

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

**КВАЛІФІКАЦІЙНА РОБОТА**  
на здобуття освітнього ступеня бакалавра

на тему: «Створення інтерактивного віртуального середовища НаУ ОА  
(екстер'єр старого корпусу та паркової зони)»

**Виконав:** студент 4 курсу, групи КН-41  
першого (бакалаврського) рівня вищої освіти  
спеціальності 122 Комп'ютерні науки  
освітньо-професійної програми «Комп'ютерні науки»  
*Семенюк Максим Олегович*

**Керівник:** викладач  
*Місай Володимир Віталійович*

**Рецензент:** кандидат технічних наук, доцент, доцент  
кафедри прикладної математики та кібербезпеки  
Донецького національного університету імені Василя  
Стуса  
*Загоруйко Любов Василівна*

**РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ**

Завідувач кафедри економіко-математичного моделювання та інформаційних  
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**

Семенюка Максима Олеговича

1. *Тема роботи:* Створення інтерактивного віртуального середовища НаУОА (інтер'єр першого та другого поверхів нового корпусу).

*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 проєкту використовувались: Blender(для моделювання усіх меблів та незначних об'єктів), Unreal Engine 5 (для створення усіх конструктивних елементів та розробка усіх механік інтерактивності), Adobe Photoshop, Adobe Illustrator.

4. *Перелік завдань, які належить виконати:* створення декорацій, взведення усіх конструктивних елементів, розробка фасаду та зведення стін старого корпусу, реалізація звуків, додавання анімацій сидіння, реалізація зміни дня/ночі.

5. *Перелік графічного матеріалу:* рисунки, таблиці.

## 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 роботи/проєкту	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/проєкту	до 31.05.2024 р.	
9	Здача кваліфікаційної роботи/проєкту на кафедрі	31.05.2024 р.	

Студент: \_\_\_\_\_  
( підпис )**Максим СЕМЕНЮК**

Керівник кваліфікаційної роботи: \_\_\_\_\_

**Володимир МІСАЙ**

**АНОТАЦІЯ**  
**кваліфікаційної роботи**  
**на здобуття освітнього ступеня бакалавра**

**Тема:** Створення інтерактивного віртуального середовища НаУОА (екстер'єр старого корпусу та прилеглої території)

**Автор:** Семенюк Максим Олегович

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

**Захищена «.....»..... 20\_\_ року.**

**Пояснювальна записка до кваліфікаційної роботи:** 55 с., 36 рис., 1 додаток, 34 джерел.

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

**Короткий зміст праці:**

Завданням кваліфікаційної роботи/проекту, було створення екстер'єру старого корпусу та прилеглої території. Це інтерактивне середовище може служити засобом представлення загального враження для вступників, які мають можливість віртуально оглянути старий корпус. Використання Blender дозволило створити дрібні деталі для подальшої реалізації, а Unreal Engine 5 забезпечив високоякісний рендеринг та інтерактивність у цьому середовищі. Даний проєкт відкриває нові можливості використання технологій у навчальному процесі та підкреслює потенціал віртуальної реальності для просування університетів та полегшення сприйняття майбутніми абітурієнтами.

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

**Summary of the work:** The task of the qualifying work/project was to create the exterior of the old building and the surrounding area. This interactive environment can serve as a means of presenting an overall impression to the entrants who have the opportunity to virtually tour the old building. Using Blender made it possible to create small details for further implementation, and Unreal Engine 5 provided high-quality rendering and interactivity in this environment. This project opens up new opportunities for the use of technology in the educational process and emphasizes the potential of virtual reality for promoting universities and facilitating acceptance by future applicants.

---

## ЗМІСТ

<b>ВСТУП.....</b>	<b>6</b>
<b>РОЗДІЛ 1. Теоретична частина Створення віртуального середовища НаУ ОА(екстер'єр старого корпусу та паркової зони) та програмного забезпечення Blender та ігрового рушія Unreal Engine 5.....</b>	<b>8</b>
1.1. Постановка проблеми створення екстер'єру старого корпусу та паркової зони, опис базових понять про Blender.....	8
1.2 Аналіз аналогічних розробок.....	13
1.3. Інтерактивна частина розробки для взаємодії з різними об'єктами.....	13
1.4 Використання системи управління проектами Atlassian JIRA.....	16
<b>РОЗДІЛ 2..... Прикладна розробка екстер'єру старого корпусу та паркової зони в програмному забезпеченні Blender.....</b>	<b>20</b>
2.1. Опис технологічного стеку.....	20
2.2. Збір основної інформації для виконання завдання.....	21
2.3. Створення 3Д моделей в Blender.....	22
2.3.2 Деталізація екстер'єру.....	24
2.3.3 Деталізація об'єктів та шейдерів.....	25
2.3.4 Імпортування елементів та їх розстановка відповідно плану.....	26
2.4. Інтеграція 3Д моделей в Unreal Engine.....	29
2.4.2. Виведення усіх конструктивних елементів в Unreal Engine 5.....	31
2.4.3. Деталізація екстер'єру старого корпусу та паркової зони в Unreal Engine 5.....	33
2.5. Інтерактивна частина створення НаУ ОА (екстер'єр старого корпусу та паркової зони).....	36
<b>ДОДАТКИ.....</b>	<b>54</b>

## ВСТУП

Ігрова індустрія у сучасному світі є однією з найбільш швидкозростаючих та впливових сфер розваг. З роками вона не лише продовжує збільшувати свій обсяг, але й стає все більш різноманітною та доступною для широкої аудиторії. Розвиток нових технологій, зростання популярності онлайн-ігор та поширення мобільних геймінгових платформ є лише деякими з факторів, що визначають сучасний ландшафт ігрової індустрії.

Unreal Engine 5 відзначається технологіями та інструментами, які підвищують якість геймплею та візуальні можливості. Зокрема, впровадження Lumen Global Illumination та Nanite Geometry дозволяє розробникам створювати дивовижно реалістичні та деталізовані ігрові світи без значних обмежень щодо оптимізації. Ці технології роблять Unreal Engine 5 потужним інструментом для розробки ігор майбутнього, які надихають гравців своєю красою та іммерсивністю.

Створення віртуального середовища старого корпусу Острозької Академії та прилеглої до нього частини парку в ігровому рушії Unreal Engine 5 - це захоплюючий та водночас відповідальний процес, спрямований на те, щоб передати в цифровому форматі всю красу та атмосферу цього місця. Ми використовували точні матеріали та фотографії для відтворення кожної деталі корпусу та парку з найвищою точністю. Додавання звукових ефектів, таких як шепіт вітру чи спів птахів, а також можливість фізично взаємодіяти з оточенням, наприклад, сидіти на лавках або відкривати двері, додасть реалізму та іммерсивності до віртуального досвіду. Використання механік Unreal Engine 5 дозволить створити надзвичайно реалістичне віртуальне середовище, яке повністю передасть естетику та архітектурну велич.

Мета кваліфікаційної роботи – це створення інтерактивного віртуального середовища старого корпусу Національного університету «Острозька академія» (екстер'єр старого корпусу та паркової зони) та основні завдання на практику:

1. Нічне освітлення;

2. Звуки;
3. Зміна дня і ночі;
4. Анімації взаємодії з предметами.
5. Статуї

Об'єктом дослідження є розробка інтерактивного віртуального середовища старого корпусу Національного університету «Острозька академія» (екстер'єр старого корпусу та паркової зони) в середовищі Unreal Engine 5.

Предметом дослідження є створення високоякісної та інтерактивної віртуальної моделі екстер'єру старого корпусу та паркової зони Національного університету Острозька академія, яка буде використовуватися для навчальних, презентаційних та туристичних цілей.

Мета дослідження:

Аналіз та розробка технічних і дизайнерських рішень для створення віртуальної моделі екстер'єру старого корпусу та паркової зони Національного університету Острозька академія.

Основні завдання:

Технічні аспекти:

- Визначення вимог до програмного та апаратного забезпечення для створення віртуального екстер'єру.
- Вибір програмних інструментів для 3D-моделювання та візуалізації (наприклад, Autodesk 3ds Max, Blender, Unity, Unreal Engine).
- Аналіз методів збору та обробки даних для створення точної 3D-моделі (лазерне сканування, фотограмметрія, дрон-зйомка).
- Розробка алгоритмів для оптимізації моделі з метою забезпечення реалістичності та високої продуктивності.
- Інтеграція моделі в інтерактивне середовище

## РОЗДІЛ 1.

### Теоретична частина Створення віртуального середовища НаУ ОА(екстер'єр старого корпусу та паркової зони) та програмного забезпечення Blender та ігрового рушія Unreal Engine 5

#### 1.1. Постановка проблеми створення екстер'єру старого корпусу та паркової зони, опис базових понять про Blender

У процесі створення віртуального середовища для Національного університету "Острозька академія" (НаУ ОА), виникає ряд значущих проблем, пов'язаних з віртуалізацією екстер'єру старого корпусу та паркової зони. Однією з головних проблем є відтворення архітектурних деталей та унікальної атмосфери цих локацій в інтерактивному віртуальному форматі.

