.

Правильно налаштовані нормалі грають важливу роль у візуальному вигляді 3D-моделі. Коли нормалі налаштовані вірно, світло вірно взаємодіє з поверхнею, створюючи реалістичні тіні та відблиски.

З іншого боку, неправильне розташування нормалей може призвести до артефактів, таких як неправильне освітлення або надто виражені тіні.

У Blender ви можете використовувати інструменти для визначення та редагування нормалей, такі як "Recalculate Outside" або "Flip Normals", щоб забезпечити правильне розташування. Це особливо важливо при створенні моделей, які будуть використовуватися в рендерінгу або анімації, де деталізований реалізм та коректне освітлення відіграють ключову роль у візуальному сприйнятті.[29]

Правильна нормалізація є необхідною для забезпечення коректної взаємодії світла з 3D-моделлю. Коли нормалі налаштовані вірно, вони допомагають визначити, яким чином світло відобразиться від кожної поверхні, що створює ефективне візуальне подання матеріалів і форм об'єктів.

Важливо також враховувати, що нормалі можуть впливати на різні ефекти, такі як заломлення світла (refraction), освітлення з текстурами (bump mapping) та інші аспекти візуалізації. Вірно встановлені нормалі дозволяють досягти більш високої реалістичності та деталізації у зображенні.

У разі складних об'єктів або моделей, що складаються з багатьох поверхонь, правильна нормалізація може вимагати уважної роботи з кожною частиною моделі.

Такий підхід сприяє підвищенню якості рендеру та покращенню загального візуального враження від створеної 3D-моделі.

2.5.2. Налаштування програмного середовища Unreal Engine 5

Хід роботи:

1. Налаштувати інтерфейс для зручнішого використання.
2. Встановлення додаткових аддонів в Unreal Engine 5.

Опис роботи:

Для зручної розробки проекту в Unreal Engine 5 налаштували інтерфейс під себе для зручнішого використання. Також додав кілька аддонів для простішої розробки моделей .

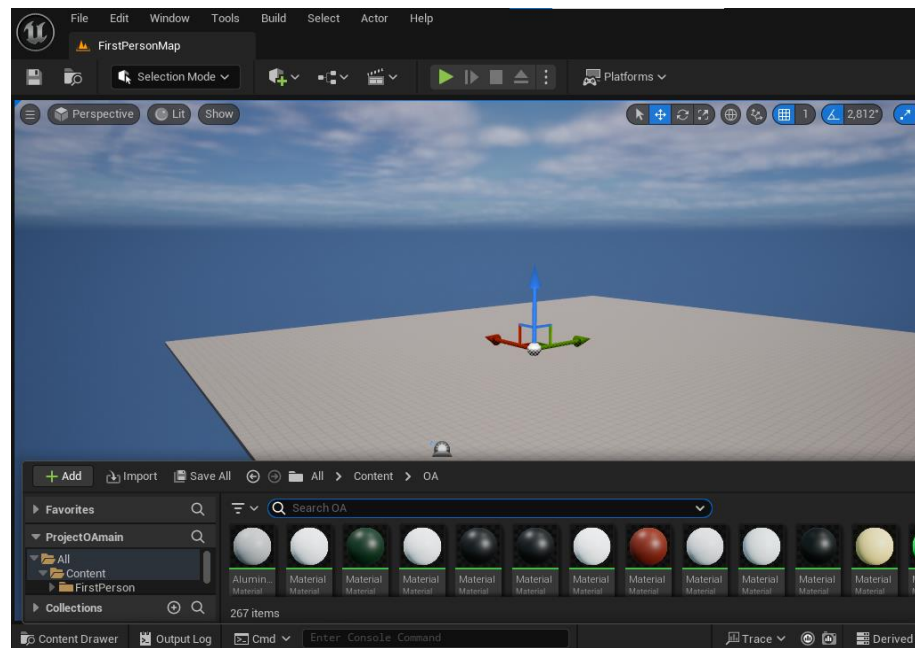


Рис.2.14. Візуальний вигляд інтерфейсу програмного середовища Unreal Engine 5.

Джерело: розроблено автором

2.5.3. Підготовка моделі для інтеграції в Unreal Engine

Хід роботи:

1. Перевірка моделі.
2. Вибір формату для інтеграції.
3. Підготовка моделі для інтеграції.
4. Перевірка текстур.

Опис роботи:

Перед інтеграцією моделі ми робили перевірку моделі які елементи ми можемо зробити в рушії ,а які можемо зробити в Blender. Це дасть змогу правильно та без додаткових дій правильно інтегрувати модель.[11]

Інтеграція 3D моделей в Unreal Engine може бути важливою частиною процесу розробки гри чи віртуальної реальності. Unreal Engine підтримує різні формати 3D моделей, такі як FBX, OBJ, і інші.

Наступне це вже сама підготовка інтеграції моделі у проект. Після завершення розробки 3D-моделі, я переконався, що вона збережена у підтримуваному форматі, такому як FBX або OBJ. Звернув увагу на деталі збереження, щоб упевнитися, що модель коректно взаємодіє з іншими програмами та платформами.

Також віддавав увагу налаштуванню текстур та матеріалів на моделі, переконавшись, що кожен елемент виглядає так, як було задумано. Налагоджував параметри для досягнення виразності та реалістичності, впевнившись, що кольори, відтінки та текстури відтворюються точно. Це сприяло створенню моделі з вишуканою та докладно деталізованою зовнішністю.

У проєкті для інтеграції моделі ми використовуємо Fbx формат який згодом будемо інтегрувати у проєкт.

Коли ми розробляли модель в Blender я робив спершу текстури які підтримує лише дана програма. Unreal Engine 5 багато матеріалів не підтримує, тому для того щоб візуально було простіше я на кожен сторону моделі наніс звичайні базові кольори, які в рушії добре видно і це дає краще розуміння де який матеріал розташований на моделі.

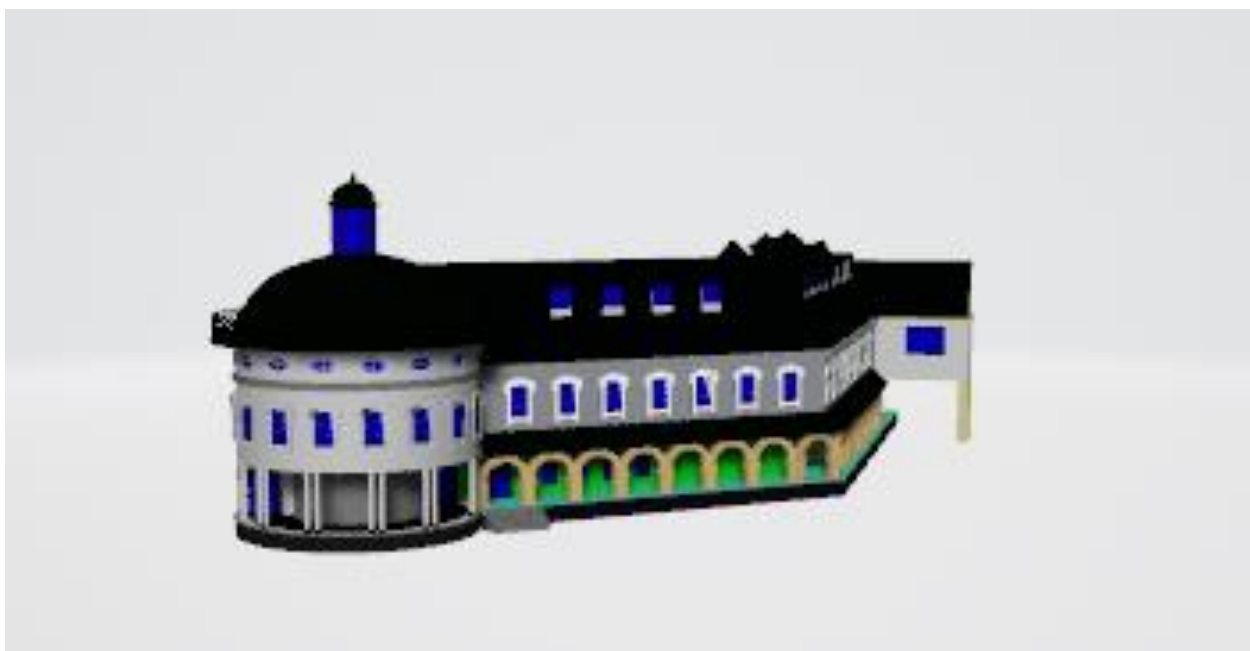


Рис.2.15. модель Адмінбудівлі в Fbx форматі .

Джерело: розроблено автором

2.5.4. Імпорт 3D моделей в Unreal Engine 5

Хід роботи:

1. Підготовка для імпорту моделі з Blender в Unreal Engine 5.
2. Імпорт моделі в Unreal Engine 5.

Опис роботи:

Після підготовки інтеграції моделей адмінбудівлі та різних елементів, можна робити імпорт в Unreal Engine 5.

Для цього нам потрібно виконати наступні кроки:

При інтеграції моєї 3D-моделі в Unreal Engine, я відкрив вікно "Content Browser" для зручності управління ресурсами. У відповідній папці обрав місце для збереження моделі в двигуні гри.

Далі, взаємодіючи з програмою, я використовував контекстне меню, клацнувши правою кнопкою миші, і обрав опцію "Import". Це відкрило можливість вибору та імпорту 3D-моделі безпосередньо в проект Unreal Engine.

За допомогою цього підходу я впевнився, що моя модель ефективно інтегрується в робочий процес Unreal Engine, готуючись до подальших етапів розробки та використання в грі.

Після імпорту моделі для кращого розуміння створюємо папку для всіх імпортованих елементів.

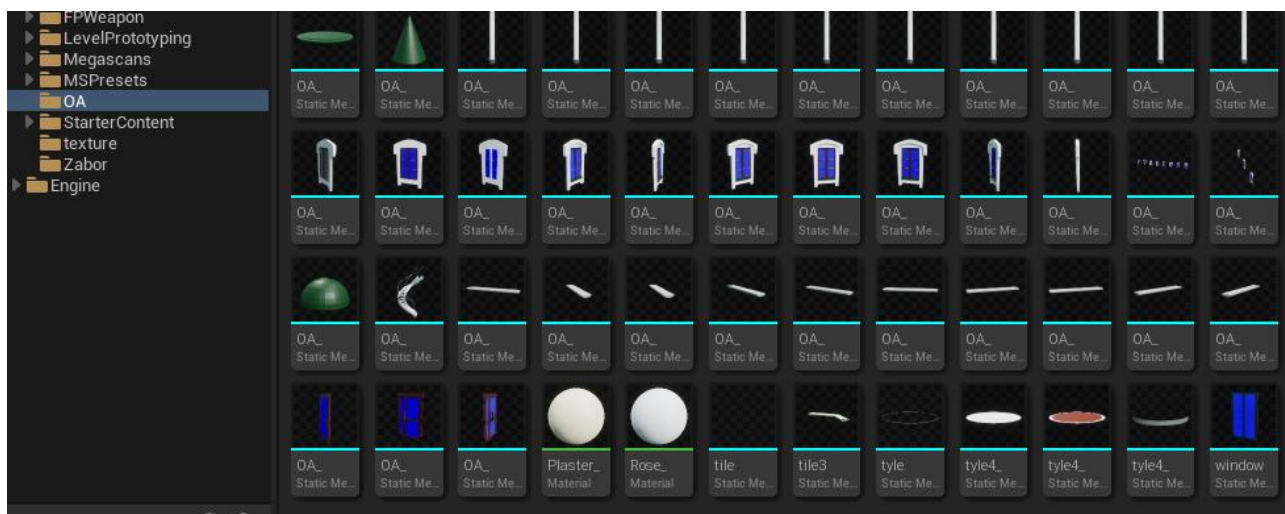


Рис.2.16. Імпортовані елементи.

Джерело: розроблено автором

2.5.5. Розміщення моделі у головній сцені проекту

Хід роботи:

1. Розміщуємо модель у ігровій сцені .
2. Налаштування колізії моделі.

Опис роботи:

Коли ми імпортували модель у проект налаштуємо її для розміщення у сцені ігрового світу:

Під час взаємодії з моєю 3D-моделлю у віртуальному просторі, я просто перетягнув її в вікно світу, розмістивши на сцені Unreal Engine. Використовуючи інструменти переміщення, обертання та масштабування, я відредагував положення та орієнтацію моделі згідно з моїми потребами та концепцією сцени.

Додатково я звернув увагу до колізії об'єктів, намагаючись забезпечити більш зручне та реалістичне взаємодію моделі з іншими об'єктами на сцені. Це важливий аспект у моделюванні, оскільки він впливає на поведінку та взаємодію об'єктів у грі чи віртуальному середовищі. Зробивши належні налаштування колізії, я максимізував реалізм та функціональність моєї 3D-моделі в рамках вибраного проекту.

Ці дії потрібно виконати для всіх окремих елементів моделі це для того щоб краще і реалістичніше можна було взаємодіяти з будівлею.

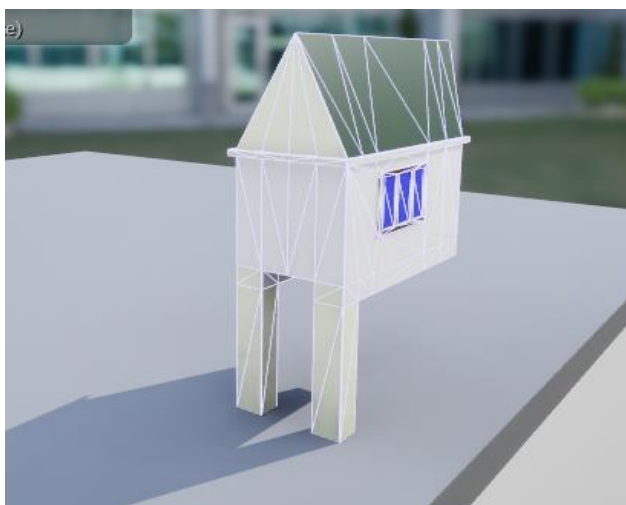


Рис.2.17. Відображення колізії.

Джерело: розроблено автором

Після виконання всіх цих дій ми отримали перший тестовий вигляд адмін-будівлі в середовищі Unreal Engine 5. Під час розгляду результатів можна відзначити успішно налаштовану колізію, що сприяє реалістичній і взаємодії об'єктів на сцені. Усі елементи правильно розташовані, враховуючи їхнє взаємодією з оточенням, завдяки інструментам рушія Unreal Engine 5.

Цей тестовий вигляд дозволяє оцінити розташування та поведінку адмінбудівлі у віртуальному просторі. Використовуючи інструменти рушія, я забезпечив оптимальну композицію та аранжування кожного елемента, щоб досягти бажаного вигляду та функціональності. Цей етап тестування важливий для виявлення можливих покращень та визначення подальших кроків у розробці адмінбудівлі в Unreal Engine 5.[8]



Рис.2.18. Стартовий варіант адмінбудівлі.