Старий корпус та паркова зона НаУ ОА володіють власною історією та архітектурною специфікою, що робить їх особливою складовою культурної спадщини університету. Створення віртуального екстер'єру має на меті не лише точне відтворення зовнішнього вигляду, але і збереження духу історії цих місць для інтеграції в освітній процес та культурне співтовариство.

Для вирішення вищезазначених проблем використовується інструментарій, зокрема програма Blender. Blender є відкритим та безкоштовним програмним забезпеченням для 3D-моделювання, анімації, рендерингу, композитингу, редагування відео та інших областей комп'ютерної графіки.

Основні поняття Blender, які використовуються у процесі створення віртуального екстер'єру, включають:

- Меш (Mesh): Основна структурна одиниця 3D-моделі, яка складається з вершин, ребер та граней.



- **Текстури та Матеріали (Textures and Materials):** Використовуються для надання об'єктам реалістичного вигляду та відтворення поверхневих властивостей.
- **Камера та Світильники (Camera and Lights):** Важливі для створення правдоподібної освітленості та відтворення перспективи віртуального середовища.
- **Анімація (Animation):** Дозволяє надавати життя об'єктам та створювати рух віртуального оточення.
- **Рендеринг (Rendering):** Процес створення зображення або анімації з 3D-сцени, що дозволяє отримувати візуально вражаючі результати.

Розуміння цих базових понять є важливим для успішного моделювання та віртуалізації старого корпусу та паркової зони НаУ ОА у Blender.[18] Такий підхід дозволить ефективно вирішувати завдання з відтворення та інтеграції віртуальних об'єктів у навчальний та культурний контекст університету

Blender має багато базових аспектів, які становлять основу для його використання в різних областях комп'ютерної графіки[18]. Давайте розглянемо деякі з найважливіших аспектів Blender:

**Інтерфейс користувача:**

- **3D Вид:** Можливість переглядати та редагувати сцену у тривимірному просторі.
- **Панелі та Вікна:** Різні панелі для редагування об'єктів, текстур, матеріалів, тощо.

**3D-моделювання:**

- **Меші та Інструменти Моделювання:** Використання мешів для створення об'єктів та інструментів для їх редагування.

### Текстурування та Матеріали:

- Розгорнутість текстур: Нанесення текстур на поверхні мешів.
- Налаштування Матеріалів: Визначення властивостей матеріалів для реалістичного відтворення поверхні.

### Анімація:

- Ключові кадри (Keyframes): Встановлення ключових кадрів для анімації об'єктів та камер.
- Граф руху: Відображення та редагування кривих анімації.

### Рендеринг:

- Cycle та Eevee: Два різних рендер-движки для створення зображень та анімацій.
- Налаштування Рендерингу: Визначення параметрів високоякісного відтворення сцени.

### Скульптування:

- Моделювання в режимі скульптування: Використання інструментів для створення складних форм.

### Відеоредагування та Композитинг:

- Відеоредагування: Обробка відео та аудіо, встановлення переходів та ефектів.
- Композитинг: Об'єднання різних елементів та ефектів для створення комплексних зображень.

### Симуляція:

- Симуляція рідини, диму, та інших ефектів: Використання вбудованих інструментів для створення реалістичних симуляцій.

Ці базові аспекти Blender дозволяють користувачам створювати різноманітні проекти, починаючи від моделювання об'єктів до створення анімацій та складних сцен[18]. Підтримка великої спільноти та постійні оновлення роблять Blender потужним та доступним інструментом для багатьох творчих завдань.

Blender вирізняється декількома важливими перевагами порівняно з іншими програмами для 3D-моделювання. Ось деякі з основних переваг:

- **Безкоштовність та Відкритий Код:** Blender є повністю безкоштовним та відкритим для користувачів. Це робить його доступним для широкого кола користувачів, незалежно від їхнього бюджету.
- **Активна Спільнота:** Blender користується великою та активною спільнотою користувачів. Це сприяє обміну знаннями, розробці додаткових засобів та вирішенню проблем шляхом взаємодопомоги.
- **Багатофункціональність:** Blender включає в себе не тільки інструменти для 3D-моделювання, але й для анімації, рендерингу, композитингу, відеоредагування, симуляції, та інших областей. Це робить його всебічним інструментом для творчих завдань.
- **Можливості для Програмування:** Blender має Python API, що дозволяє користувачам створювати власні скрипти для автоматизації завдань та розширення функціоналу.
- **Підтримка Великої Кількості Форматів:** Blender підтримує широкий спектр форматів файлів для імпорту та експорту, що спрощує обмін даними з іншими програмами.
- **Багатоплатформенність:** Blender працює на різних операційних системах, таких як Windows, macOS та Linux, що надає більшу гнучкість користувачам.
- **Потужність та Продуктивність:** Незважаючи на те, що Blender є безкоштовним, він має вражаючий набір функцій та можливостей, які забезпечують високу продуктивність у творчому процесі.

- **Активний Розвиток та Оновлення:** Blender постійно розвивається, з новими версіями, оновленнями та нововведеннями, що додають нові функції та покращують користувацький досвід.

Ці переваги роблять Blender конкурентоспроможним інструментом для багатьох творчих завдань та привертають користувачів різного рівня досвіду.

У світі ігрової індустрії і візуальних ефектів непинно зростають вимоги до реалізму та вражаючої графіки. Однак існуючі рішення часто зіштовхуються із викликами, пов'язаними з обмеженнями ресурсів, швидкістю розробки та здатністю до створення вражаючих візуальних ефектів. Потрібно знайти збалансоване рішення, яке враховує високу якість графіки та оптимізацію продуктивності.

Unreal Engine 5 - це високопродуктивний движок для створення ігор та візуальних ефектів, розроблений компанією Epic Games. Його основні аспекти включають:

- **Nanite Virtualized Geometry:** Система, яка дозволяє робити вражаючі графічні зображення за рахунок безшовної роботи з великою кількістю високорозгорнутих деталей.
- **Lumen Global Illumination:** Технологія глобальної освітленості, яка автоматично визначає та розподіляє світло, створюючи реалістичні відбиття та тіні в реальному часі.
- **MetaHuman Creator:** Інструмент для створення реалістичних цифрових людей з вражаючою якістю деталей та анімацією.
- **World Partition:** Механізм, який дозволяє ефективно керувати та оптимізувати великі світи гри, завантажуючи лише ті області, які знаходяться в полі зору гравця.
- **Unreal Insights:** Інструмент для аналізу продуктивності та визначення факторів, що впливають на роботу гри, що дозволяє оптимізувати розробку та використання ресурсів.

## 1.2 Аналіз аналогічних розробок

У контексті високотехнологічного індустріального сегменту та розробок, спрямованих на використання ігрового рушія Unreal Engine 5 для створення вражаючих візуальних ефектів, важливо провести аналіз подібних проектів та технологічних рішень. Це дозволить виявити успішні практики, визначити можливі обмеження та здобути важливі уроки для оптимізації власної розробки.

Пошукавши в різних джерелах інтернету, ми не змогли знайти подібних проектів у межах студентських робіт.

Можу ще відмітити розробку компанії Skeiron (Рис.1), які продемонстрували модель старого корпусу ОА, але він аж ніяк не є аналогом нашої роботи, так як це не 3D модель, а всього лиш копія зовнішнього виду з поганим текстурованням.



Рис 1. Вигляд середовища НаУ ОА, розробленого компанією Skeiron

*Джерело: [www.oa.edu.ua](http://www.oa.edu.ua)*

## 1.3. Інтерактивна частина розробки для взаємодії з різними об'єктами

Загальна інформація та використання Blueprints.

Blueprints - це візуальна система програмування в Unreal Engine, яка дозволяє розробникам створювати логіку гри та взаємодію об'єктів без написання коду. За допомогою Blueprints розробники можуть створювати складні системи AI, управління персонажем, інтерактивні об'єкти, ефекти та багато іншого, просто перетягуючи та з'єднуючи різні блоки логіки у візуальному інтерфейсі[27].

Використання Blueprints в Unreal Engine 5 дозволяє розробникам зосередитися на швидкій ітерації та прототипуванні своїх ідей, що дозволяє швидше створювати прототипи та випробовувати нові механіки гри[27]. Окрім того, Blueprints також є потужним інструментом для навчання новачків у галузі розробки ігор, оскільки вони дозволяють розглядати та редагувати логіку гри у візуальному форматі, що полегшує розуміння та вивчення процесу програмування в ігровому середовищі.

Важливо відзначити, що Blueprints в Unreal Engine 5 надають розробникам великий ступінь гнучкості та можливості створювати складні системи без необхідності в програмуванні на мові коду[27]. Завдяки візуальному інтерфейсу, розробники можуть швидко визначати логіку гри, встановлювати умови та дії об'єктів, створювати анімації та взаємодіювати з різними елементами гри, що дозволяє значно прискорити процес розробки.