Джерело: розроблено автором

2.5.6. Деталізація адмінбудівлі та створення паркових зон з дворовою частиною

При розробці деталей фасаду в Unreal Engine, я приділив увагу створенню паркової зони та задньої дворової частини, надаючи кожній деталі відповідний вигляд. Додавання різних елементів до зовнішньої частини двору створило більш живописний та атмосферний образ.

Під час імпортування додаткових елементів до фасаду будівлі, я ретельно вибирав деталі та акценти, щоб забезпечити гармонійний та естетичний вигляд.

Розробка різних дрібних деталей для застосування до фасаду дозволила додати індивідуальний характер та унікальність кожному елементу.

Тестування та налагодження відіграли ключову роль у вдосконаленні моделі. Цей етап дозволяє визначити, наскільки коректно 3D-модель відображається та веде себе в грі.

По завершенні всіх кроків я отримав готовий варіант моделі, який можна демонструвати в тестовому показі, відображаючи усі допрацьовані та вдосконалені деталі

2.5.7. Готові результати

Перш за все, тут представлені рендери екстер'єру адмінбудівлі, що включають в себе деталізований фасад, правильно розташовані елементи та зони відпочинку. Зображення відображають ступінь деталізації текстур, виразність матеріалів та освітлення, що підкреслює важливість розробленого віртуального середовища.



Рис.2.19. Адмінбудівля та прилеглі території в Unreal Engine 5

Джерело: розроблено автором

Далі, у розділі висвітлені різні ракурси та перспективи, що відображають архітектурні особливості адмінбудівлі з різних точок зору. Це дозволяє аудиторії отримати повну картину проекту та детально розглянути архітектурні рішення.

Крім того, рендери охоплюють прилеглі території, зокрема паркову зону та задню дворову частину. Вони демонструють створення природних елементів,

інфраструктури та різноманітних деталей, що збагачують вигляд віртуального середовища.

Кожен рендер має на меті відобразити не лише архітектурні деталі адмінбудівлі, але й атмосферу та настрій створеного віртуального середовища. Зображення віддзеркалюють вдумливий підхід до створення паркових зон, розташування додаткових елементів до фасаду, а також уважність до деталей, яка виражається в кожному піднятому плитці, кожній зеленій насаженій рослині та кожній архітектурній формі



Рис.2.20. Адмінбудівля, котельня та прилеглі території в Unreal Engine 5

Джерело: розроблено автором

Завершуючи аналіз "Готових рендерів", можна підкреслити, що вони не лише становлять важливий етап у візуалізації проекту, а й є засобом взаємодії з аудиторією та клієнтом. Готові рендери стають віртуальним вікном у світ розробленого проекту, дозволяючи сприймати його в найкращому світлі та оцінювати кожну деталь з усіх можливих ракурсів.

Ці візуалізації також сприяють обговоренню та внесенню коректив, зменшуючи можливість помилок. Вони можуть бути використані для створення презентаційних матеріалів, демонструючи проект у найкращому вигляді.

Крім того, готові рендери забезпечують чітке уявлення про кінцевий результат і стандарти, які необхідно досягти. Це допомагає уникнути розбіжностей між задумами та їх реалізацією на практиці, забезпечуючи більш точне виконання робіт.



Рис.2.21. Загальний вигляд території НаУ ОА Unreal Engine 5

Джерело: розроблено автором

2.6. Створення інтерактивного віртуального середовища в Unreal Engine 5

2.6.1. Визначення концепції інтерактивності у проекті

Визначення концепції інтерактивності у проекті - це ключовий етап у розробці інтерактивного віртуального середовища, де кожен аспект має велике значення для створення приємного та зручного використання.

Перш за все, потрібно ретельно визначити, які саме типи інтеракції будуть доступні користувачам.

Це може бути рух по середовищу, взаємодія з об'єктами, активація різних механізмів або навіть спілкування з персонажами.

Наступний важливий крок - це визначення методів управління інтерфейсом. Наприклад, чи буде це за допомогою клавіатури, миші, геймпада або сенсорного екрану? Важливо, щоб користувачі відчували зручність у використанні.

Ключовим елементом є також реакція віртуального середовища на дії користувачів. Як об'єкти реагують на дотик або переміщення? Як змінюється оточення залежно від дій гравця?

Далі, важливо визначити основні механіки гри або додатку, які стануть основою інтерактивності. Це можуть бути різні геймплейні елементи або особливості дослідження віртуального світу.

Не менш важливим є також рівень взаємодії, який буде наданий користувачам. Вони будуть лише спостерігачами або матимуть можливість активно впливати на події і оточуючий світ?

Прогресія інтеракції - це ще один аспект, який потрібно врахувати. Як інтерактивність буде розвиватися протягом проекту? Чи буде вона змінюватися в залежності від рівня гри чи прогресу користувача?

Узагальнюючи, визначення концепції інтерактивності включає в себе розробку стратегії, що дозволить створити зручний та захоплюючий досвід для користувачів, в якому кожен елемент і взаємодія має своє місце і значення.

Крім вищезгаданих аспектів, важливо також враховувати адаптивність інтерактивності до різних типів користувачів.

Окрім того, необхідно враховувати емоційний аспект інтерактивності. Як користувачі відчують взаємодію з віртуальним середовищем? Чи сприяє вона відчуттю задоволення та захоплення?

Також, важливо відзначити, що інтерактивність не завжди означає складність. Простота та доступність також можуть бути ключовими перевагами, особливо коли мова йде про широкий коло користувачів з різним рівнем досвіду та навичок.

У нашому проекті я відповідаю за реалізацію різноманітних інтерактивних елементів, які сприяють збагаченню віртуального середовища університету та покращенню взаємодії користувачів з ним.

Моя робота включає в себе наступні аспекти:

- **Генерація снігу:** Розробка механіки, що дозволяє створювати та управляти сніговим покривом у віртуальному просторі, надаючи зимового настрою та атмосфери.
- **Інтерактивність з предметами:** Реалізація можливості брати у руки різноманітні предмети у віртуальному середовищі, що дозволяє користувачам взаємодіяти з ними та виконувати певні дії.
- **Зимове оточення:** Створення та втілення відповідного зимового оточення з використанням відповідних текстур, моделей та ефектів, які допомагають перенести користувачів у атмосферу зими.
- **Звукове супроводження:** Імплементация звукових ефектів, які доповнюють візуальний досвід та створюють більш іммерсивну атмосферу віртуального середовища.
- **Міні мапа:** Розробка та впровадження мінімапи, яка допомагає користувачам орієнтуватися у великому віртуальному просторі та швидко знаходити необхідні об'єкти або місця.
- **Вказівники:** Створення візуальних вказівників або підказок, що спрощують навігацію користувачів та надають їм необхідну інформацію про взаємодійні об'єкти.

Ці аспекти взаємодії з віртуальним середовищем спрямовані на створення захоплюючого та іммерсивного досвіду для користувачів, що дозволяє їм вільно взаємодіяти з цим світом та досліджувати його.

Важливо відзначити, що створення інтерактивного віртуального середовища Національного університету "Острозька академія" є складним та відповідальним завданням, яке потребує злагодженої роботи команди та використання передових технологій, таких як Unreal Engine 5 та Blender.

Реалізація різноманітних інтерактивних елементів, забезпечення комфорту та зручності для користувачів, а також створення атмосферного та реалістичного оточення - це ключові аспекти успішного втілення проекту. Завдяки вдосконаленню цих елементів ми маємо можливість надати користувачам захоплюючий

та пізнавальний досвід, який сприятиме їхньому взаємодії з університетським середовищем та сприяти подальшому залученню нових користувачів.

2.6.2. Розробка генерації снігу

Розробка генерації снігу у віртуальному середовищі університету в Unreal Engine 5 є захоплюючим завданням, яке додає реалістичності та атмосферності до оточення. Використовуючи UE5, ми можемо створити ефективні та вражаючі ефекти снігу, які допоможуть перенести користувачів у зимовий світ.

Варто зазначити це довгий і кропіткий процес, який забирає багато часу для різних варіантів створення.

Для реалізації цього завдання я розділив його на кілька етапів для розробки загальної концепції ,яку потім можна буде інтегрувати у проект. Також виявилось при розробці вийшло зробити дві версії генератора , але друга версія вийшла модернізацією першої.

Основні етапи розробки генерації снігу включають в себе:

1. Встановлення додаткового плагіну:

Для роботи з геометрією , щоб динамічні об'єкти працювали нормально я встановив плагін **Geometry Script** , він слугує основою для подальшого створення генерації снігу. Цей плагін дає набір функцій та бібліотек, що дозволяють створювати та модифікувати геометричну сітку, використовуючи Blueprints.[16]

Встановлення цього плагіну дозволяє зручно працювати з динамічними об'єктами та геометрією сцени, що стає необхідним для створення ефекту снігу.

Цей плагін (Рис.2.22) надає доступ до набору функцій та бібліотек, які можна використовувати для програмування скриптів, які створюють або змінюють геометричні об'єкти у віртуальному середовищі. Це дозволяє розробникам налаштовувати форми та розміри об'єктів, а також здійснювати їхню динамічну модифікацію, яка може бути важливою для реалістичного відображення снігу та його взаємодії з оточуючим середовищем.

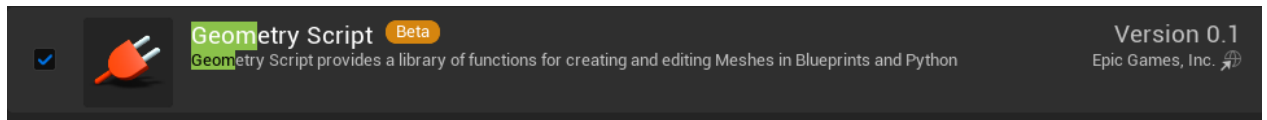


Рис.2.22. Вигляд плагіну Geometry Script

Джерело: Unreal Engine 5

2. Початкове налаштування сценарію геометрії:

Також я визначив параметри та створив основи для роботи з геометрією. Це етап дає нам розуміння як і з якими об'єктами ми будемо працювати. Далі налаштовуємо сцену для загального розуміння з чим працювати.

Під час налаштування сценарію геометрії ми також можемо проводити попередню підготовку геометричних об'єктів, таких як створення заготовок або базових форм. Це допомагає забезпечити зручне робоче середовище та зосередити увагу на основних завданнях без зайвих перешкод або непотрібних деталей.

На даному етапі ми також можемо визначити основні характеристики об'єктів, такі як їхні матеріали, текстури, освітлення та інші параметри, що впливають на їхній зовнішній вигляд та поведінку у віртуальному середовищі.

3. Створення геометричного сценарію:

Тут я розробив загальні форми та структуру сцени, яка служить основою для генерації снігу. Додав примітиви та базові фігури з якими буду працювати і надалі.

Під час створення геометричного сценарію я акцентував увагу на розробці загальної композиції та структури сцени, що відображає основні елементи та області, які будуть піддаватися впливу генерації снігу. Розміщення примітивів та базових фігур було здійснене з урахуванням географічних особливостей та ландшафту середовища, щоб максимально точно відобразити природні умови та взаємозв'язки.

Крім того, під час створення геометричного сценарію я також врахував потенційні місця найбільшої активності снігу та його можливу накопиченість. Це дозволить нам більш точно і реалістично симулювати розподіл снігу в майбутньому.

4. Робота зі сіткою та змінними:

Створивши наш Blueprint я додав параметри сітки, таких як розмір та кількість вершин. Використання цих змінних дає нам можливість збереження та маніпулювання даними сітки, наприклад, зберігання координат вершин.

Це дозволяє налаштовувати естетику та функціональність наших об'єктів, створюючи бажаний ефект в середовищі. Також, ми можемо використовувати ці параметри для динамічного зміщення, обертання та масштабування об'єктів, розширюючи можливості нашого творчого процесу і дозволяючи створювати більш складні та реалістичні сцени.

5. Використовуємо ієрархію обмежувальних томів:

Дозволяє краще організувати та управляти складними сценами в Unreal Engine 5, що сприяє підвищенню продуктивності та полегшує роботу розробників.

Це спрощує навігацію та редагування великих та складних сцен, дозволяючи розробникам швидко знаходити потрібні об'єкти та зручно взаємодіяти з ними. Більш того, ієрархічна організація об'єктів сприяє збереженню порядку та чистоти в проекті, що підвищує його скальпельність та полегшує спільну роботу команди розробників.

6. Створюємо Вузол послідовності:

У Blueprint я додаю **Sequence Node**, який дозволяє впорядковувати виконання різних дій в певній послідовності. Далі робимо перевірку чи все працює нормально, щоб зрозуміти чи на правильному ми шляху.[16]

Це особливо важливо у випадках, коли потрібно виконати кілька дій послідовно або забезпечити правильний порядок взаємодії різних компонентів програми. Після налаштування послідовності важливо перевірити її на коректність роботи, щоб переконатися, що програма виконує очікувані дії у правильному порядку та без помилок.



Рис.2.23. Додаємо Sequence Node

Джерело: Unreal Engine 5

7. Використання Raycast для визначення положення на поверхні об'єктів:

Додаю Raycast щоб визначати точне положення користувача або об'єктів у віртуальному просторі відносно поверхні інших об'єктів або об'єктів у грі. Це дає змогу правильно генерувати і шукати об'єкти з якими буде взаємодіяти генерація.[23]

Це відкриває безліч можливостей для створення цікавих інтерактивних сценаріїв, таких як виявлення зіткнень, взаємодія з об'єктами, визначення траєкторій руху, реалістичне розташування об'єктів у просторі та багато іншого. Завдяки Raycast можемо створювати динамічні та реалістичні ігрові сцени.

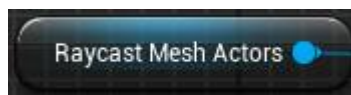


Рис.2.24. Додаємо Raycast Mesh Actors

Джерело: Unreal Engine 5

8. Використання меж об'єкта для положення початку променя:

У Blueprint визначаємо точку на поверхні об'єкта, звідки буде відправлено промінь.

Також це дозволяє точно визначити місцезнаходження та орієнтацію променя у просторі відносно конкретного об'єкта. Це дозволяє нам контролювати початкові умови для взаємодії з іншими об'єктами та елементами гри. Наприклад, можемо використовувати межі об'єкта для визначення точки старту для променя, який буде використаний для визначення шляху руху гравця або для взаємодії з ігровими об'єктами. Це дозволяє створювати складні та реалістичні взаємодії у віртуальному середовищі.

9. Розміщення сфери в точці ураження Raycast:

Створюємо та додаємо у ноди точку для генерації сфери яка буде генеруватися при виборі іншого існуючого об'єкта зі сцени. Надалі це буде як основа для подальшої генерації частинок на різній поверхні.

Ця сфера виступає як маркер або індикатор місця зіткнення, що вказує на точку ураження променя. Далі, використовуючи цю інформацію, ми можемо взаємодіяти з об'єктами або виконувати певні дії на базі місцезнаходження сфери. Наприклад, це може бути використано для визначення місця попадання кулі під час стрільби в ігровому середовищі або для встановлення точок спрацювання ефектів ударів або взаємодії з об'єктами у віртуальному світі.

10. Використання кількох рейкастів для додавання багатьох сфер на поверхню:

Застосовуємо кілька променів одночасно для виявлення точок зіткнення з поверхнею об'єкта. Це дає змогу для генерації кількох сфер на поверхні. Також дозволяє нам точніше визначити форму та рельєф поверхні об'єкта, а також виявити різні точки зіткнення з цією поверхнею.

Завдяки цьому можна створити не лише одну сферу на поверхні, але й додати багато сфер, що представляють різні точки взаємодії з об'єктом.

Наприклад, якщо поверхня об'єкта має нерівності або деталізовану текстуру, застосування кількох рейкастів дозволить нам більш точно відобразити цю складність та створити більш реалістичне відтворення поверхні. Такий підхід полегшує процес генерації об'єктів або ефектів, які реагують на різні частини поверхні об'єкта, і забезпечує більшу гнучкість у розробці віртуальних середовищ.

11. Рандомізація позиції Raycast:

Тут до генератора додаємо кілька випадкових точок генерації вказавши кількість сфер для генерації на поверхню.

Ми випадковим чином вибираємо точки на поверхні об'єкта для виявлення точок зіткнення. Це дозволяє нам створити випадковий розподіл сфер на поверхні, що призводить до більш природного та реалістичного вигляду снігу чи інших об'єктів, які генеруються. Рандомізація дозволяє уникнути одноманітності

та підвищити візуальну цікавість віртуального середовища, а також додати в нього елемент несподіваності та унікальності.

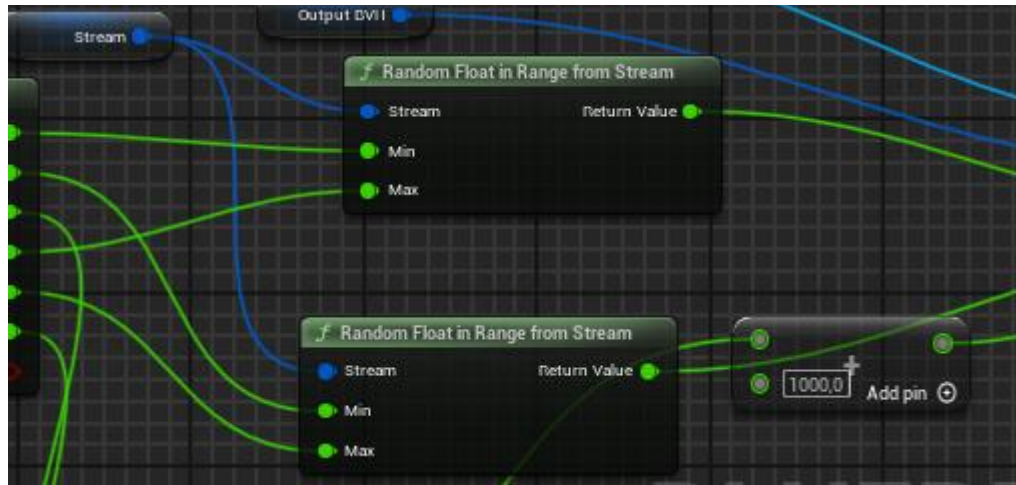


Рис.2.25. Додаємо рандомізацію позиції Raycast

Джерело: розроблено автором

12. Утримання об'єктів у світі:

У попередньому пункті є деякий момент, генерація робиться так що генерують випадкові частинки та поверхня на якій має відбуватися генерація . Щоб уникнути цього ми додаємо до Blueprint Transform Mesh (Рис.2.26) і вказуємо також прив'язку до світу. Це дасть змогу генерувати частинки на різній поверхні за вибором користувача.



Рис.2.26. Логічна зв'язка нодів Transform Mesh

Джерело: розроблено автором

Це дозволяє уникнути ситуацій, коли генеровані об'єкти "загубляються" у просторі або випадково виходять за межі області, що призводить до неправильного відображення або недоліків у візуалізації. За допомогою утримання об'єктів у світі ми забезпечуємо правильне їхнє позиціонування та взаємодію з оточуючим середовищем, що покращує реалістичність.

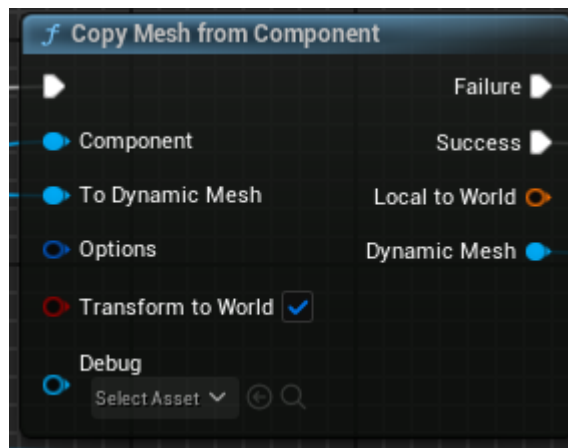


Рис.2.27. Transform to World

Джерело: розроблено автором

13. Додавання снігу на кілька об'єктів:

Для того щоб генерація відбувалась не тільки на одному об'єкті нам потрібно створити масив для вибору об'єктів на сцені.

Після створення масиву для вибору об'єктів на сцені, ми можемо додати більше гнучкості та реалізму у генерації снігу. Це дозволяє розширити можливості нашого генератора снігу, роблячи його більш універсальним та адаптованим до різних сценаріїв. Користувач може вибирати бажані об'єкти, на яких хоче бачити сніг, що дозволяє створювати різноманітні та цікаві віртуальні сцени.

Такий підхід дозволяє забезпечити більшу свободу та контроль над генерацією снігу, підвищуючи рівень іммерсії та задоволення користувачів від взаємодії з віртуальним середовищем.

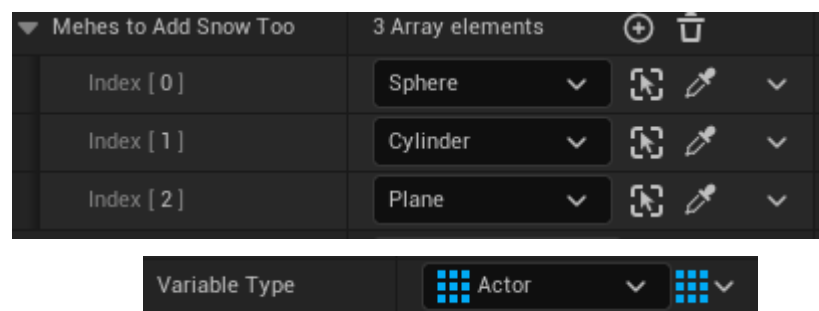


Рис.2.28. Створення масиву вибірки для генерації снігу

Джерело: розроблено автором

14. Compute Meshes:

Використовуємо для відображення снігу віртуальному середовищі. Цей етап відповідає за конвертацію даних про сніг, отриманих під час генерації, у

відповідні меші, які можна відобразити на екрані. В результаті цього процесу ми отримуємо візуально привабливий та реалістичний сніг, який може бути відображений користувачам у віртуальному середовищі.

15. Створюємо параметр для кількості генерованих частинок:

До Blueprint додаємо параметр Number Of Sphere .Цей параметр дозволяє змінювати і регулювати густину снігового покриву, відповідно до потреб і вимог конкретної сцени або ігрового середовища. Це надає розробникам більшу гнучкість і контроль над процесом генерації снігу, а також дозволяє створювати більш реалістичні та деталізовані візуальні ефекти.

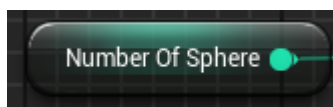


Рис.2.29. Додавання параметра Number of Sphere

Джерело: Unreal Engine 5

16. Додаємо параметр згладжування:

Для того щоб наші частинки виглядали більш реалістичною та не були сфероподібними ми додамо згладжування до нашого генератора. Це дасть змогу вказувати величину згладжування , а також це дасть більш привабливий та красивий вигляд.

Після додавання параметра згладжування до нашого генератора ми отримуємо можливість контролювати ступінь округлення країв частинок снігу. Це дозволяє нам створювати більш природні та реалістичні форми сніжних зграй, оскільки реальний сніг має тенденцію до згладжування та округлення під впливом різних факторів, таких як температура та тиск. Крім того, параметр згладжування дозволяє нам створювати більш привабливі та естетично задовільні візуальні ефекти, що підвищує загальний рівень реалізму та привабливості для віртуального середовища.



Рис.2.30. Додавання параметра згладжування і під'єднування його до Blueprint

Джерело: розроблено автором

17. Додаємо текстури :

Для того щоб сніг не виглядав як примітив ми накладаємо текстуру снігу у параметрах об'єкта. Це дозволяє уникнути враження "примітивності" чи штучності снігового покриву та забезпечує більш природній і естетичний вигляд візуальних ефектів. Такий підхід забезпечує більшу привабливість для користувачів та створює атмосферу реалізму у віртуальному середовищі.

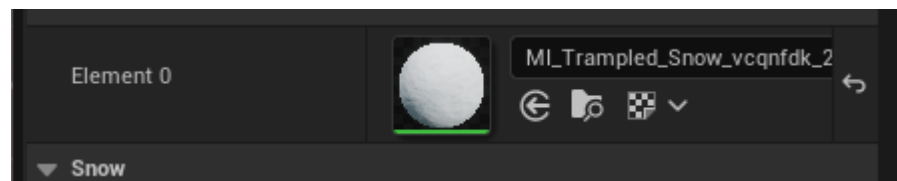


Рис.2.31. Додавання текстури на об'єкт

Джерело: розроблено автором

18. Тестування:

Після завершення розробки першого варіанту генератора снігу, проведено тестування для перевірки його ефективності, надійності та відповідності вимогам проекту. Тестування включало в себе сценарії з різними параметрами та умовами, а також перевірку на відповідність очікуваним результатам. Результати тестування допомогли виявити та виправити можливі проблеми та недоліки, а також підтвердили правильність розробленого рішення.

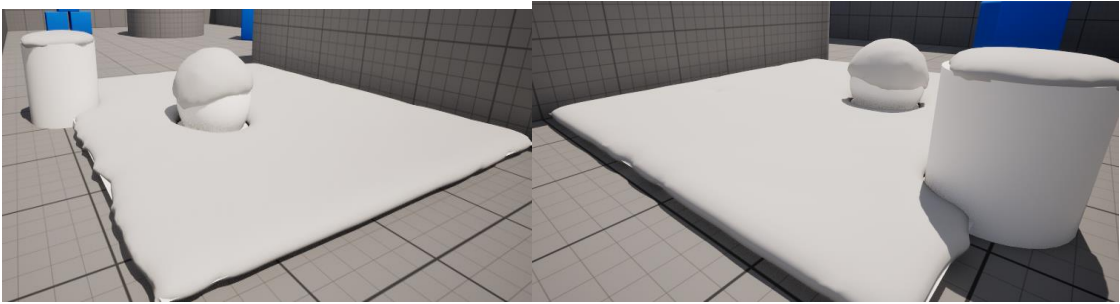


Рис.2.32. Результат тестування генератора снігу.

Джерело: розроблено автором

Після всіх цих етапів розробки в мене вийшла перша тестова версія генератора снігу у середовищі Unreal Engine 5. Я зрозумів і дійшов висновків що це не дуже вдалий варіант , тому що він має кілька недоліків такі як :

- Генерація відбувається тільки тоді коли ми задамо кількість сфер . Це є дуже незручно і при великій кількості сфер генерація займає багато часу.
- Також я мало виділив уваги для текстурування поки ми маємо прототип а не фінальний варіант тому цей момент я упустив.
- Генерація відбувається хаотично, тобто сфери які створюють сніг генеруються випадково і не структуровано що є трохи не правильно це буде впливати на роботу у проекті.

Це основні недоліки які я виділив при розробці першого варіанту генератора. Тому згодом я приступив до розробки другого(фінального) варіанту генератора.

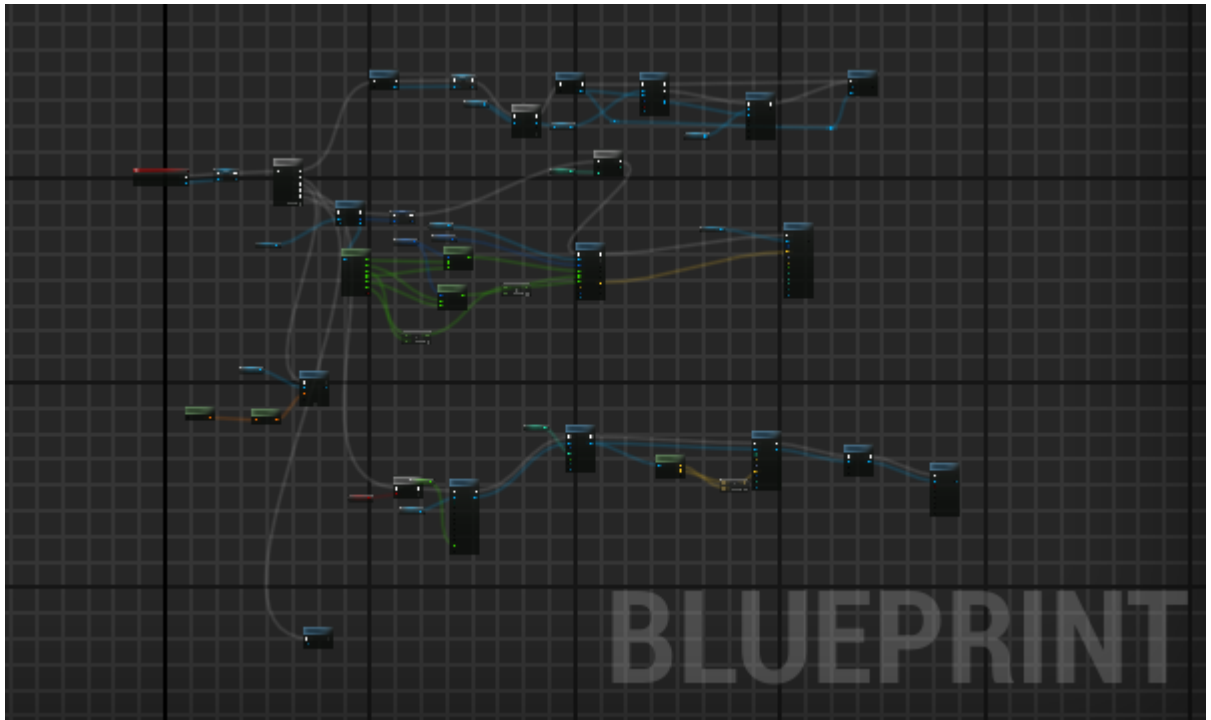


Рис.2.33. Загальний вигляд першого варіанту генератора снігу в Unreal Engine 5

Джерело: розроблено автором

За основу подальших розробок я взяв свій перший варіант . Тому я просто продовжив роботу над виправлення та удосконаленням генератора.

Основні етапи розробки другого варіанту генерації снігу включають в себе:

1. Використання складного тестового ресурсу:

Покращив якість тестування та забезпечив більш точне відтворення реального використання програмного забезпечення. Це включає в себе використання складних тестових сценаріїв, які можуть відтворювати різні умови та ситуації, що допомагає виявити та виправити потенційні проблеми у програмному забезпеченні перед його впровадженням. Такі ресурси дозволяють більш повно та об'єктивно оцінити якість роботи програми в різних умовах та взаємодії з різними факторами.