Однією з ключових переваг використання Blueprints є їхня інтеграція з іншими аспектами Unreal Engine, такими як рендеринг, фізика, звук, мережевий геймплей та інші. Це дає можливість розробникам створювати повністю функціональні та іммерсивні ігри, додавати нові елементи та вдосконалювати гру через різноманітні блоки логіки.

Крім того, Unreal Engine 5 продовжує розвивати та покращувати Blueprints, додаючи нові функції та можливості для розробників. Це включає в себе покращену підтримку для VR та AR, розширені можливості роботи з анімаціями та фізикою, а також інші інструменти для полегшення процесу розробки та підвищення якості готового продукту[27].

Level Blueprints проти Blueprint Classes

Level Blueprints і Blueprint Classes - це два різних підходи до програмування логіки гри в Unreal Engine, кожен з яких має свої власні переваги та використання.

1. Level Blueprints: Level Blueprints - це спеціальний тип Blueprints, який прикріплюється до конкретного рівня (level) у вашому проекті. Вони дозволяють вам визначати логіку та взаємодію об'єктів саме на рівні цього конкретного локації[4]. Level Blueprints зручно використовувати для створення логіки, яка відноситься лише до певного рівня, такої як активація подій, взаємодія з об'єктами та інше. Вони легко доступні та редагуються прямо в редакторі рівнів.

2. Blueprint Classes: Blueprint Classes - це шаблони або класи, що містять логіку та властивості об'єктів у вашій грі. Вони можуть бути використані для створення нових типів об'єктів (таких як персонажі, предмети, вороги тощо), які потім можуть бути розміщені на будь-якому рівні. Blueprint Classes дозволяють створювати повторно використовувані компоненти гри та логіку, що дозволяє швидше створювати ігри та підтримувати їх у майбутньому[6].

Отже, хоча Level Blueprints і Blueprint Classes обидва використовуються для програмування логіки гри в Unreal Engine, вони служать різним цілям і можуть використовуватися разом для створення повноцінного ігрового досвіду.

Functions - Macros.

Функції та макроси - це два різних підходи до організації та використання логіки в програмуванні, включаючи ігрову розробку в середовищі Unreal Engine. Кожен з них має свої власні переваги та використання.

1. Функції (Functions):

- Функції - це блоки коду, які призначені для виконання певних завдань або обчислень. Вони можуть приймати вхідні дані (параметри), обробляти їх та повертати результат.

- Функції в Unreal Engine можуть бути створені як частина Blueprint (у візуальному інтерфейсі) або в кодї (у мовах програмування, таких як C++).

- Використання функцій дозволяє структурувати код, роблячи його більш зрозумілим та легким у розширенні та підтримці[4]. Вони також можуть бути використані для уникнення дублювання коду та сприяють більш ефективному управлінню логікою програми.

## 2. Макроси (Macros):

- Макроси - це фрагменти коду, які використовуються для заміни певних шаблонів або підстановок у вихідному коді. Вони використовуються для створення коротших або більш зрозумілих виразів, які можуть бути використані в різних частинах програми.

- У Unreal Engine макроси можуть бути створені за допомогою Blueprint (в Blueprint Editor) або у коді (у C++).

- Макроси корисні для визначення певних шаблонів або повторюваних дій, які можуть бути використані багаторазово[4]. Вони можуть спрощувати та прискорювати розробку, а також забезпечувати більш чітку та конкретну семантику у вихідному коді.

Отже, обидва підходи мають свої власні сценарії використання, і їх вибір залежить від конкретних вимог та завдань проекту.

## 1.4 Використання системи управління проектами Atlassian JIRA

Atlassian JIRA - це один з найпопулярніших інструментів управління проектами, який використовується компаніями по всьому світу для планування, відстеження і управління різноманітними видами проектів[10]. Він дозволяє командам ефективно співпрацювати, використовуючи різні методології розробки програмного забезпечення, такі як Scrum, Kanban та інші.

Основні можливості JIRA включають в себе:

1. Створення завдань і задач: Користувачі можуть створювати різні види завдань і задач, призначати їх учасникам команди та встановлювати терміни виконання.



2. Планування та пріоритизація: JIRA надає можливості для створення спринтів (в Scrum) або докладного планування робіт (в Kanban), де можна встановлювати пріоритети та розподіляти завдання між учасниками команди.

3. Відстеження прогресу: Користувачі можуть відстежувати статус завдань, оновлювати їх прогрес і вносити зміни в розклади в реальному часі.

4. Звітність і аналітика: JIRA надає різноманітні засоби для аналізу прогресу проектів, включаючи графіки спринтів, звіти про продуктивність команди та інші звіти.

5. Інтеграція з іншими інструментами: JIRA може інтегруватися з іншими інструментами розробки програмного забезпечення, такими як Bitbucket, GitHub, Confluence та інші, для автоматизації процесів розробки та обміну інформацією між різними системами.

6. Підтримка різних методологій розробки: Ви можете налаштувати JIRA для роботи з різними методологіями розробки програмного забезпечення, такими як Scrum, Kanban, або використовувати його для управління будь-якими типами проектів.

Загалом, JIRA є потужним інструментом для управління проектами, який дозволяє командам ефективно організувати свою роботу, відстежувати прогрес і досягати своїх цілей[10].

### 1.5 Використання Jira для управління та планування проектами

При початку роботи в Jira Software ми розподілили обов'язки, надаючи кожному учаснику команди спринти з відповідними завданнями для виконання. Кожен спринт має визначений час для виконання і встановлений статус, що відображає фазу розробки кожного завдання.

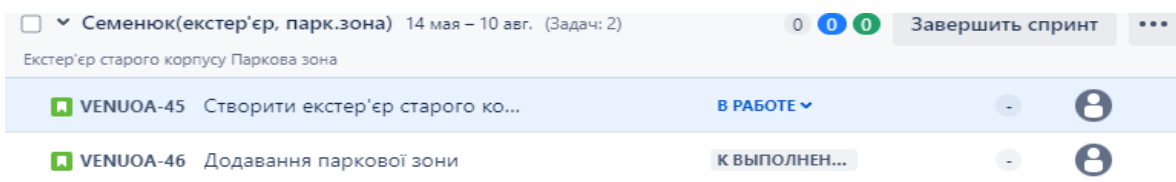


Рис 1.1. Показ наявних спринтів у вкладці «Backlog»

*Джерело: створено автором*

Далі ми можемо бачити умову конкретного завдання після натискання на НЬОГО.

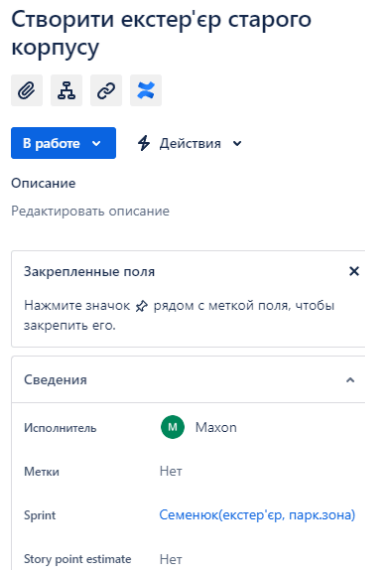


Рис. 1.2. Конкретне завдання

*Джерело: створено автором*

Робочий процес Jira – це послідовність дій команди з ведення завдання через усі етапи, від створення до завершення. На наступному малюнку показано приклад робочого процесу.

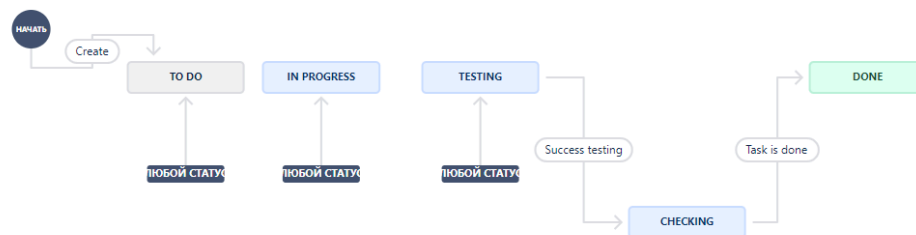


Рис. 1.3. Налаштований робочий процес під наші потреби

*Джерело: створено автором*

На цьому скріншоті можна чітко побачити, через які етапи проходить конкретне завдання, щоб отримати статус «DONE». Наприклад, завдання не може безпосередньо перейти зі статусу «TESTING» до статусу «DONE»,

оскільки спочатку воно повинно пройти через етап «CHECKING», а потім вже перейти в статус «DONE».

Також для кожного релізу існують свої типи:

- Епічні (Epic) - вони відстежують колекції пов'язаних помилок, історій і завдань.
- Помилки (Bug) - вони фіксують проблеми або помилки.
- Історії (Story) - вони відстежують функціональність або функції, які виражені як цілі користувача.
- Завдання (Task) - вони відстежують невеликі окремі частини роботи.
- Підзавдання (Subtask) - вони відстежують дрібні фрагменти роботи, які є частиною більшого завдання.