2. Додавання значка інструмента:

Також додав значок генератора для того щоб не шукати його у списку об'єктів а на сцені просто вибрати його.

Значок інструмента може бути візуально відрізнитися та легко розпізнаватися серед інших об'єктів, що сприяє ефективності та продуктивності роботи



Рис.2.34. Зовнішній вигляд значка інструмента

Джерело: розроблено автором

3. Тестування базового генератора:

В даному етапі я провів ще одне тестування для перевірки чи все працює. Тестування базового генератора дозволило перевірити функціональність основних елементів програми та визначити їхню правильність та ефективність роботи.

Цей етап тестування є важливим для виявлення потенційних проблем та вдосконалення якості програмного забезпечення перед його впровадженням або використанням користувачами.

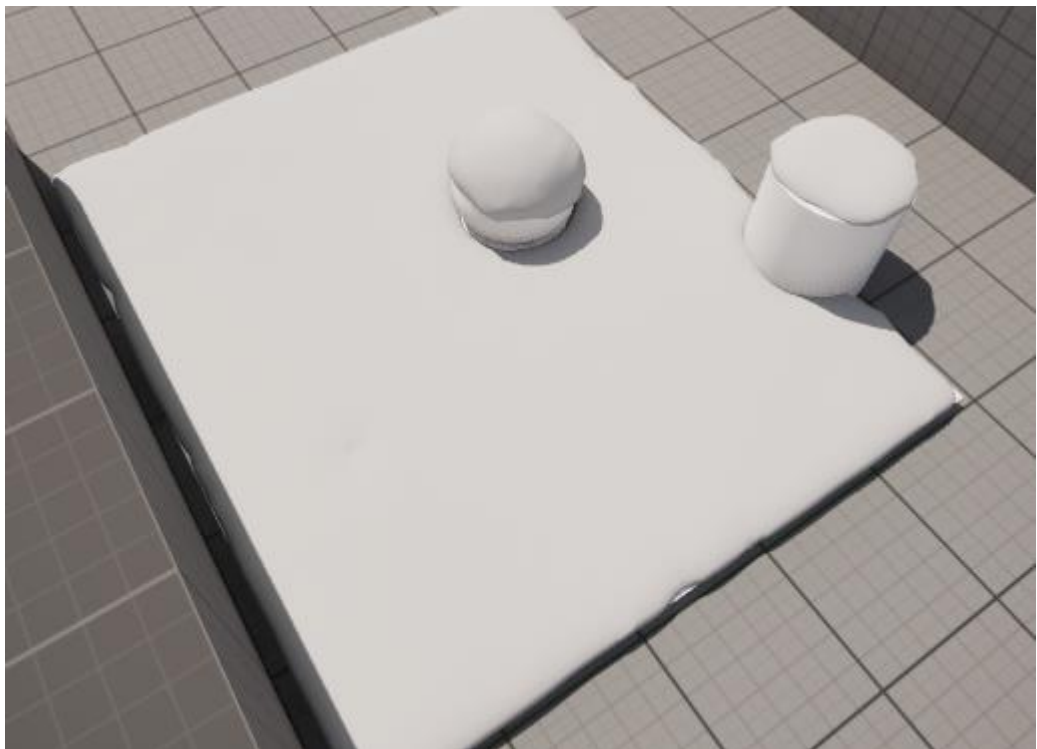


Рис.2.35. Результат тестування після внесених попередньо змін

Джерело: розроблено автором

4. Оптимізація BVH:

Дозволяє досягти кращої продуктивності та ефективності у віртуальних середовищах, забезпечуючи плавну та реалістичну візуалізацію. Це дозволяє краще управляти геометричними об'єктами та їхніми колізіями, що призводить до підвищення продуктивності та ефективності в середовищах реального часу, таких як ігрові двигуни чи симулятори.

Оптимізація BVH дозволяє оптимізувати роботу ігрових сцен, зменшуючи навантаження на процесор та пам'ять і забезпечуючи плавну та реалістичну візуалізацію об'єктів у віртуальному середовищі.

Також ми отримали можливість генерувати кеш який буде зберігатися при генерації.

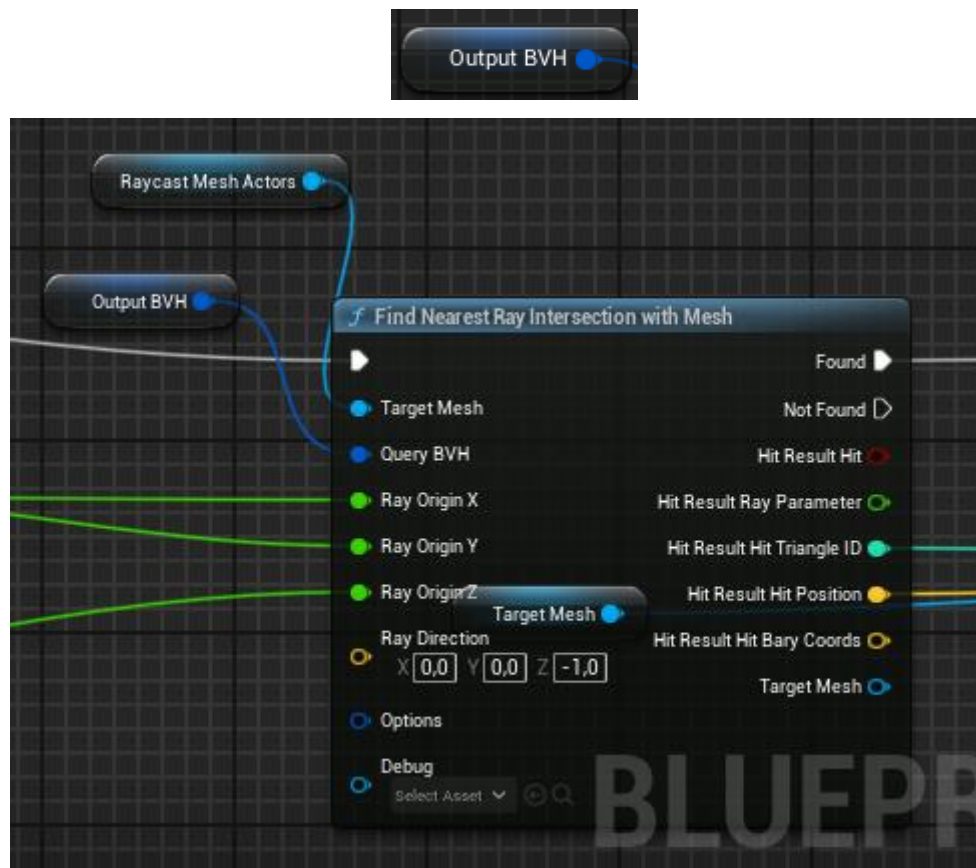


Рис.2.36. Параметр BVH та його підключення до основного Blueprint

Джерело: розроблено автором

5. Налаштування та виявлення змінних:

Налаштування параметрів генератора, що дозволяє створити бажаний вміст або функціонал у віртуальному середовищі.

Цей етап включає в себе налаштування параметрів генератора, які визначають різноманітні аспекти віртуального середовища, такі як геометрія, освітлення, анімація та інші елементи. Налаштування параметрів дозволяє створити бажаний вміст або функціонал у віртуальному середовищі, відповідно до специфікацій проекту та вимог користувачів.

Під час виявлення змінних важливо виявляти та вирішувати проблеми, які можуть виникнути під час розробки або взаємодії з віртуальним середовищем, щоб забезпечити якість та ефективність роботи програмного забезпечення.

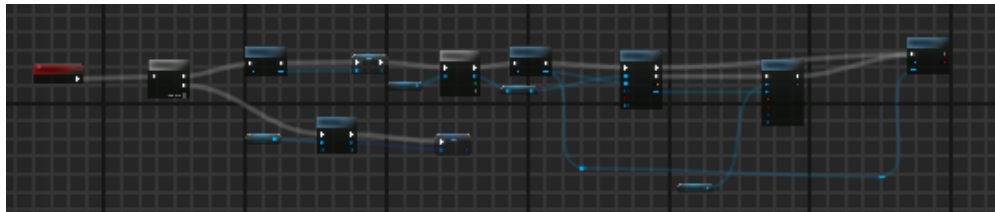


Рис.2.37. Загальний вигляд блоку Generate Mesh Cash

Джерело: розроблено автором

6. Покращення Raycasting:

Тут я розробив метод за яким будуть генеруватися частинки на поверхні. Це просто сітка де сфери генеруються на поверхні впорядковано. Це означає, що буде створено алгоритм, який забезпечує рівномірне та систематичне розміщення частинок на поверхні шляхом використання сітки.

Цей метод дозволяє забезпечити більш однорідний та зручний розподіл частинок, що підвищує реалізм та естетику генерованого снігу або іншого візуального елемента у віртуальному середовищі.

7. Створення сітки для вихідних позицій Raycast:

Створення сітки для вихідних позицій Raycast в Blueprint дозволяє нам генерувати частинки у вигляді систематичної сітки на поверхнях об'єктів у віртуальному середовищі. Це значно полегшує процес генерації, оскільки забезпечує більш точне та прогнозоване розміщення частинок.

Використання такої сітки дозволяє нам з легкістю контролювати розмір, форму та розташування генерованих частинок, що робить процес створення візуальних ефектів більш ефективним та гнучким.

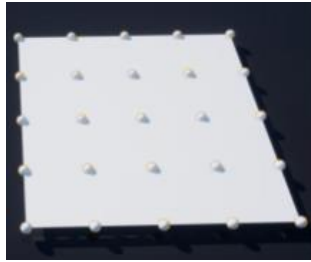


Рис.2.38. Сітка для вихідних позицій Raycast

Джерело: розроблено автором

8. Регулювання снігового покриву:

Додав параметр снігового покриву, який регулює рівень снігового покриву відносно параметрів. Цей параметр дозволяє точно контролювати глибину та інтенсивність снігового покриву відповідно до потреб та умов віртуального середовища. Наприклад, змінюючи значення цього параметра, ми можемо змінити відображення снігу від легкого покриття до густого шару, а також контролювати швидкість накопичення снігу. Такий підхід дозволяє нам створювати більш реалістичні та адаптивні візуальні ефекти, що відтворюють різноманітність умов у віртуальному середовищі.



Рис.2.39. Параметр Snow Coverage та його підключення до основного Blueprint

Джерело: розроблено автором

9. Remesh:

До Blueprint я додав параметр Remesh що дало змогу поєднувати сфери разом. Додавання параметру Remesh у Blueprint дозволило мені об'єднувати сфери разом, утворюючи одне ціле. Цей процес, відомий як Remesh, забезпечує більш однорідну та згладжену текстуру поверхні, яка є більш реалістичною та естетично приємною для спостереження.

Застосування параметру Remesh дозволяє нам створювати більш складні та деталізовані структури зі сфер, що покращує візуальний вигляд та реалістичність снігового покриву віртуального середовища.

10. Контроль товщини снігу:

Додавши кілька параметрів до Blueprint, я можу контролювати товщину снігу на поверхні віртуального середовища. Ці параметри дозволяють регулювати товщину шару снігу залежно від вимог або умов у грі. Змінюючи ці значення, ми можемо створити різноманітні ефекти і змінювати вигляд снігового покриву, що додає більшу гнучкість та реалістичність до віртуального середовища.



Рис.2.40. Параметр Snow Thickness та його підключення до основного Blueprint

Джерело: розроблено автором

11. Налаштування зручності:

Розглянувши всі видимі параметри, необхідні для генерації, я розділив їх на категорії та створив діалогове вікно у Unreal Engine 5. Це полегшує процес генерації, роблячи його більш зручним та зрозумілим. Тепер користувачам не

потрібно шукати окремі параметри, вони групуються за відповідними категоріями, що спрощує їх введення під час генерації снігу.

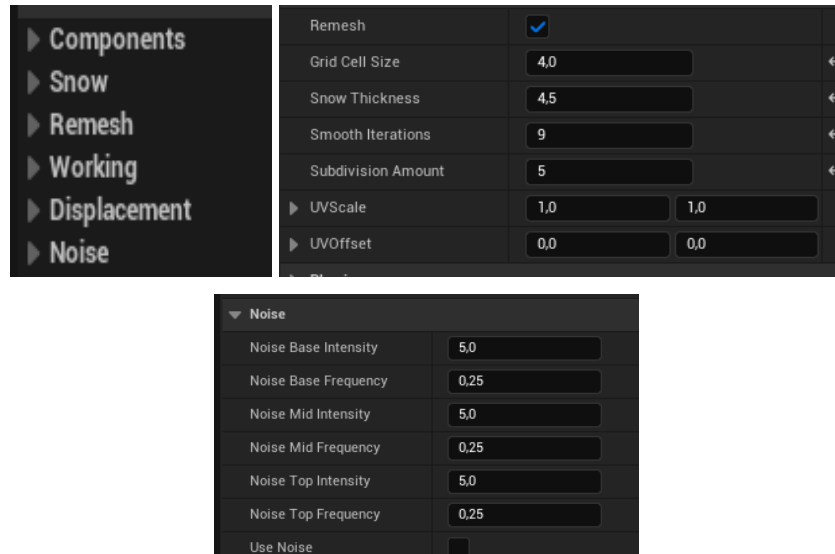


Рис.2.41. Зручний вигляд параметрів генератора снігу

Джерело: розроблено автором

12. Використання LOD для Raycasts:

До загального Blueprint я завів LOD . Це дозволяє оптимізувати продуктивність шляхом використання менш деталізованих моделей для обчислення перетинів. Це зменшує навантаження на систему, зберігаючи високу продуктивність.

13. Деталізація снігу за допомогою текстур та їх зміщення:

В даному пункті я створив та додав параметри для внесення текстур у генератор це додало відчуття випуклості та текстурованості на поверхні, використовуючи спеціальні текстури, які модифікують геометрію, щоб створити рельєфність. Це дозволяє отримати більш природний вигляд снігу з різноманітними переходами між вершинами.

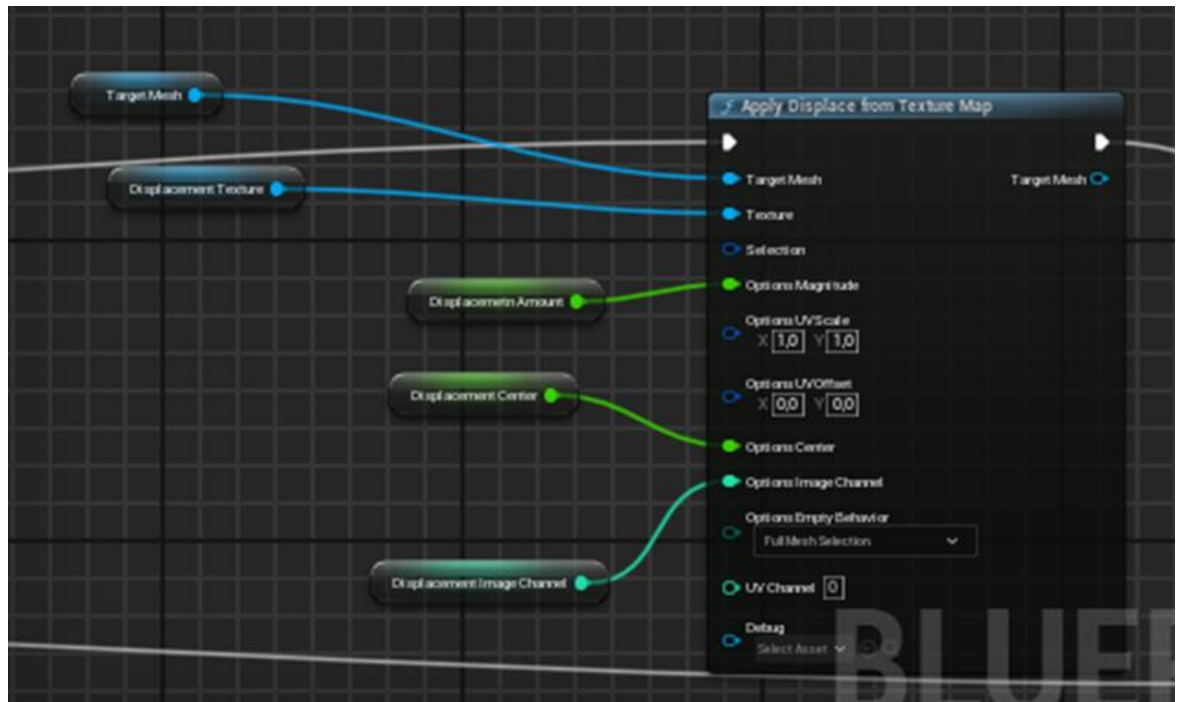


Рис.2.42. Параметри зміщення текстур та їх підключення до основного Blueprint

Джерело: розроблено автором

14. Розбиття для додавання деталей:

Розбиття для додавання деталей дозволяє генерувати частинки більш реалістично. Завдяки цій можливості ми можемо створювати більш деталізовані частинки, що призводить до покращення якості генерації. Оскільки тепер частинки генеруються впорядкованою сіткою, ці зміни вносять ще більше бажаних результатів у вигляді більш реалістичного візуального ефекту.

15. Поліпшення UV:

Генератор створює динамічний mesh який має свою сітку для текстурування. Тому я покращив та удосконалив UV сітку згенерованих елементів, на які текстура буде лягати більше реалістично.

Шляхом покращення та удосконалення UV сітки згенерованих елементів, ми забезпечуємо більш точне та ефективне розташування текстури, що сприяє створенню більш реалістичного візуального враження.

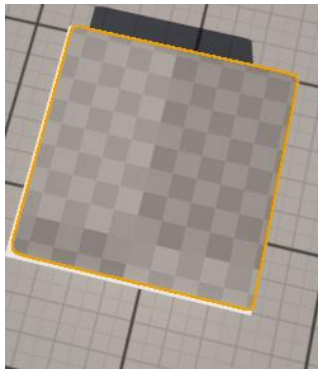


Рис.2.43. Поліпшена UV карта згенерованого об'єкта

Джерело: розроблено автором

16.Покращення нормалей:

Для правильного відображення я виправив помилки з нормаллями. Покращення нормаллей дозволяє досягти більш точного відображення поверхневих деталей та освітлення об'єктів. Шляхом виправлення помилок з нормаллями ми забезпечуємо більш природній та реалістичний вигляд сцен та об'єктів у віртуальному середовищі.

17.Виправлення шумів та покращення візуального вигляду згенерованих частинок:

Виправлення шумів та покращення візуального вигляду згенерованих частинок сприяє створенню більш реалістичного та приємного для спостереження віртуального середовища. Шляхом виправлення помилок і усунення шумів ми досягаємо більшої чіткості та якості зображення, що покращує загальний візуальний ефект та реалізм.

18.Конвертація:

Конвертація снігу в Static Mesh дозволяє зберегти створений сніг як окремий об'єкт у проекті. Це спрощує управління та збереження створеного снігу, оскільки він стає частиною проекту і не зникає при закритті програми.

Крім того, використання Static Mesh дозволяє оптимізувати ресурси системи, оскільки менше навантажується під час відтворення сцени.

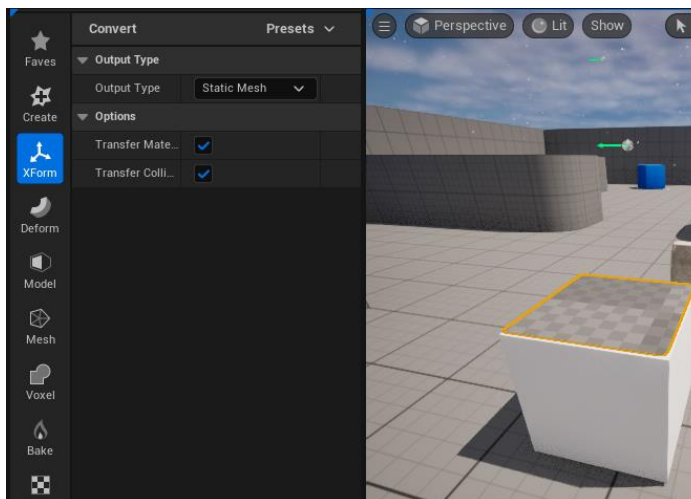


Рис.2.44. Процес конвертації в Static Mesh

Джерело: розроблено автором

19. Тестування:

На даному етапі ми маємо робочий та стабільний генератор снігу який я протестував.

Результати тестування показали, що робочий генератор снігу демонструє високу стабільність та надійність у відтворенні. Проведені випробування дозволили виявити і усунути всі можливі недоліки, забезпечуючи безперебійну роботу системи генерації снігу.

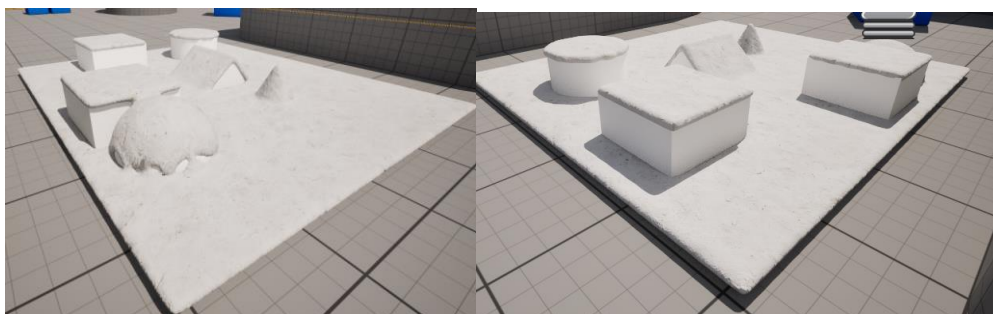


Рис.2.45. Результат тестування генератора снігу на примітивах

Джерело: розроблено автором

Підсумовуючи внесені зміни та вдосконалення до генератора снігу, можемо констатувати значний прогрес у розробці системи. Аналізуючи попередню версію генератора можемо сказати що ми вирішили деякі проблеми та помилки що заважали нормальній роботі генерації. В результаті вдосконалення різних аспектів, таких як точність розміщення снігу, оптимізація продуктивності та поліпшення візуальної якості, ми досягли стабільної та ефективної роботи генератора.

Нові можливості, такі як керування товщиною снігу та поліпшення текстур та нормалей, роблять систему більш гнучкою та реалістичною. Тестування підтвердило працездатність та надійність системи, а конвертація в Static Mesh дозволяє забезпечити постійний доступ до згенерованого снігу.

В цілому, наш генератор снігу тепер працює більш ефективно та надійно, відображаючи вдосконалену функціональність та візуальну якість. Ці всі аспекти додають більш реалістичність та візуально приваблюють своїм зовнішнім виглядом.

2.6.3 Розробка інтерактивності з предметами

Реалізація можливості брати у руки різноманітні предмети у віртуальному середовищі полягає у створенні віртуальних об'єктів, які можна взаємодіяти за допомогою контролера або спеціальних рухомих пристроїв. Ця функція надає користувачам можливість взаємодіяти з об'єктами у віртуальному просторі, так само, як вони зробили б це у реальному світі.

Для реалізації цієї можливості, розробники створюють віртуальні об'єкти з властивостями фізичного тіла, що дозволяє їм реагувати на дії користувача. Наприклад, об'єкт може мати параметри ваги, розміру, форми та матеріалу, які впливають на спосіб, яким користувач може взаємодіяти з ним.

Користувач може брати об'єкти у руки, використовуючи контролер або інші пристрої вводу, і виконувати з ними різноманітні дії, такі як піднімання, пересування, обертання та відкладання. Ця можливість розширює інтерактивність та іммерсивність віртуального досвіду, дозволяючи користувачам більш активно взаємодіяти з віртуальним середовищем.

Загалом розробка інтерактивності з предметами включає в себе два етапи реалізації:

1. Реалізація Grabbing System:

На даному етапі я реалізував систему піднімання предметів для основного персонажа через Blueprint.

- Початковий етап створення системи взаємодії з предметами включав в себе визначення клавіші для взаємодії з об'єктами у віртуальному середовищі. У моєму випадку ця клавіша - F.[26] Це може бути початок для реалізації функціоналу піднімання предметів у віртуальному світі.

Розробка починається з призначення клавіші, що визначає основний засіб для користувача взаємодіяти з об'єктами. Після визначення цієї клавіші може розпочатися програмування логіки, яка дозволяє користувачеві взаємодіяти з об'єктами у віртуальному середовищі, наприклад, піднімати їх, рухати, взаємодіяти з ними тощо.



Рис.2.46. Призначення клавіші на яку буде підніматися предмет

Джерело: розроблено автором

- Далі ми приступили до реалізації зміни позиції предметів на поверхні відносно ігрового світу. Реалізація зміни позиції предметів на поверхні ігрового світу є важливим етапом в розробці ігор. Цей процес може включати в себе використання різних технік, таких як рейкасти, фізичне моделювання або розрахунок колізій.

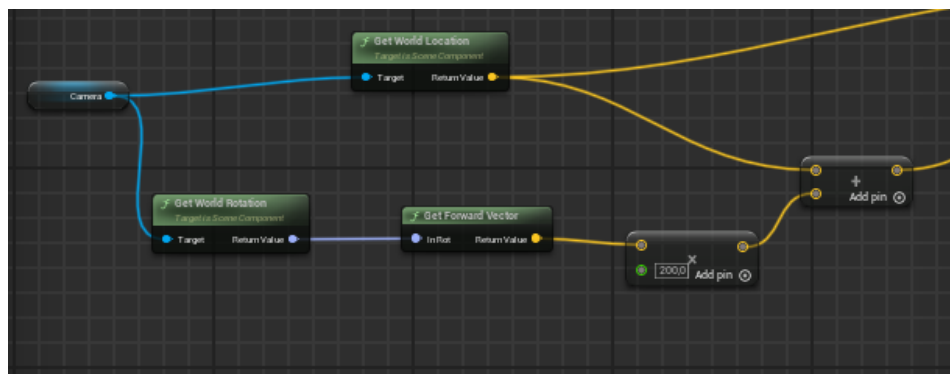


Рис.2.47. Схема реалізації зміни позиції

Джерело: розроблено автором

- Вказав властивість **Physics Handle**. [31]

Вказання властивості Physics Handle є важливим кроком для забезпечення можливості взаємодії з фізичними об'єктами в ігровому середовищі. Physics Handle - це компонент, який дозволяє здійснювати фізичну взаємодію з об'єктами в грі, такими як піднімання, пересування та відкладання.

Вказавши цю властивість, ви дозволяєте вашому герою (або гравцю) взаємодіяти з фізичними об'єктами у грі. Наприклад, це може бути використано для того, щоб зробити можливим піднімання предметів у грі, пересування їх у просторі та відкладання у потрібному місці.

Physics Handle дозволяє вам створити реалістичну та інтерактивну взаємодію з об'єктами у вашій грі, що підвищує її ігрову динаміку та реалізм.

- Для того щоб рушій розумів чи беру я предмет чи ні я створив розгалуження і вказав умову за якої предмет взявся чи ні. Це дало змогу розуміти чи береться цей предмет чи ні.

Створення розгалуження і визначення умови для того, щоб з'ясувати, чи берете ви предмет, є важливим етапом для реалізації взаємодії з об'єктами в грі. Це дозволяє рушій розрізняти, коли гравець взаємодіє з предметом, і виконувати відповідні дії відповідно до цього.

Наприклад, умова може бути пов'язана з позицією гравця відносно предмета: якщо гравець перебуває у певній відстані від предмета і натискає відповідну клавішу, то рушій розуміє, що гравець бере цей предмет.

Крім того, можливою умовою є колізія між гравцем і предметом: якщо об'єкти зіштовхуються або перетинаються, це може вказувати на те, що гравець тримає предмет.

Такі умови дозволяють системі реагувати на дії гравця та виконувати необхідні дії у відповідь на ці дії.

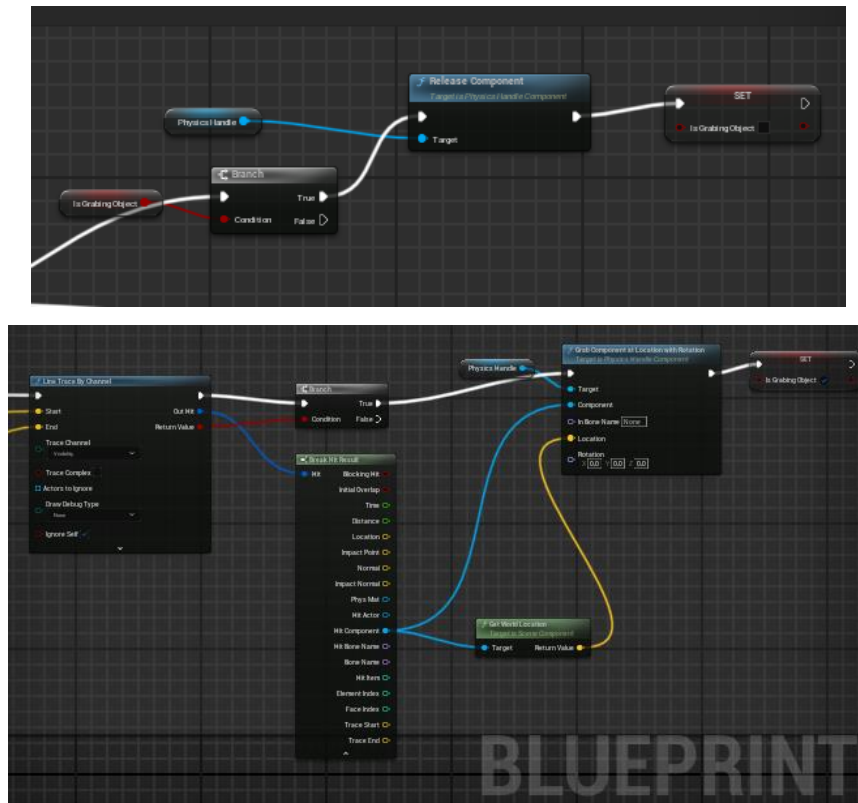


Рис.2.48. Схема розгалуження коли можна брати предмет , а коли ні .