## РОЗДІЛ 2.

### Прикладна розробка екстер'єру старого корпусу та паркової зони в програмному забезпеченні Blender

#### 2.1. Опис технологічного стеку

Мною було використано програми Blender, Unreal Engine 5, Adobe Photoshop.

Blender - це потужний безкоштовний програмний продукт для 3D моделювання, анімації та створення візуальних ефектів. Він використовується для створення 3D моделей об'єктів, персонажів, анімації, текстур, візуальних ефектів та багато іншого[1].

В даній програмі також використовували аддони, а саме:

**Bool Tool**- надає можливість зручно та швидко вирізати отвори в об'єктах;

**BlenderKit**- надає можливість користуватись безкоштовними та платними матеріалами та моделями.

Unreal Engine 5 - це один з найпопулярніших та потужних ігрових движків, створений компанією Epic Games. Цей движок використовується для створення високоякісних ігор та інтерактивних додатків. Unreal Engine 5 надає інструменти для розробки, рендерингу, анімації та створення вражаючих графічних ефектів.

Adobe Photoshop - це програмне забезпечення для редагування та обробки графіки. Це відомий графічний редактор, який використовується для обробки фотографій, створення різноманітних графічних дизайнів, малювання, ретуші, створення ефектів та багатьох інших завдань, пов'язаних з обробкою зображень.

Усі ці програми є потужними інструментами для створення візуальних контентів у різних галузях: від створення 3D моделей і анімації до редагування фотографій та створення ігор чи інтерактивних додатків.

## 2.2. Збір основної інформації для виконання завдання

Мета роботи: пошук детальних схем старого корпусу

Хід роботи:

1. Ручний розпис місцевості; складання плану.
2. Детальне фотографування території старого корпусу, паркової зони для моделювання усіх елементів та розробки матеріалів.

Для виконання успішного проєкту, мені потрібно було з початку віднайти плани старого корпусу, а також для того, аби виконувати роботу в повному обсязі та зі всіма об'єктами декору, я мав розуміти повну картину зі всіма кольорами. Тому, мною було відфотографовано весь фасад старого корпусу та паркову зону, але, так як план парку я ніде не зміг знайти, мені довелося відобразити його вручну (Рис 1.1), а саме маркером на листку.

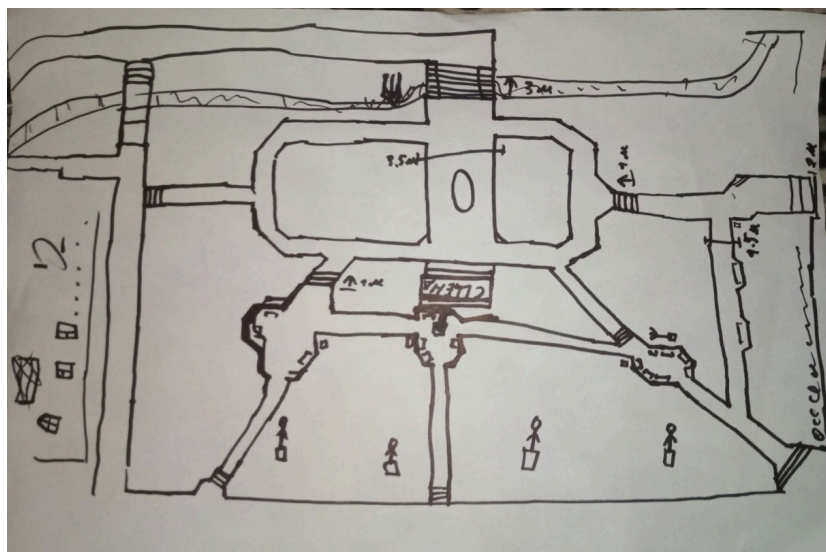


Рис 1.4. План парку

*Джерело: Розробка автора*

Мною було зроблено фотографії фасаду:



Рис 1.5. Зовнішній вигляд фасаду основного корпусу НаУ ОА

*Джерело: Розробка автора*

### 2.3. Створення 3Д моделей в Blender

Мета роботи: за допомогою плану створити контури парку в 3Д

Хід роботи:

1. Створення макету парку;
2. Додавання фасаду академії

За допомогою ручної роботи, тобто створеного мною на листку паперу макету, я задав необхідні розміри та приблизний вигляд паркової зони зверху.

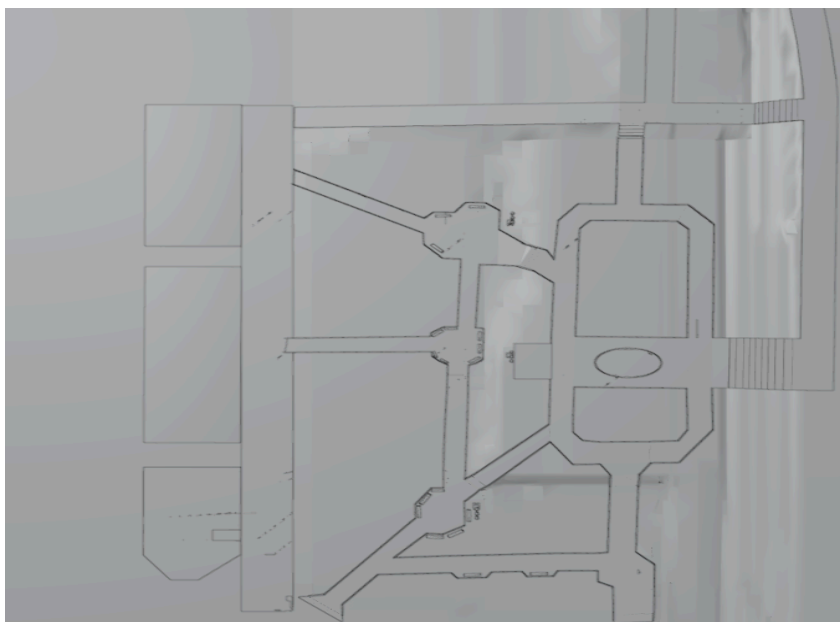


Рис 2.1. Фото макету парку

*Джерело: Розробка автора*

Також завдяки планам та фото, було зроблено початкову 3D-модель фасаду за розмірами та розставленими вирізами під вікна та двері.

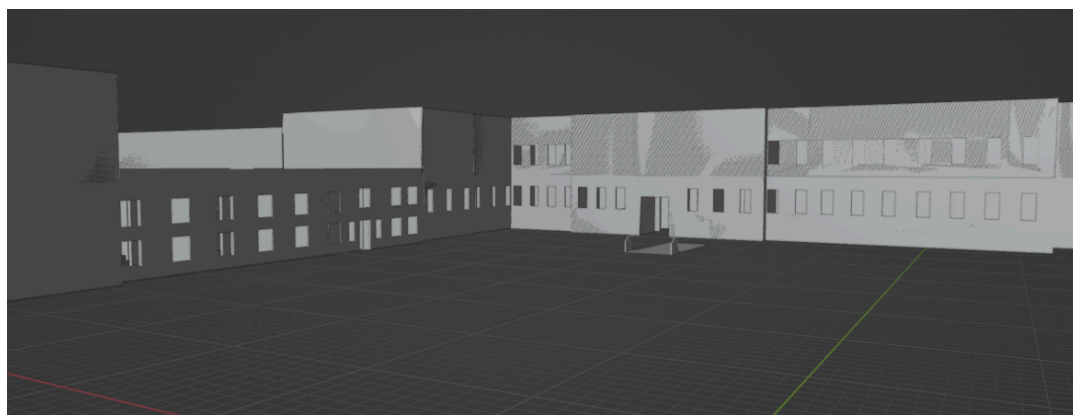
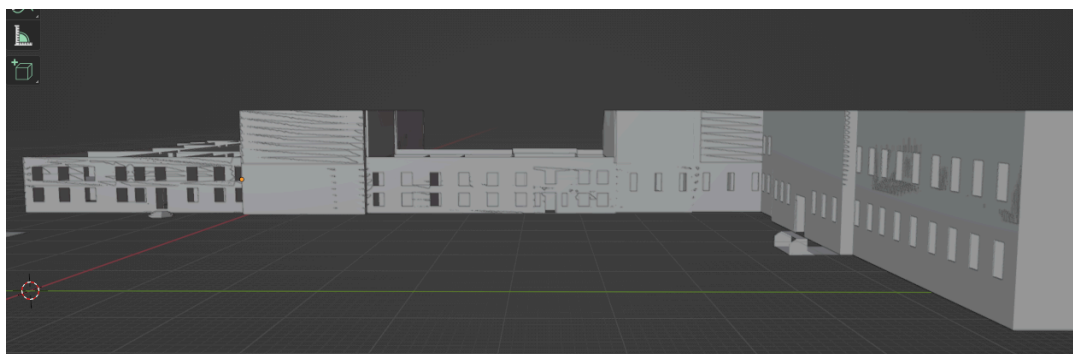


Рис 2.2. Конструювання фасаду академії

*Джерело: Розробка автора*

### 2.3.2 Деталізація екстер'єру

Мета роботи: моделювання елементів

Хід роботи:

1. Створення елементів парку: перебрики , бруківка , сцена , окремі земельні ділянки , окружні клумби , підйоми , ліхтарі , лавки , дерева.
2. Створення елементів фасаду: вікна , двері , криши , проектування стін , паркану , плитки , надписів , декорацій.

Для початку було створено куб, за допомогою якого далі витягувались та розроблялись окремі елементи парку та фасаду. Під час розробки кожного елементу було використано безліч способів досягнення кінцевого результату, тобто за допомогою використання багатьох модифікацій[1].

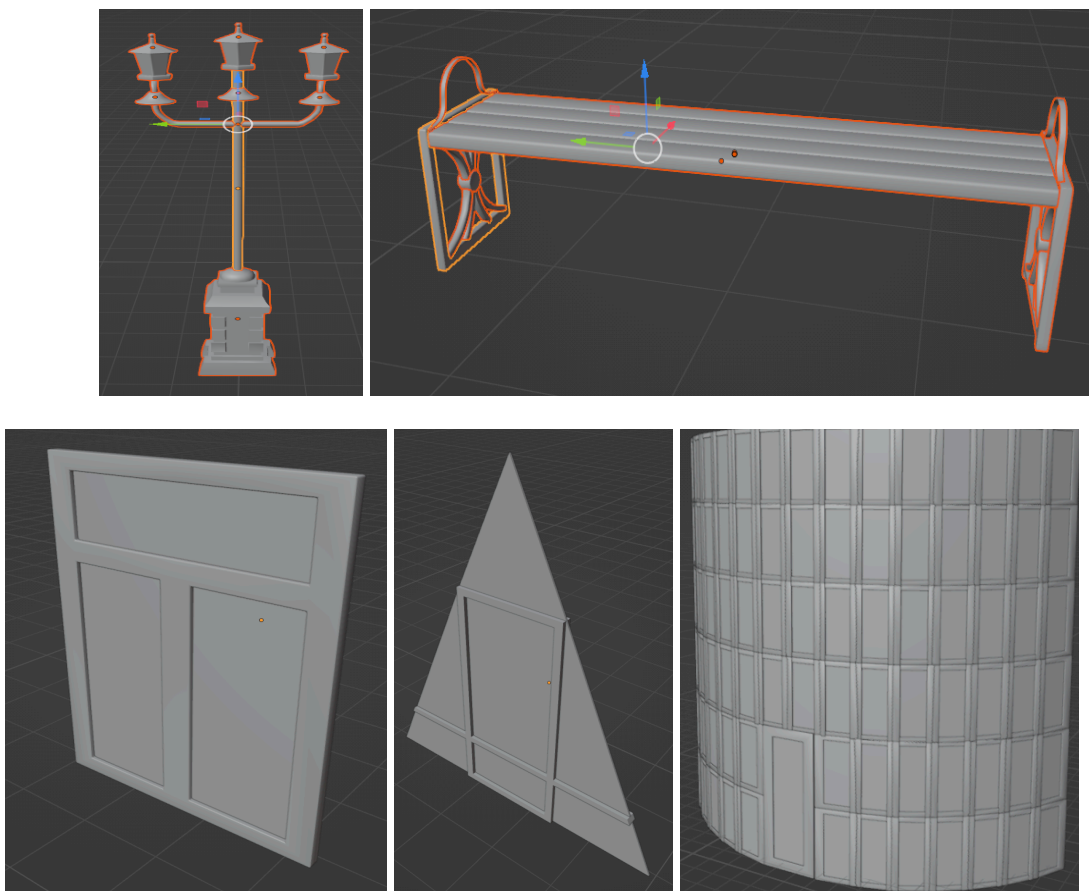


Рис 2.3. Елементи фасаду

*Джерело: Розробка автора*



### 2.3.3 Деталізація об'єктів та шейдерів

Мета роботи: деталізація усіх елементів.

Хід роботи:

1. Створення та підналаштування кольору і матеріалу для текстур та елементів;

Результат:

За допомогою утиліти “Blender Kit” було використано декілька готових важких матеріалів, які зробити самостійно ніяк не вийшло б, через малий досвід користування блендеру, проте, більш простіші матеріали я робив вручну(Рис2.4), фото прикріпив нижче для ознайомлення із нодовою системою.

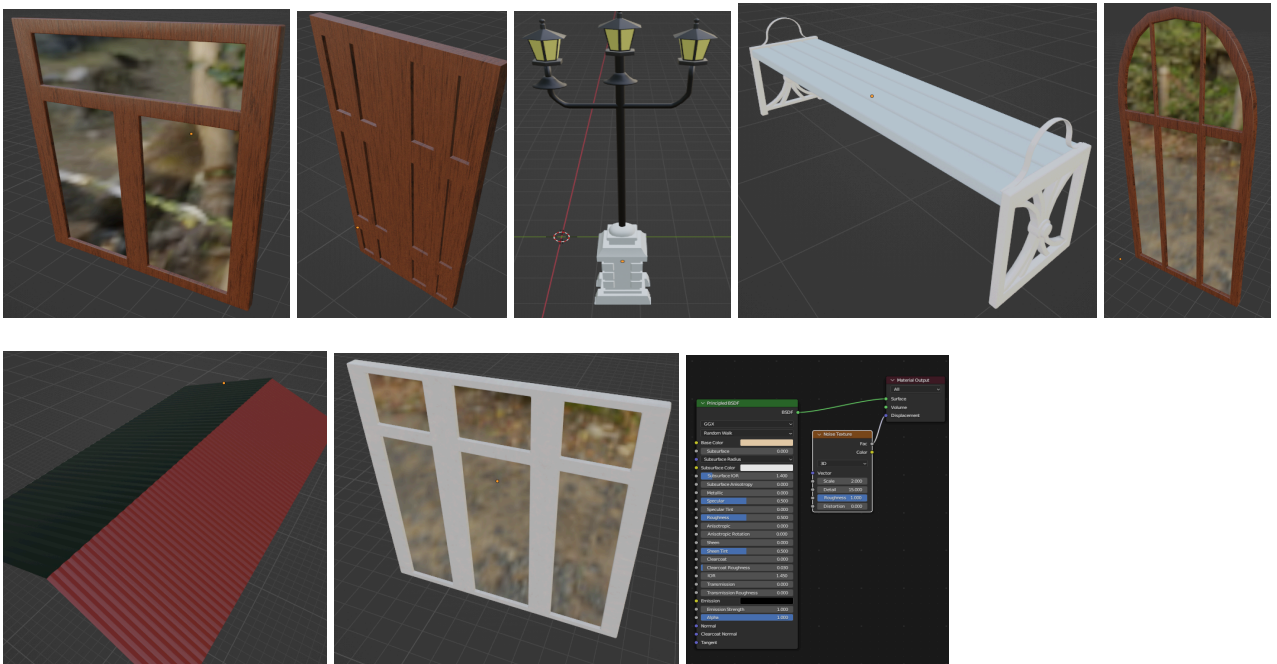


Рис 2.4. Елементи декору з матеріалами та текстурами(+ноди)

*Джерело: Розробка автора*

### 2.3.4 Імпортування елементів та їх розстановка відповідно плану

Мета роботи: імпортувати усі елементи та матеріали до основного фасаду, розстановка декорацій по прилеглий території.

Хід роботи:

1. Імпортувати усі елементи та матеріали до основного фасаду.
2. Розстановка декорацій по прилеглий території.

Результат:

Для початку я підготував усе необхідне та створив і підігнав різні дрібні деталі такі як, вікна, двері, елементи декорацій, даху. Далі приступив до надання правильних форм і виступів на стінах корпусу для найбільш реалістичного вигляд[1]. Після того як надав потрібні форми, почав робити вирізи під вікна та двері за допомогою метода Booltool, розставляти їх по периметру.



Рис 2.5. Вид фасаду з розставленими елементами

*Джерело: Розробка автора*

При роботі з матеріалом фасаду, мною було використано одну з вбудованих шумових текстур, а саме: **Musgrave Texture**. Вона є доступною для створення складних матеріалів у вузлі Shader Editor. Ця текстура отримала свою назву на честь автора інструмента Перліна Musgrave, який розробив цей метод генерації шуму.

Основні параметри текстури Musgrave включають:

- Розміри (Dimensions): Визначають масштаб шуму. Чим вище значення, тим більша деталізація текстури.
- Лакунарність (Lacunarity): Цей параметр контролює "нерівномірність" текстури. Його зміна впливає на ступінь рівномірності або нерівномірності шуму в текстурі.
- Октави (Octaves): Кількість октав шуму, що об'єднуються для створення текстури. Більша кількість октав може призвести до більшої деталізації текстури.
- Сила (Strength): Контролює інтенсивність шуму та його вплив на матеріал.

Текстуру Musgrave часто використовують для створення різних ефектів у текстурах та матеріалах. Вона може бути змішана з іншими текстурами та налаштовуватись залежно від потреб користувача для досягнення бажаного візуального ефекту.