Джерело: розроблено автором

Даний етап полягає у розробці функціональності, яка дозволяє користувачу взаємодіяти з цими об'єктами. Це включає в себе програмування логіки взаємодії, наприклад, обробку введення користувача (наприклад, натискання кнопок на контролері), розрахунок фізики для взаємодії об'єктів та відтворення анімаційних рухів об'єктів під час взаємодії.

2. Реалізація Object System:

- Спершу я задав Event Tick , що слугує визначником того що головний персонаж може взаємодіяти з предметом.[26]

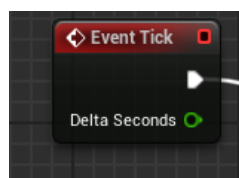


Рис.2.49. Event Tick .

Джерело: розроблено автором

Event Tick - це подія, яка відбувається кожен кадр у грі. Задавши Event Tick, це дозволяє головному персонажу постійно перевіряти своє оточення та взаємодіяти з ним у реальному часі.

Ця подія дозволяє виконувати дії або перевіряти умови кожен кадр, що робить її дуже корисною для реалізації різноманітних інтерактивних ефектів та поведінки героя в ігровому світі.

Задавши Event Tick для головного персонажа, ви можете постійно перевіряти, чи перебуває він у зоні взаємодії з предметом. Це дозволяє вам реалізувати можливість взаємодії з предметами у реальному часі, коли герой наближається до них.

- Далі зробив розгалуження за умови якої предмет взявся або ні.

Це розгалуження є важливим кроком для визначення того, чи головний персонаж взяв предмет, чи ні. Ви використовуєте це розгалуження для того, щоб зробити відповідні дії в залежності від стану взяття предмета.

Якщо умова виконується (тобто предмет взятий), ви можете виконати певні дії, такі як утримання предмета в руках головного персонажа або зміна його позиції.

Якщо умова не виконується (тобто предмет не взятий), то ви можете продовжити перевірку стану взаємодії із зонами предметів або здійснити інші дії, наприклад, перевірити інші можливі інтерактивні об'єкти в ігровому світі.

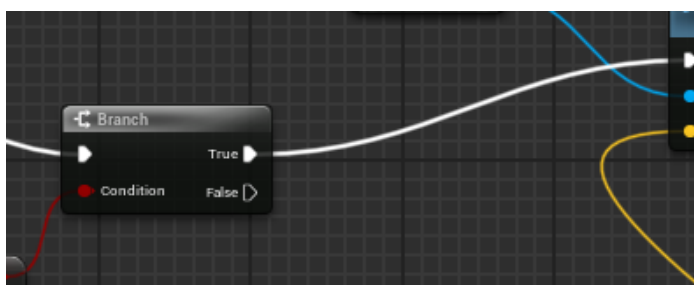


Рис.2.50. розгалуження за умови якої предмет взявся або ні.

Джерело: розроблено автором

- Додав можливість того щоб гра зареєструвала переміщення предмету відносно його стартової позиції. Також вказав властивість для об'єктів за якої можна взаємодіяти з предметом.

Додавання можливості реєстрації переміщення предмету відносно його стартової позиції - це важливий крок для створення реалістичної інтеракції з предметами в ігровому середовищі.

Це означає, що гра може відслідковувати, наскільки і на якій відстані переміщується предмет, коли гравець тримає його в руках. Ця функціональність відкриває можливості для реалізації різноманітних механік гри, таких як кидання предметів, перетягування або розміщення їх у певному місці.

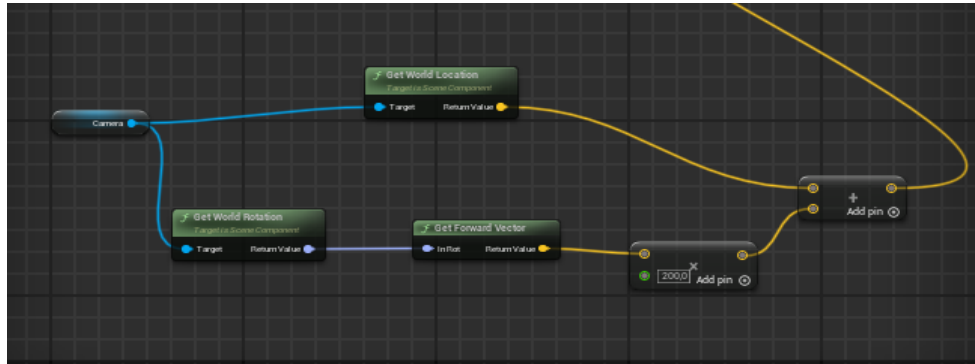


Рис.2.51. Переміщення предмету відносно його стартової позиції.

Джерело: розроблено автором

Це дає гравцям більш можливостей з ігровим світом, а розробникам - можливість створювати цікаві і відчутні механіки гри, які роблять гру більш цікавою і захоплюючою.

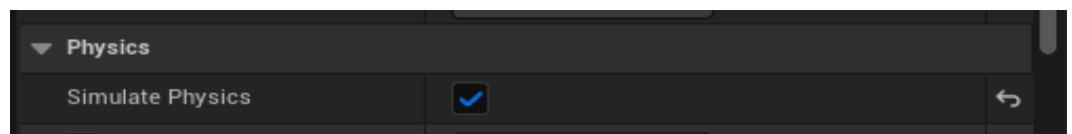


Рис.2.52. Властивість для об'єктів за якої можна взаємодіяти з предметом.

Джерело: розроблено автором

Реалізація Object System - це ключовий етап у розробці будь-якої ігри чи програмного забезпечення, де потрібно взаємодіяти з об'єктами в ігровому світі або в програмі. Ця система визначає, як об'єкти створюються, управляються та взаємодіють один з одним та з гравцем чи користувачем.

Основні компоненти реалізації Object System включають створення об'єктів, визначення їх властивостей та параметрів, управління їхнім життєвим циклом.

лом, обробку взаємодії та зміни стану. Це може включати в себе роботу з класами, інстансами, компонентами, подіями та іншими концепціями програмування об'єктно-орієнтованих мов.

Успішна реалізація Object System дозволяє створювати складні інтерактивні середовища, де об'єкти можуть взаємодіяти між собою та з користувачем у відповідь на їхні дії. Це створює основу для реалізації різноманітних механік гри, штучного інтелекту, фізики та багато іншого, що робить ігровий процес цікавим та захоплюючим для користувачів.

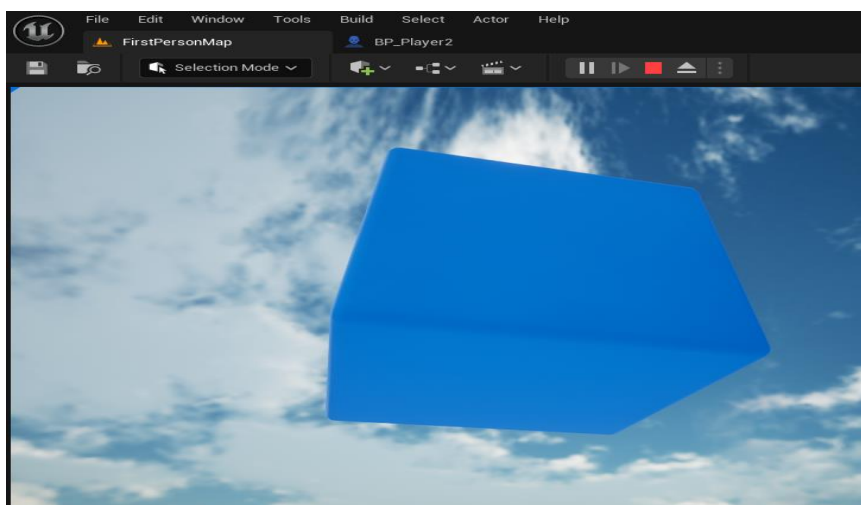


Рис.2.53. Об'єкт піднятий вверх відносно площі

Джерело: розроблено автором

У процесі розробки інтерактивності з предметами, було важливою метою створити систему, яка дозволяє користувачам взаємодіяти з об'єктами в ігровому світі або програмному середовищі. Ця система передбачала можливість брати у руки та взаємодіяти з різноманітними предметами, виконуючи певні дії.

Починаючи з призначення клавіші для підняття предметів та подальшої реалізації функцій обробки цієї дії, було розроблено механізми взаємодії та управління рухом предметів у віртуальному середовищі. Реалізація зміни позиції предметів, визначення їхнього стану під час піднесення та спуску, а також взаємодія з фізичною обстановкою були важливими етапами розробки.

Додатково, створено механізм для визначення, чи був взятий предмет, що дозволяє коректно обробляти дії гравця. Також було використано Physics Handle для керування фізичними властивостями об'єктів.

Використання Event Tick та розгалужень дозволило ефективно контролювати та синхронізувати стан предметів під час взаємодії з ними. Тестування системи дозволяє підтвердити її працездатність та коректну реалізацію функціоналу.

У підсумку, розроблена система інтерактивності з предметами забезпечує гравцям можливість взаємодіяти з об'єктами в ігровому середовищі, що робить гру більш цікавою і захоплюючою.

2.6.4 Створення зимового оточення

Створення зимового оточення - це захоплюючий процес, який дозволяє створити вражаючі зимові ландшафти та сцени у віртуальному середовищі.

Розробивши систему генерації снігу, можна швидко та ефективно створити сніжні покриви, сніжинки, дерева з покриттям снігу та інші зимові елементи, що робить можливим створення атмосферних та реалістичних ігрових світів або сцен у віртуальній реальності.

Після завершення фінальної розробки системи генерації снігу, я інтегрував її у основний проект за допомогою функції Migrate.[29] Це дало можливість імпортувати генератор з усіма його функціями. Провівши кілька маніпуляцій та виконавши кілька генерацій на різних об'єктах території університету, я отримав наступні результати:



Рис.2.54. Готові результати генерації на території університету.

Джерело: розроблено автором

Отримані результати підтвердили, що система генерації снігу успішно інтегрувалися у головний проект та працює стабільно. Сніг ефективно генерувався на різних об'єктах території університету, зберігаючи при цьому високу якість і реалістичний вигляд. Інтеграція системи забезпечує користувачам можливість взаємодії з віртуальним середовищем.



Рис.2.55. Готові результати генерації на території університету.

Джерело: розроблено автором

Створивши зимове оточення за допомогою генератора, я отримав вражаючий результат. За допомогою цього інструменту я зміг швидко та ефективно створити сніжні пейзажі, які додали атмосферності та реалізму моєму проекту. Результат виявився відмінним, і я задоволений зробленою роботою.

2.6.5 Створення звукових ефектів

Під час розробки звукових ефектів я прагнув створити атмосферу, що сприяє більш глибокому зануренню гравця у віртуальне оточення.

Моєю задачею було реалізація звукового оточення та звуків ходьби на деяких поверхнях

Створення звукових ефектів включає в себе кілька етапів реалізації:

1. Для створення відтворення звукового оточення та звуків ходьби на різних поверхнях, я використовував різноманітні аудіофайли та техніки обробки звуку.



Рис.2.56. Звукова доріжка .

Джерело: розроблено автором

2. Потім для матеріалів які застосовуються на об'єктах я задав фізичну властивість для кожного матеріалу з моєї частини роботи.

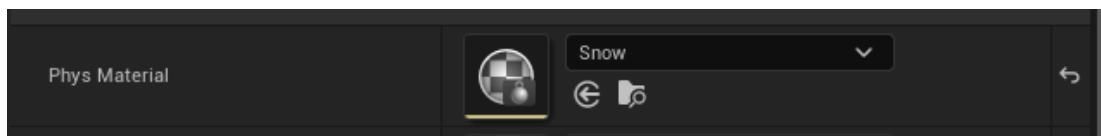


Рис.2.57. Фізична властивість для матеріалів.

Джерело: розроблено автором

3. Завантаживши звукову доріжку я додав в Blueprint до основного персонажа кілька звукових ефектів ходьби на різні поверхні та реалізував схему за якої рушій буде розуміти коли і як генерувати звуки.

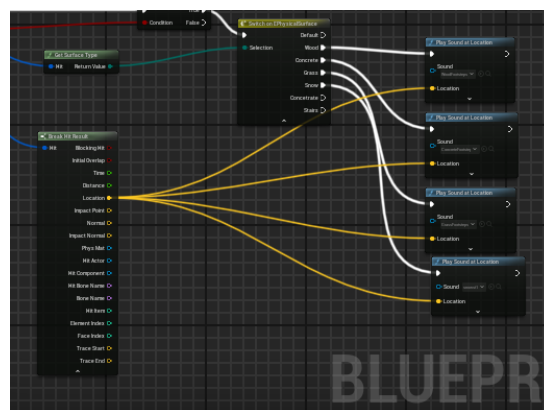


Рис.2.58. Схема генерації звуків на поверхні .

Джерело: розроблено автором

Створюючи звукові ефекти, я досягнув реалістичного звукового оточення та збільшив кількість вражень від гри. Використання різноманітних аудіофайлів

та технік обробки звуку дозволило мені створити більш виразне та живе аудіо-відтворення, що збагачує геймплей та робить гру більш захоплюючою.

Я використовував звукові ефекти для передачі характеристик кожної поверхні, такі як текстура, масштаб і матеріал, щоб створити відчуття реалізму та взаємодії. Також я звертав увагу на динамічні зміни звуків у залежності від швидкості руху та типу підлоги, щоб забезпечити більшу відтворюваність та природність звукових ефектів.

2.6.6 Розробка та впровадження мінімапи

Розробка та впровадження мінімапи в ігрове середовище є важливим кроком для поліпшення навігації гравців та їх загального досвіду гри.

Міні мапа забезпечує гравцю зручний спосіб візуального орієнтування у великих локаціях, вказуючи їм місцезнаходження, напрямок руху та розташування ключових об'єктів чи ворогів. Процес розробки міні мапи включає в себе створення відповідного інтерфейсу, збір та обробку даних про об'єкти на карті, а також програмування логіки відображення та взаємодії з мінімапою. Правильно розроблена та інтегрована міні мапа може значно підвищити ігровий комфорт та зручність для гравців.

Загалом сама мапа реалізована наступним чином:

1. Для створення міні мапи я використав **Widget**, який слугує елементом інтерфейсу для покращення зручності використання самої мапи. Використання **Widget** дозволяє не лише відображати міні-мапу, але і надавати користувачам різноманітні функції, такі як масштабування, переміщення та відображення додаткової інформації про об'єкти на карті. Це створює більш інтерактивний та корисний інтерфейс для гравців, що допомагає їм краще орієнтуватися у грі.[24]

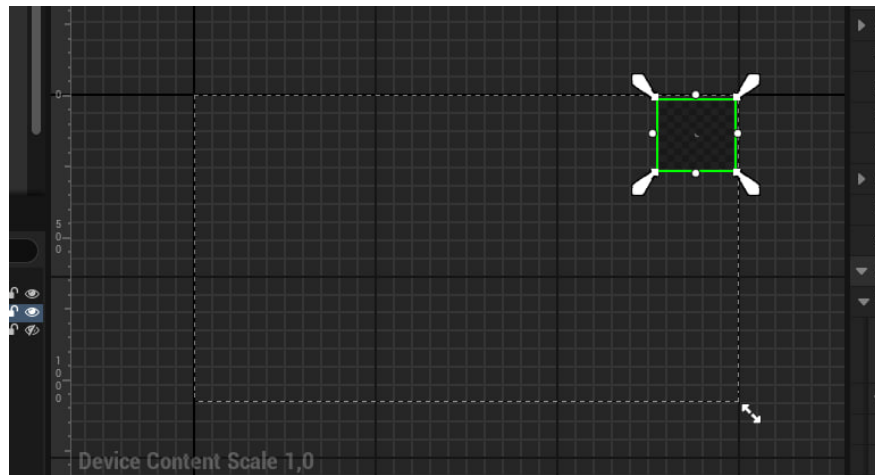


Рис.2.59. Загальний вигляд Widget в Unreal Engine 5 .