Також мною було використано два модифікатори для розробки економічного входу:

Solidify та Bevel - це два різних інструменти, які часто використовуються в графічних програмах для моделювання 3D об'єктів.

Solidify (Зміцнення):

Solidify використовується для створення товщини у 3D моделі. Він "зміцнює" поверхню, додаючи до неї товщину іншого об'єкту. Наприклад, якщо

у вас є плоска поверхня, Solidify може зміцнити її, роблячи її триразовою або будь-якою іншою товщиною. Це корисно для створення товстості в об'єктах, які можуть бути друквані на 3D принтері або для моделювання об'єктів з реальною товщиною.

Bevel (Фаска):

Bevel використовується для округлення граней або кутів об'єкта[1]. Він додає фаску, що зменшує різкі кути на моделі, замінюючи їх більш округлими кромками або гранями. Це дає об'єкту більш реалістичний вигляд і дозволяє світлу краще відображатись на його поверхні.

Обидва ці інструменти є корисними при моделюванні 3D об'єктів, оскільки вони дозволяють створювати більш складні, реалістичні та деталізовані моделі.

Аналогічна ситуація із парковою зоною, підігнавши до правильних розмірів площу самого парку, надав текстуру трави та зверху стежки із заданим раніше на них матеріалом бруківки. Після чого, було розставлено дрібні деталі, такі як лавки, ліхтарі, смітники та дерева.

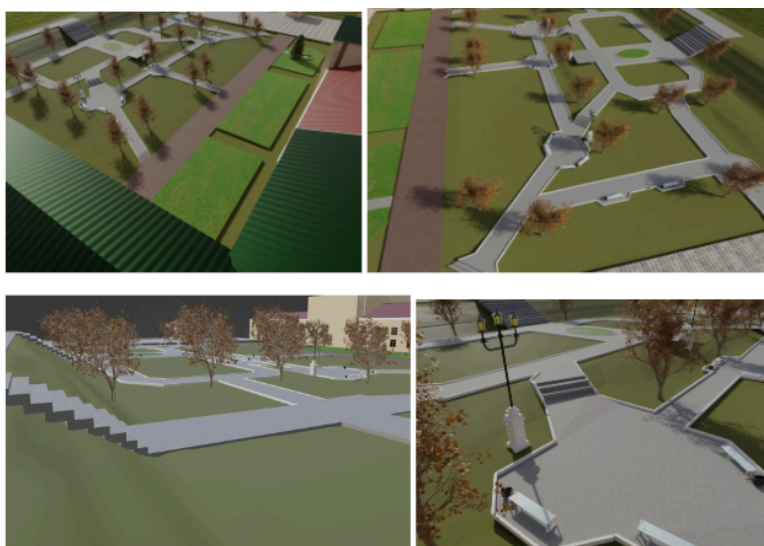


Рис 2.6. Вид парку

*Джерело: Розробка автора*

## Фінальні результати та рендери



Рис 2.7. Фінальні результати рендеру

*Джерело: Розробка автора*

### 2.4. Інтеграція 3D моделей в Unreal Engine

Інтеграція 3D моделей з Blender в Unreal Engine є важливою частиною розробки візуальних ефектів та ігор. Нижче наведено деякі особливості цього процесу:

#### 1. Формати файлів:

- Blender: Під час експорту 3D моделі з Blender, зазвичай використовують формати файлів, які підтримуються Unreal Engine, наприклад, FBX.
- Unreal Engine: Unreal Engine підтримує імпорт різних форматів, але FBX - це один з найбільш поширених і підтримуваних форматів для імпорту 3D об'єктів.

## 2. Матеріали та текстури:

- Blender: При створенні матеріалів у Blender, слід пам'ятати, що Unreal Engine може використовувати зовнішні текстури та матеріали.
- Unreal Engine: Під час імпорту важливо налаштувати текстури та матеріали в Unreal Engine, щоб вони відображалися правильно.

## 3. Скелет та анімації:

- Blender: Якщо ви працюєте з анімованими моделями або скелетними анімаціями, переконайтеся, що вони налаштовані правильно в Blender.
- Unreal Engine: Unreal Engine підтримує імпорт анімацій та скелетних структур. Під час імпорту слід уважно налаштовувати параметри імпорту анімацій.

## 4. Оптимізація:

- Blender: Перед експортом важливо оптимізувати моделі для зменшення обсягу файлу та полегшення завантаження в Unreal Engine.
- Unreal Engine: Unreal Engine має різні інструменти для оптимізації, такі як Level of Detail (LOD) для моделей, які можуть зменшити навантаження на систему.

## 5. Масштаб та одиниці вимірювання:

- Blender: Важливо дотримуватися однакових одиниць вимірювання та масштабу в Blender під час створення моделі.
- Unreal Engine: Unreal Engine може має свої одиниці вимірювання, тому важливо правильно налаштувати масштаб під час імпорту.

## 6. Управління світлом та матеріалами:

- Blender: Ураховуйте взаємодію світла та матеріалів під час створення 3D моделі в Blender.

- Unreal Engine: Unreal Engine має власну систему освітлення та матеріалів, тому деякі параметри можуть потребувати налаштування під час імпорту.

Хочу зазначити , що мною було прийнято рішення розробити проект в UE5 повністю з нуля та взявши із блендера лише окремі елементи декору та фасаду[6].

#### **2.4.2. Виведення усіх конструктивних елементів в Unreal Engine 5**

Мета роботи: за допомогою планів робимо вивід конструктивних елементів в Unreal Engine5

Хід роботи:

1. Додавання плану на робочу поверхню,
2. Створення стін(фасаду) академії ,
3. Розгортання ландшафту парку.

Результат:

Для початку мною було добавлено фотографію плану старого корпусу, по якій я почав зводити стіни та інші конструктивні елементи за розмірами 400x400.

Стіни зменшував та збільшували за допомогою параметрів “Scale” по вісі X,Y,Z

Щоб все чітко спозиціонувати , розміщував усі об’єкти за допомогою параметру “Location” по вісі X,Y,Z, та задавав точну висоту та ширину стін.

Матеріали брав з платформи Quixel Bridge, далі наносив їх на об’єкт, та за допомогою інструмента “UV Project” вирівнював та розтягував до потрібних масштабів.

Quixel Bridge - це програма, яка забезпечує користувачів доступом до бібліотеки сканованих матеріалів, текстур, 3D-моделей та інших ресурсів для використання у проектах CG (комп'ютерна графіка) та візуалізації. Це потужний інструмент, розроблений компанією Quixel, який дозволяє швидко знаходити, переглядати та імпортувати реалістичні ресурси безпосередньо у програми для створення 3D-графіки, такі як Blender, Unreal Engine, Unity та інші.

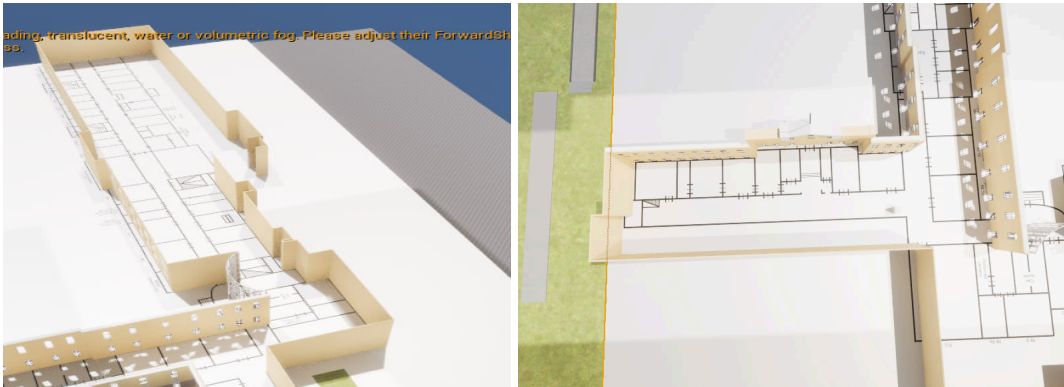


Рис 3.1. Виведення фасаду  
Джерело: Розробка автора

Далі за допомогою методу “Landscape” було розроблено приблизну площу паркової зони, з урахуванням розмірної сітки по якій я робив в блендері.

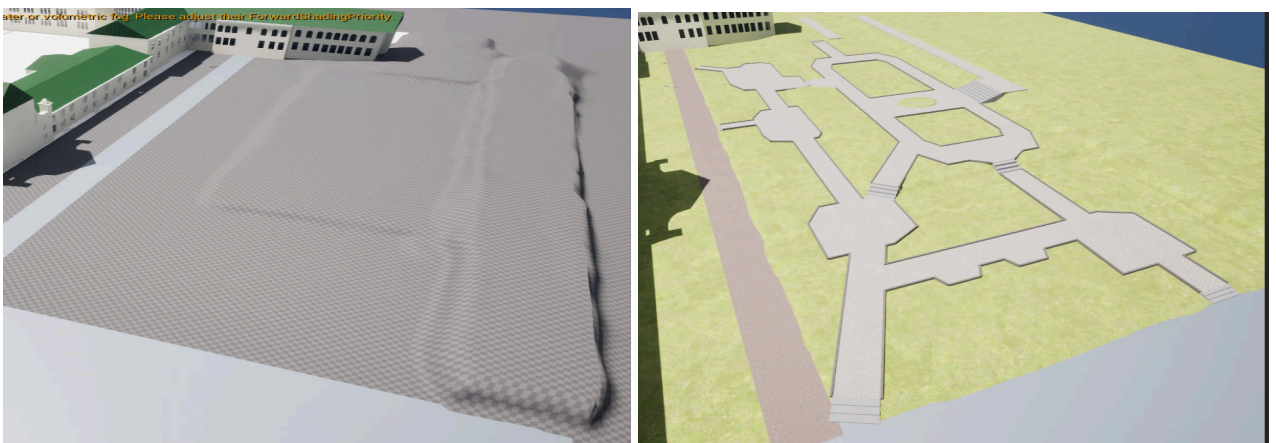


Рис 3.2. Налаштування ландшафту парку  
Джерело: Розробка автора



### 2.4.3. Деталізація екстер'єру старого корпусу та паркової зони в Unreal Engine 5

Мета роботи: деталізація усіх елементів екстер'єру по раніше зібраних фото.

Хід роботи:

1. Імпортування усіх декорацій(вікна, двері, ліхтарі, лавки,окремі елементи фасаду) з Blender;
2. Вибір кольору стін, ландшафту, дверей, вікон, налаштування декору;
3. Розстановка усіх елементів академії;
4. Розстановка елементів та налаштування паркової зони.

Результати роботи:

Перед внесенням 3D-моделі до Unreal Engine, ми ретельно вивчили та розподілили елементи, які можна було обробити в межах самого ігрового рушія та ті, які було доцільніше опрацьовувати у Blender. Цей підхід не просто забезпечив належну інтеграцію, але й запобіг надмірним труднощам під час розробки[25]

Інтеграція 3D-моделей у фреймворк Unreal Engine була важливою частиною створення ігор або віртуальної реальності. Для успішної інтеграції ми ретельно обирали підтримуваний формат, такий як FBX чи OBJ, і враховували його особливості під час моделювання.

Перед імпортом моделі ми переконалися, що всі текстури та матеріали налаштовані належним чином, оскільки під час розробки 3D-моделі в Blender важливо створювати текстури, які належним чином співпрацюють з програмою. Це важливо, оскільки Unreal Engine 5 може не підтримувати окремі матеріали.

Ми також уважно готувалися до інтеграції, обираючи формат FBX, щоб уникнути несумісностей та забезпечити плавну візуальну інтеграцію 3D-моделі

у Unreal Engine. Однак під час завершального результату деякі об'єкти не перенесли належним чином через втрату свого матеріалу[25].

На даному етапі, а саме в енжині також було найважчим підібрати правильні розміри під вікна та висоту будівлі, адже якщо десь щось не співпадає як в реальному житті, то потрібно перероблювати, так як робота повинна бути симулятором того, що є насправді.



Рис 3.3. Фасад академії

*Джерело: Розробка автора*

Матеріали на кожен з окремих об'єктів наносились окремо та змінювались за допомогою функції UV Editor, котра є інструментом для

редагування UV-координат текстур для моделей. UV-координати - це спосіб відображення текстур на поверхні 3D-моделі. UV Editor дозволяє користувачам візуально редагувати та маніпулювати цими координатами для кращого відображення текстур на об'єктах.

Основні функції UV Editor в UE5 включають:

- Редагування UV: Користувачі можуть переміщувати, масштабувати, обертати та трансформувати UV-координати текстур для кожного вершини, грані або об'єкта.
- Управління текстурами: UV Editor дозволяє завантажувати та відображати текстури безпосередньо в редакторі, що допомагає візуалізувати, як текстури будуть виглядати на моделі під час редагування UV-координат.
- Узагальнені інструменти редагування: Включають інструменти, такі як ручне редагування, вирівнювання, згладжування граней та інші функції для точного налаштування UV-координат.
- Відображення змін: Зміни, внесені до UV-координат, відображаються безпосередньо на 3D-моделі в реальному часі, допомагаючи візуально оцінити результати редагування.

UV Editor в UE5 є важливим інструментом для творців контенту, який дозволяє створювати та налаштовувати текстури для 3D-об'єктів, щоб досягти кращого візуального ефекту під час розробки ігор чи візуальних проектів.

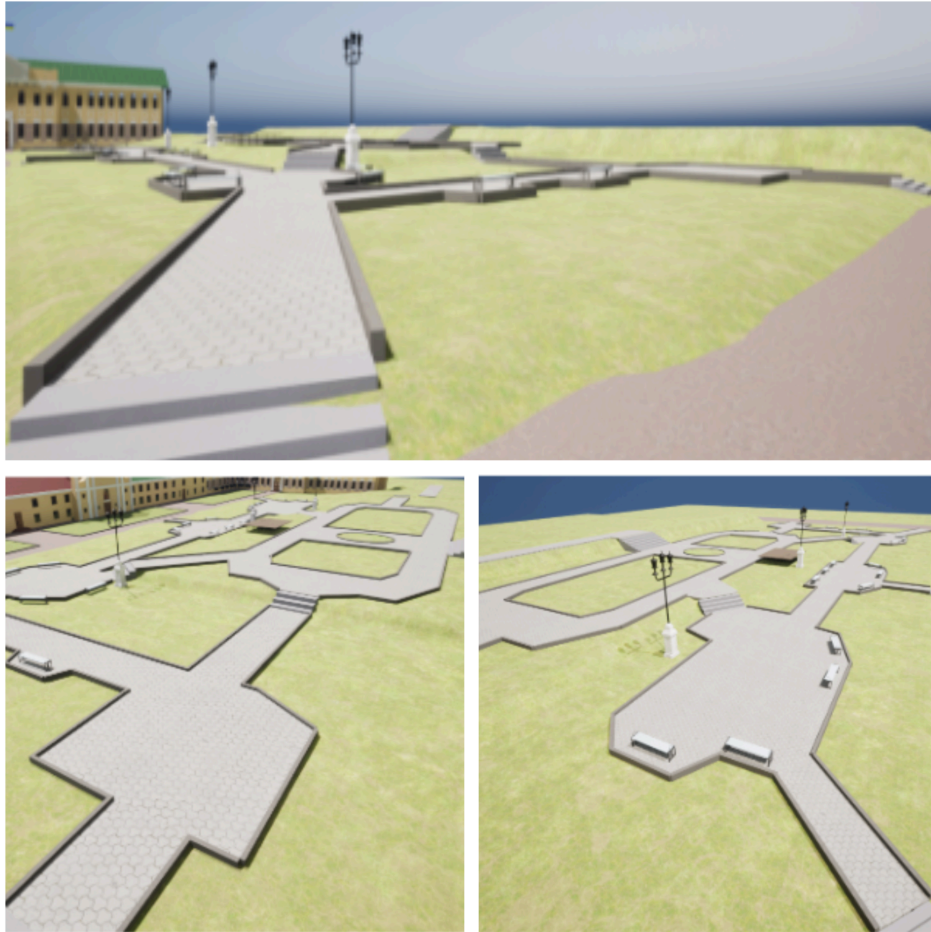


Рис 3.4. Паркова зона

*Джерело: Розробка автора*

У роботі з парковою частиною більш затратним по часу було зробити правильні розміри ландшафту та підналаштувати під нього усі об'єкти.

## **2.5. Інтерактивна частина створення НаУ ОА (екстер'єр старого корпусу та паркової зони)**

У Unreal Engine 5 звуки ходьби на різних поверхнях можуть бути створені за допомогою системи звукових ефектів та налаштувань фізики. Для початку реалізації, на просторах інтернету було знайдено декілька звуків: (трава, бруківка) які в подальшому були імпортовані у наш проєкт[25].

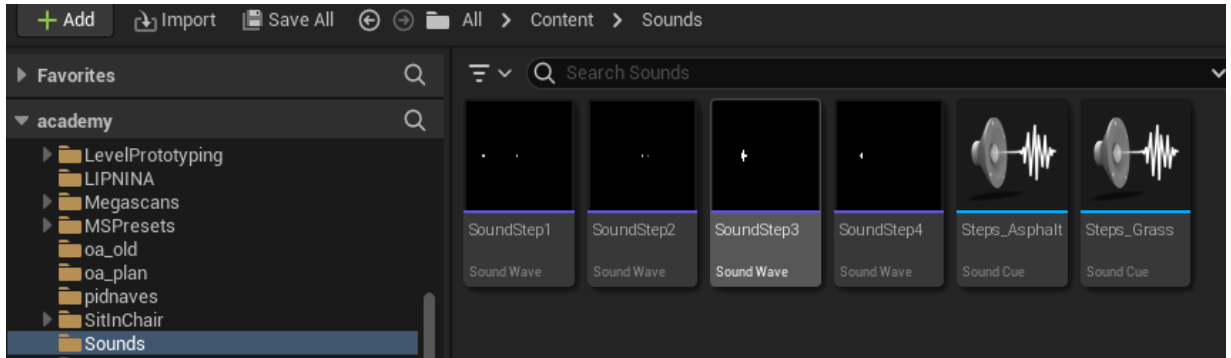


Рис 3.5. Звуки та налаштування Sound Cue

*Джерело: Розробка автора*

Sound Cue в Unreal Engine - це вузлова система, яка дозволяє створювати складні звукові ефекти та обробляти звукові файли в реальному часі[6]. Вона надає широкі можливості для створення різних ефектів звуку, таких як ехо, затримка, фільтрація та багато інших[25].