Джерело: розроблено автором

2. Для покращення відображення персонажа на міні-мапі, я вирішив додати камеру, яка розміщується над персонажем. Це дозволить гравцям з легкістю визначати положення свого персонажа на карті, надаючи їм більшу контроль і зручність під час навігації по грі. Такий підхід допомагає забезпечити більш повне та зручне використання міні-мапи в грі.

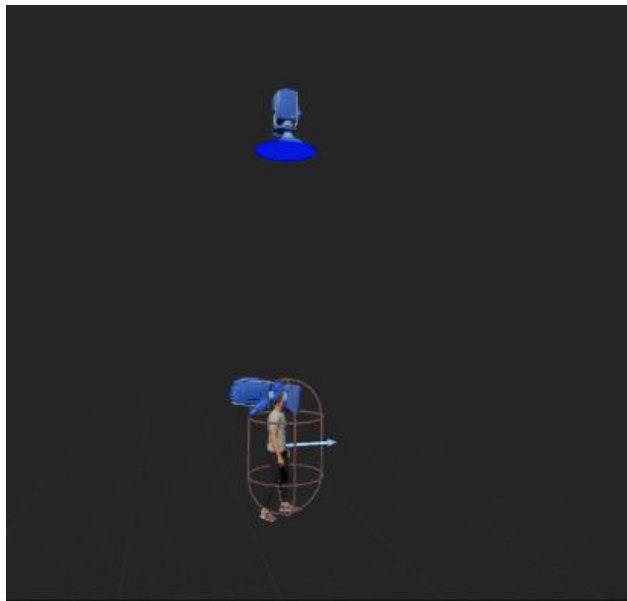


Рис.2.60. Розташування камери над персонажем в Unreal Engine 5 .

Джерело: розроблено автором

3. Після внесення налаштувань щодо розширення міні-мапи та її рендеру, ми забезпечили більшу якість візуалізації на екрані. Ці налаштування дозволили підвищити роздільну здатність міні-мапи та забезпечити більш чітке

відображення деталей, що сприяє покращенню користувацького досвіду та зручності навігації. Тепер гравці можуть більш точно визначати своє місце знаходження на карті та здійснювати швидку орієнтацію у грі.

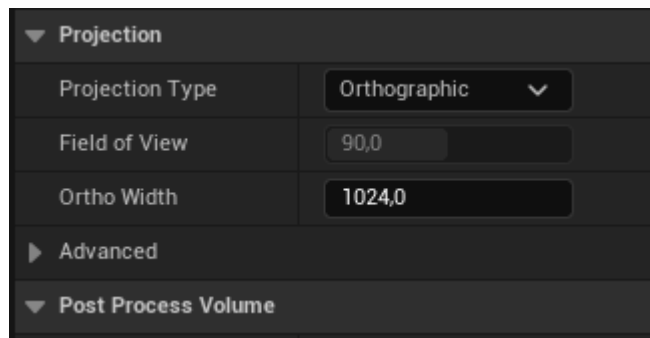


Рис.2.61. Налаштування щодо розширення міні-мапи в Unreal Engine 5 .

Джерело: розроблено автором

4. Далі, щоб покращити візуальну привабливість та інформативність міні-мапи, я створив спеціальний матеріал, який відображається в реальному часі на мапі. Цей матеріал був згенерований в іншому форматі, що дозволило підвищити якість відображення деталей та забезпечити більш точне відображення об'єктів на міні-мапі. Таке вдосконалення сприяє покращенню загального користувацького досвіду та забезпечує більш зручну навігацію гравців у грі.

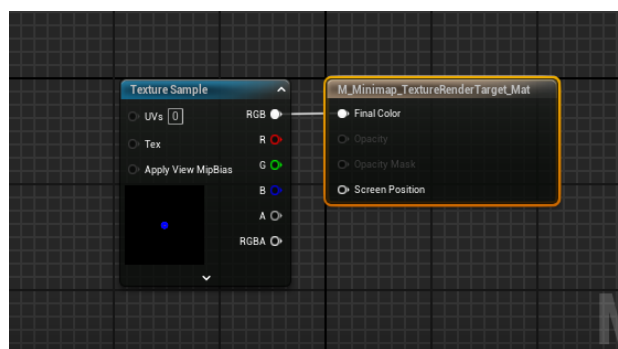


Рис.2.62. Матеріал, який відображається в реальному часі на мапі .

Джерело: розроблено автором

5. Після створення матеріалу ми успішно застосували його на відповідний Widget, що відображається на міні-мапі. Це дозволило нам досягти більшого рівня деталізації та реалістичності відображення об'єктів на міні-мапі, що поліпшує загальний візуальний ефект та забезпечує користувачам більш точну інформацію про їх оточення.[24]

6. Після виконання всіх налаштувань у Blueprint головного персонажа, я додав логіку відображення міні-мапи прямо у грі. Це означає, що тепер гравці можуть бачити міні-мапу безпосередньо на екрані під час гри, що полегшує їм навігацію та розуміння їхнього місця у віртуальному світі.

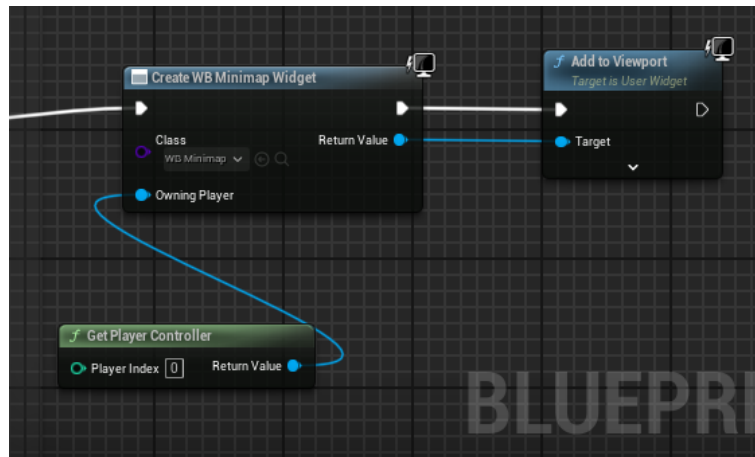


Рис.2.63. Логіка відображення міні-мапи .

Джерело: розроблено автором

7. Для того щоб персонажа було видно на міні-мапі, я додав спрайт точки, тобто зробив точку яка замінює персонажа на міні-мапі.[22]

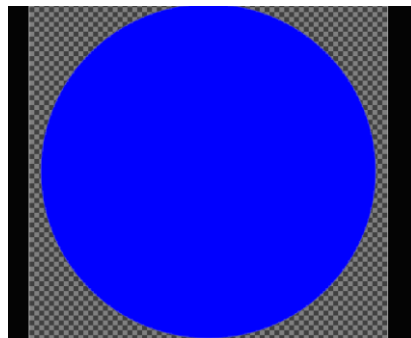


Рис.2.64. Зовнішній вигляд спрайту в Unreal Engine 5 .

Джерело: розроблено автором

Підсумовуючи розробку та впровадження міні-мапи, можна сказати, що цей елемент інтерфейсу став важливим додатком до гри. Шляхом використання Widget та відповідних матеріалів ми створили зручний та функціональний інструмент для гравців. На (Рис.2.65) ми можемо спостерігати відображення міні-мапи в інтерфейсі гри.[24]



Рис.2.65. Зовнішній вигляд міні-мапи Unreal Engine 5 .

Джерело: розроблено автором

Відображення персонажа на міні-мапі та налаштування рендеру дозволили забезпечити точне відображення гравця у віртуальному просторі. Це покращує загальний досвід гри, роблячи навігацію більш зручною та ефективною.

2.6.7 Створення візуальних вказівників

Створення візуальних вказівників - це важлива складова будь-якої гри або віртуальної середовища, що дозволяє гравцям або користувачам легше орієнтуватися у просторі та взаємодіяти з оточенням.

Ці вказівники можуть включати різноманітні об'єкти, такі як вказівники напрямку, покажчики цілей, маркери місць або предметів тощо. Вони допомагають покращити ігровий досвід та забезпечують користувачам зручну навігацію та взаємодію з грою.

Розробка візуальних вказівників була реалізована наступним чином:

1. Перед початком реалізації візуальних вказівників я використовував Adobe Illustrator[6] для створення спрайту вказівника. Цей спрайт, розроблений у відповідному програмному забезпеченні, був потім імпортований в наш проект, де його можна було використовувати для створення візуальних ефектів та вказівників у грі. Такий підхід дозволив забезпечити високу якість та стильний вигляд вказівників.

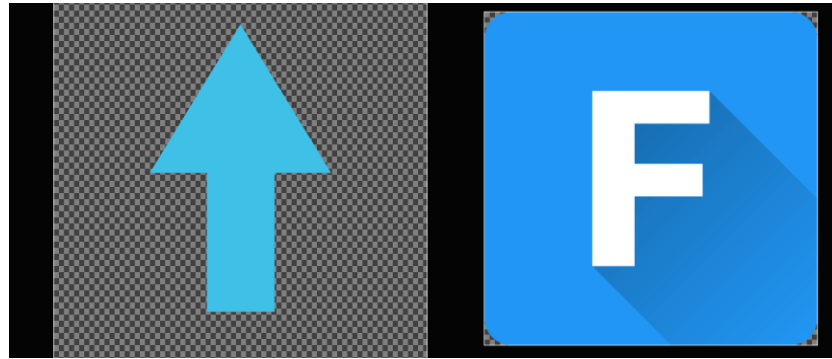


Рис.2.66. Зовнішній вигляд спрайтів Unreal Engine 5 .

Джерело: розроблено автором

- Після створення спрайту вказівника у програмі Adobe Illustrator я приступив до розробки нового Widget у моєму проєкті. Цей Widget був призначений для відображення вказівника у віртуальному середовищі гри. У процесі розробки я вказав конкретне розташування спрайту вказівника та здійснив певні базові налаштування, щоб забезпечити його коректне відображення та функціонування в грі. Це дало можливість створити якісний та ефективний вказівник для користувача, що допомагає взаємодіяти з віртуальним середовищем більш зручно та інтуїтивно.

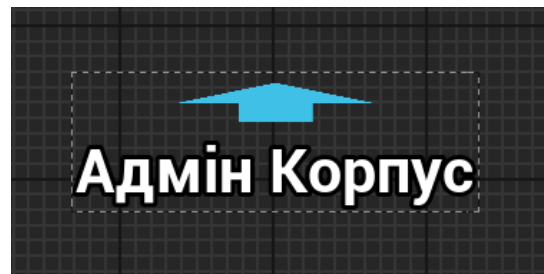


Рис.2.67. Зовнішній вигляд Widget для вказівника Unreal Engine 5 .

Джерело: розроблено автором

- Після створення спрайту вказівника у програмі Adobe Illustrator я приступив до розробки нового Blueprint у моєму проєкті. Цей Blueprint був призначений для відображення та управління вказівником у віртуальному середовищі гри. У процесі розробки я створив логіку, що визначає поведінку вказівника, включаючи його рух та взаємодію з іншими об'єктами у грі. Це дозволило створити функціональний та інтуїтивно зрозумілий вказівник,

який сприяє покращенню взаємодії користувача з віртуальним середовищем.

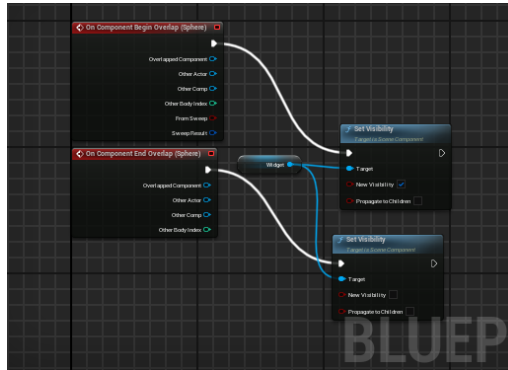


Рис.2.68. Логіку, що визначає поведінку вказівника Unreal Engine 5 .

Джерело: розроблено автором

4. Після створення логіки вказівника у Blueprint, я провів оптимізацію та налаштування параметрів, щоб забезпечити його коректну роботу у віртуальному середовищі гри. Після завершення цього процесу я інтегрував вказівник у головну сцену проекту. Тепер вказівник працює належним чином у віртуальному середовищі, допомагаючи користувачам орієнтуватися та взаємодіяти з об'єктами гри.



Рис.2.69. Зовнішній вигляд вказівника Unreal Engine 5 .

Джерело: розроблено автором

Візуальні вказівники допомагають гравцям орієнтуватися у просторі, знаходити цілі, місця або предмети, а також сприяють загальному покращенню іг-

рового досвіду. У процесі розробки вказівників було здійснено кілька кроків, кожен з яких відігравав важливу роль у створенні якісного та ефективного продукту.

Після завершення всіх етапів розробки візуальних вказівників, було досягнуто створення якісного, функціонального та інтуїтивно зрозумілого продукту.

ВИСНОВКИ

У ході розробки проекту, який був спрямований на створення інтерактивного та віртуального середовища для Національного університету "Острозька академія", ми досягли вагомих успіхів і отримали корисний досвід у використанні віртуальних технологій, для створення моделей та різної інтерактивності.

Перед початком роботи ми з командою розподілили обов'язки між учасниками та розробили алгоритм за яким будемо працювати. . Кожен учасник отримав свою частину завдань згідно з власними навичками та досвідом. Алгоритм містить в собі кроки від розробки концепції та створення першої версії до тестування і вдосконалення продукту.

У програмному середовищі Blender та Unreal Engine 5 були розроблені моделі будівель Адміністративного корпусу Національного університету "Острозька академія", а також різні архітектурні елементи корпусу та прилеглих територій. У ході пошуку інформації були знайдені плани будівель, які згодом стали основою для створення моделей. Плани дали базові розміри та правильність архітектурних аспектів.

Після завершення розробки моделей, ми перейшли до нанесення текстур. У процесі текстурування ми використовували як власні, так і загальнодоступні матеріали. Використання різних видів текстур дозволило нам досягти більшої реалістичності та деталізації віртуального середовища для університету.

Розробивши моделі ми розпочали визначати концепції інтерактивності. Було розроблено генерацію снігу, інтерактивність з предметами, зимове оточення, звукові ефекти, мінімапа та віртуальні вказівники.

Процес розробки враховував усі унікальні аспекти та особливості університету. Розроблені елементи, які були впроваджені у наш проект, покращили реалізм та зробили віртуальний світ більш інтерактивним.

Рішення, які ми прийняли, спрямовані на покращення функціональності віртуального середовища університету. Це дозволило успішно вирішити поставлені завдання та покращити розроблене середовище.

Розробка інтерактивного віртуального середовища для Національного університету "Острозька академія" у рамках Unreal Engine 5 є важливим кроком у впровадженні сучасних технологій в освітній процес. Інтерактивність, як важливий аспект, збільшує реалізм та привабливість віртуального середовища для користувачів, що сприяє покращенню досвіду користувача та підвищенню ефективності навчального процесу.

Враховуючи всі аспекти розробки інтерактивного віртуального середовища в Unreal Engine 5, можна стверджувати, що застосування нових технологій і інноваційних рішень значно покращило наш проект і зробило його більш реалістичним.

Наш проект дозволяє краще зрозуміти технології, які використовуються для створення інтерактивності та віртуалізації університетського середовища. Це відкриває можливості для внесення вагомого внеску у розвиток інноваційних методів взаємодії та візуалізації в навчальному процесі, сприяючи розширенню знань та практичного застосування сучасних технологій в освіті.

Список використаних джерел

1. Брито А. Blender 3D Architecture, Buildings, and Scenery / Allan Brito. – 2015. – Режим доступу: <https://www.packtpub.com/product/blender-3d-architecture-buildings-and-scenery/9781783550181>. (дата звернення: 04.12.2023 р.).
2. Брито А. The Blender 3D Architect: Design and Modeling for a Sustainable Built Environment / Allan Brito. – 2018. – Режим доступу: <https://www.amazon.com/Blender-3D-Architect-Sustainable-Environment/dp/1982052616> (дата звернення: 04.12.2023 р.).
3. Кодрон Р. Blender 3D: Designing Objects [Електронний ресурс] / Romain Caudron. – 2019. – Режим доступу: <https://www.packtpub.com/product/blender-3d-designing-objects/9781789619475>. (дата звернення: 04.12.2023 р.).
4. Blender Kit бібліотека онлайн: <https://www.blenderkit.com/> (дата звернення: 08.12.2023 р.).
5. Adobe Photoshop URL: <http://surl.li/tsatl> (дата звернення: 15.12.2023 р.).
6. Adobe Illustrator URL: <https://www.adobe.com/ua/products/illustrator.html> (дата звернення: 15.12.2023 р.).
7. Official site Unreal Engine URL: <http://surl.li/tsaqw> (дата звернення: 20.12.2023 р.).
8. Working with Meshes: <http://surl.li/tuujs> (дата звернення: 25.12.2023 р.).
9. Dynamic Mesh: <http://surl.li/tuuiz> (дата звернення: 10.01.2024 р.).
10. BRIDGE: <https://quixel.com/bridge> (дата звернення: 15.01.2024 р.).
11. Introduction Blender URL: <http://surl.li/hhlbz> (дата звернення: 25.01.2024 р.).
12. BoolTool: https://docs.blender.org/manual/uk/dev/addons/object/bool_tools.html (дата звернення: 08.02.2024 р.).
13. Download blender: <https://www.blender.org/> (дата звернення: 01.02.2024 р.).
14. Blender render Cycles URL: <http://surl.li/hvgym> (дата звернення: 20.02.2024 р.).
15. Blender render Eevee URL: <http://surl.li/tuuqi> (дата звернення: 23.02.2024 р.).
16. Geometry script documentation in Unreal Engine 5 : <http://surl.li/tuusi> (дата звернення: 04.03.2024 р.).
17. Python API in Blender URL: <http://surl.li/tuusx> (дата звернення: 10.03.2024 р.).

- 18.Geometry script in Blender:<https://carson-katri.github.io/geometry-script/> (дата звернення: 10.03.2024 р.).
- 19.Офіційний сайт НаУОА URL: www.oa.edu.ua (дата звернення: 01.03.2024 р.).
- 20.Introduction to Jira URL: <http://surl.li/taajd> (дата звернення: 01.12.2023 р.).
- 21.Events in Unreal Engine 5 URL: <http://surl.li/tsamh> (дата звернення: 15.03.2024 р.).
- 22.Sprite in Unreal Engine 5:<http://surl.li/tuuwt> (дата звернення: 25.03.2024 р.).
- 23.UA5.0 Geometry Script - Using a BVH to Sample a Mesh Surface with Ray Intersections:<http://surl.li/tuuyq> (дата звернення: 25.03.2024 р.).
- 24.Using Widgets in Unreal Engine 5 URL: <http://surl.li/tsaqq> (дата звернення: 25.03.2024 р.).
- 25.Specialized Blueprint Node URL: <http://surl.li/tsaqq> (дата звернення: 25.03.2024 р.).
- 26.EventHit URL: <http://surl.li/tsany> (дата звернення: 25.03.2024 р.).
- 27.Creating your backlog URL: <http://surl.li/tsaor> (дата звернення: 25.03.2024 р.).
- 28.Physical Material User Guide URL: <http://surl.li/tsaqj> (дата звернення: 02.04.2024 р.).
- 29.UA5: Modeling Mode - Dynamic/Static Mesh Problems and Solutions (Convert, Merge, Duplicate, Split, Export):<http://surl.li/tuvgl> (дата звернення: 02.04.2024 р.).
- 30.Nanite Virtualized Geometry:<http://surl.li/tuvso> (дата звернення: 05.02.2024 р.).
- 31.Introduction to Blueprints:<http://surl.li/tuvtd> (дата звернення: 05.02.2024 р.).

ДОДАТКИ

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ

проекту

Створення інтерактивного віртуального середовища НаУ “Острозька академія”

Метою цього проекту є створення деталізованої і достатньо точної віртуальної інтерактивної копії території корпусів Національного університету “Острозька академія”, яка може бути використана в розробці ігор, віртуальній реальності та WEB-інтерфейсах. Фінальний проект повинен відображати реальний вигляд території, включаючи всі будівлі, споруди, відкриті простори та топографію, а також наповнення інтерактивними елементами.

Виконання цього проекту включає наступні завдання:

- **Збір даних:** Зібрати всю необхідну інформацію та дані, необхідні для процесу 3D-моделювання. Це можуть бути креслення, плани поверхів, плани висот, фотографії, відео та інші відповідні документи. Важливо зібрати якомога більше інформації, щоб забезпечити точність кінцевої 3D-моделі.

- **Програмне забезпечення:** Виберіть відповідне програмне забезпечення для 3D-моделювання, яке може працювати з архітектурними моделями і має необхідні функції для створення точного представлення території університету. Воно повинно дозволяти створювати високоякісні текстури, матеріали та світлові ефекти.

- **Моделювання будівель:** Змоделюйте всі будівлі на території кампусу, використовуючи точні розміри, матеріали та текстури. Будівлі повинні бути змоде-

льовані таким чином, щоб відображати їхній реальний вигляд, включаючи архітектурні деталі, такі як вікна, двері та дахи. 3D-модель повинна також включати внутрішні приміщення, такі як класи, лабораторії та інші. Внутрішні приміщення повинні бути змодельовані з відповідними меблями та обладнанням.

- **Ландшафт і рельєф:** Змодельуйте всі відкриті простори, ландшафти і рельєф в межах кампусу, щоб відобразити топографію місцевості. Це включає в себе моделювання рослинності, такої як дерева, кущі та квіти, а також твердого покриття, такого як доріжки, тротуари та дороги.

- **Освітлення:** Додайте освітлення до 3D-моделі, щоб імітувати умови освітлення в різні пори дня. Це включає моделювання природного освітлення, такого як сонячне світло і тіні, а також штучного освітлення, такого як вуличні ліхтарі, освітлення будівель і внутрішнє освітлення. Освітлення повинно бути налаштоване так, щоб створити реалістичну і візуально привабливу 3D-модель.

- **Текстури та матеріали:** Використовуйте відповідні текстури і матеріали для точного відображення поверхонь будівель і ландшафтів. Це включає в себе моделювання матеріалів, таких як цегла, бетон, скло і метал, а також застосування відповідних текстур, таких як фарба, штукатурка і підлога. Текстури та матеріали повинні бути підібрані таким чином, щоб відображати реальний вигляд університету.

- **Функціональність:** Пророблена вами робота повинна бути реалізована в Unreal Engine з метою подальшого використання з метою створення інтерактивного середовища. Окрім переміщення по створеному вами простору, повинні бути реалізовані інтерактивні елементи про які буде вказано далі.

- **Результати:** Надайте фінальний проект в форматі чергоїст, разом з усіма пов'язаними файлами та текстурами.

- **Забезпечення якості:** Перевірте створене середовище на точність і зручність використання, а також внесіть будь-які необхідні зміни або корективи.

Сюди входить перевірка на наявність помилок моделювання, таких як перекриття полігонів, інвертовані нормалі та відсутні текстури. Також важливо протестувати середовище на зручність використання, наприклад, навігацію по моделі в реальному часі та рендеринг високоякісних зображень і анімації.

Терміни реалізації цього проекту обмежені терміном виробничої практики, тому складають до 4 березня 2024 року.

Роботу над проектом можна розбити на наступні етапи:

- Етап збору даних: На цьому етапі команда збирає всі необхідні дані, необхідні для процесу 3D-моделювання. Це можуть бути креслення, плани поверхів, плани висот, фотографії, відео та інші відповідні документи. Команда повинна буде впорядкувати дані та забезпечити їх точність і актуальність. Також може знадобитися виїзд на об'єкт для збору додаткової інформації та фото- і відеозйомки кампусу.

- Етап вибору програмного забезпечення для 3D-моделювання: На цьому етапі команда обирає відповідне програмне забезпечення для 3D-моделювання, яке може працювати зі складними архітектурними моделями і має необхідні функції для створення точного представлення університету. Команда повинна буде оцінити різні варіанти програмного забезпечення і вибрати той, який найкраще відповідає вимогам проекту та бюджету.

- Етап моделювання: На цьому етапі команда починає моделювати всі будівлі на території кампусу. Команда повинна буде змоделювати будівлі, використовуючи точні розміри, матеріали та текстури. Будівлі повинні бути змодельовані таким чином, щоб відображати їхній реальний вигляд, включаючи архітектурні деталі, такі як вікна, двері та дахи. 3D модель також повинна включати внутрішні приміщення, такі як класи, лабораторії та офіси. Внутрішні приміщення повинні бути змодельовані з відповідними меблями, обладнанням.

- **Етап ландшафтного дизайну та моделювання місцевості:** На цьому етапі команда моделює всі відкриті простори, ландшафти і рельєф в межах університету, щоб точно відобразити топографію місцевості. Сюди входить моделювання рослинності, такої як дерева, кущі та квіти, а також благоустрою, наприклад, доріжок, тротуарів та доріг.

- **Етап освітлення:** На цьому етапі команда додає освітлення до 3D-моделі, щоб імітувати умови освітлення в різні пори дня і року. Сюди входить моделювання природного освітлення, такого як сонячне світло і тіні, а також штучного освітлення, такого як вуличні ліхтарі, освітлення будівель і внутрішнє освітлення. Освітлення повинно бути налаштоване так, щоб створити реалістичну і візуально привабливу 3D модель.

- **Етап роботи з текстурами та матеріалами:** На цьому етапі команда використовує відповідні текстури та матеріали для точного відтворення поверхонь будівель і ландшафтів. Це включає моделювання матеріалів, таких як цегла, бетон, скло та метал, а також застосування відповідних текстур, таких як фарба, штукатурка та підлога. Текстури і матеріали повинні бути підібрані так, щоб точно відображати реальний вигляд кампусу.

- **Етап перенесення моделі в Unreal Engine:** На цьому етапі команда імпортує створені моделі в Unreal Engine, за потреба переробляє проблемні моделі використовуючи функціонал моделювання UE, налаштовує матеріали, освітлення, налаштовує колізії, додає персонажа.

- **Етап створення інтерактивних елементів:** На цьому етапі команда реалізовує всі необхідні інтерактивні елементи відповідно до вимог та стандартів.

- **Етап забезпечення якості:** На цьому етапі команда тестує середовище на точність і зручність використання, а також вносить будь-які необхідні зміни або корективи. Сюди входить перевірка на наявність помилок моделювання, таких

як перекриття полігонів, інвертовані нормалі та відсутні текстури. Також важливо протестувати модель на зручність використання, наприклад, навігацію по моделі в реальному часі та рендеринг високоякісних зображень і анімації.

Кожен виконавець проекту має свою зону відповідальності:

● Новий корпус

1. Кошубінський Павло(інтер'єр першого та другого поверху):

- звукове оточення
- реалізація інтерактивного комп'ютера з можливістю виходу в інтернет через нього та анімацією відсунення стільця;
- відкриття-закриття дверей і вікон;
- увімкнення/вимкнення світла;
- реалізація механіки ліфта;
- камери безпеки та можливість їх перегляду на окремому моніторі;
- реалізація механіки бігу для основного персонажа;

2. Синиця Дмитро (інтер'єр третього та четвертого поверху):

- реалізація чоловічого та жіночого ігрових персонажів з анімаціями;
- реалізація інтерактивного меню з можливістю вибору персонажа.

3. Лаптев Богдан(фасад адмінбудівлі, частина проці перед корпусом, територія за корпусом):

- генератор снігового покриву;
- створення зимового оточення генератором снігового покриву;
- інтерактивне піднімання предметів;
- міні мапа;
- віртуальні вказівники;
- звуки ходьби.

4. Белінський Олександр (фасад корпусу, частина проці перед корпусом):

- годинник на башні академії котрий показує реальний час;

- шлагбаум який відкривається при натисканні клавіші;
- реалізація динамічних погодних ефектів (дощ, гроза, чисте небо, хмарне небо);
- звукове оточення.

5. Мосійчук Василь (фасад академічного дому та визначених будівель біля нового корпусу):

- генератор туманного покриву території;
- реалізація полярного сяйва.

● **Основний корпус**

6. Даніель Демчук (фасад бібліотеки, бібліотечний парк):

- звуки оточення;
- реалізація транспорту у вигляді самокату чи іншого схожого, але не автомобілів;
- можливість сісти на лавку;

7. Ілля Гупалюк (інтер'єр першого поверху):

- реалізація фоторежиму з тим функціоналом який реально реалізувати в межах визначеного часу.

8. Максим Семенюк (фасад корпусу, паркова зона перед корпусом):

- зміна циклу зміни доби;
- нічне освітлення парку;
- анімація сидіння на лавці в парку;
- звуки ходьби по різних поверхнях.

9. Пучко Олександр (стадіон, тенісні корти, фасад господарської частини):

- гра в теніс по типу пінг-понг;
- взаємодія з м'ячем біля футбольних воріт;
- підсвітка/приціл на об'єкт з яким можна взаємодіяти;
- піднімання об'єкта;

- звуки ходьби;
- нічне освітлення стадіону.

10. Черевко Вікторія (інтер'єр академічної бібліотеки):

- реалізація правильного куполу бібліотеки;
- оптимізація колізій;
- реалізація базового функціоналу телефону з можливістю набору номера.