Ось кілька кроків для створення Sound Cue в Unreal Engine:

- Створення нового Sound Cue: В Unreal Engine перейдіть до вкладки Content Browser та натисніть правою кнопкою миші, оберіть "Sounds", а потім "Sound Cue" для створення нового Sound Cue.
- Додавання звукових елементів: Перетягніть аудіо файли, які ви хочете використовувати, у Sound Cue.
- Побудова графа звуку: Відкрийте Sound Cue та почніть побудову графа звуку, додавши вузли та з'єднуючи їх між собою. Вузли можуть представляти різні ефекти, такі як гучність, панорамування, фільтрацію тощо.
- Налаштування параметрів звуку: Налаштуйте параметри кожного вузла для досягнення бажаного звукового ефекту. Наприклад, ви можете налаштувати гучність, частоти, довжину затримки тощо.

Sound Cue дозволяє створювати складні та реалістичні звукові ефекти у вашому проекті в Unreal Engine, і може бути дуже потужним інструментом для

підвищення іммерсивності вашої гри або додавання атмосфери до вашого досвіду користувача.

Приклад налаштування Sound Cue.

Тут ми можемо бачити, що 2 звуки, кожен з яких використовується під ліву та праву ногу відповідно, відтворюється у випадковій послідовності.

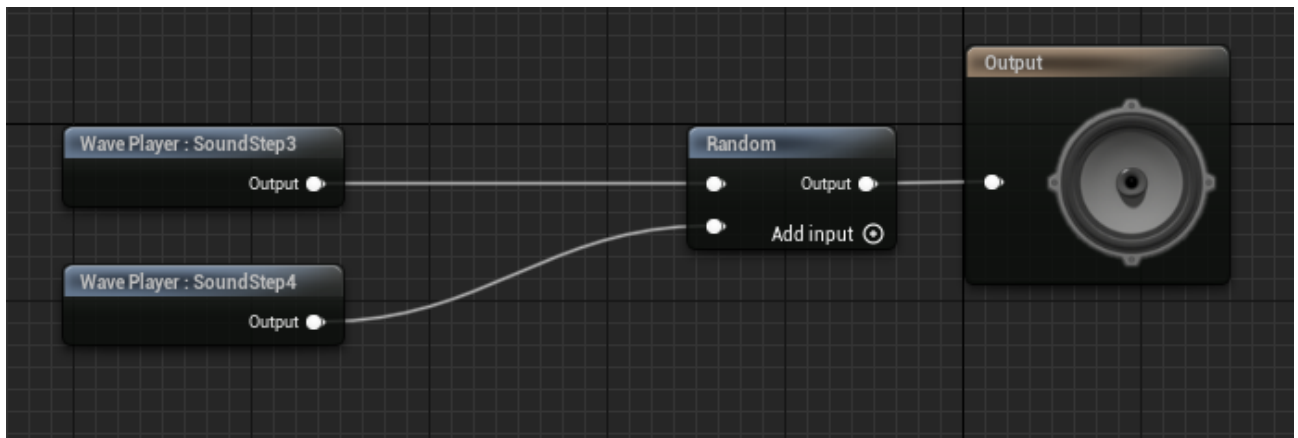


Рис 3.6. Налаштування Sound Cue

*Джерело: Розробка автора*

Наступним кроком потрібно створити фізичні матеріали(трава, бруківка) і додати їх в **Physical Surfaces**.

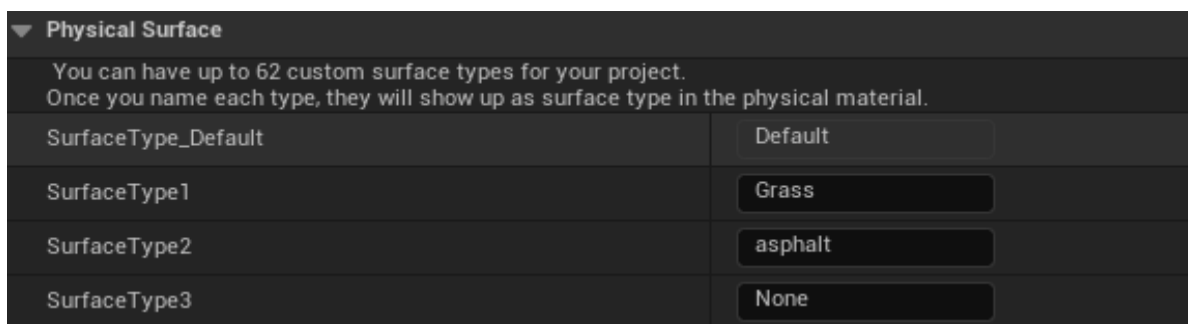


Рис 3.7. Додавання фіз. матеріалів

*Джерело: Розробка автора*

Далі додаємо до нашого фізичного матеріалу саму властивість

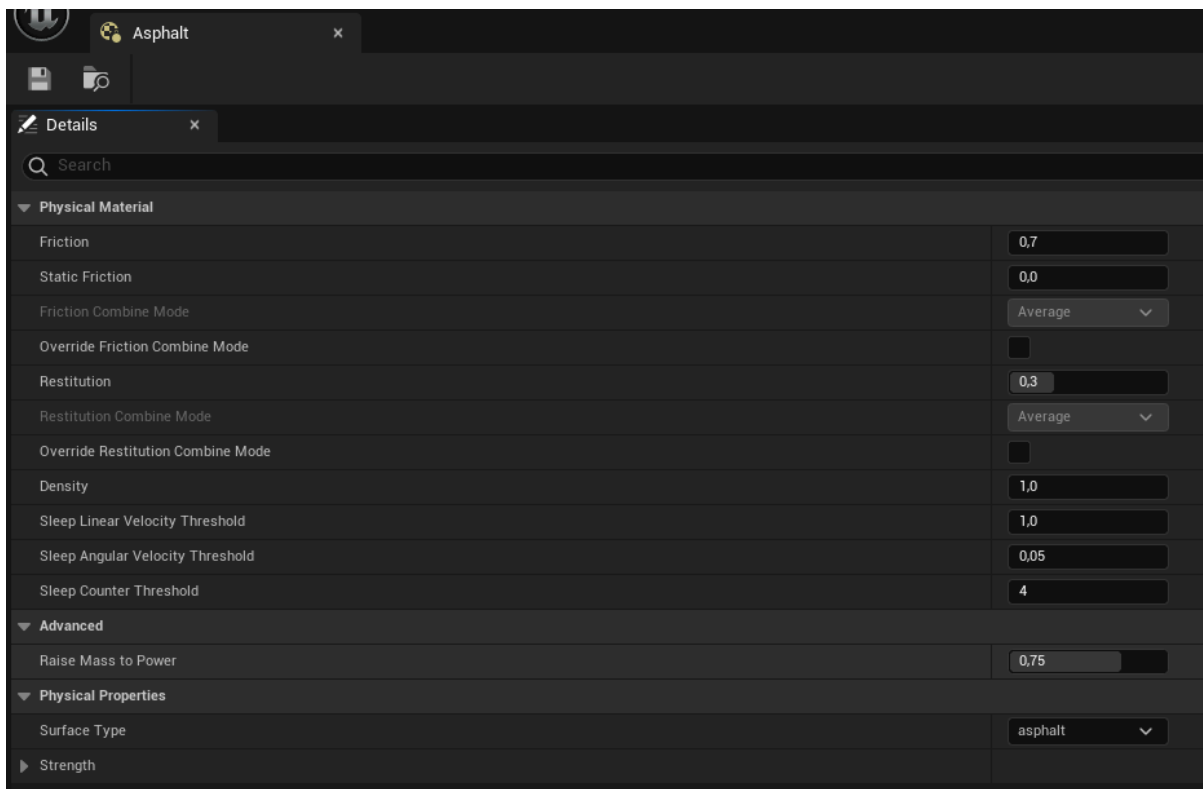


Рис 3.8. Додаємо властивість до фіз.матеріалу

*Джерело: Розробка автора*

Додаємо до нашого основного матеріалу який знаходиться на поверхні(в даному випадку це асфальт) наш фізичний матеріал “Asphalt”

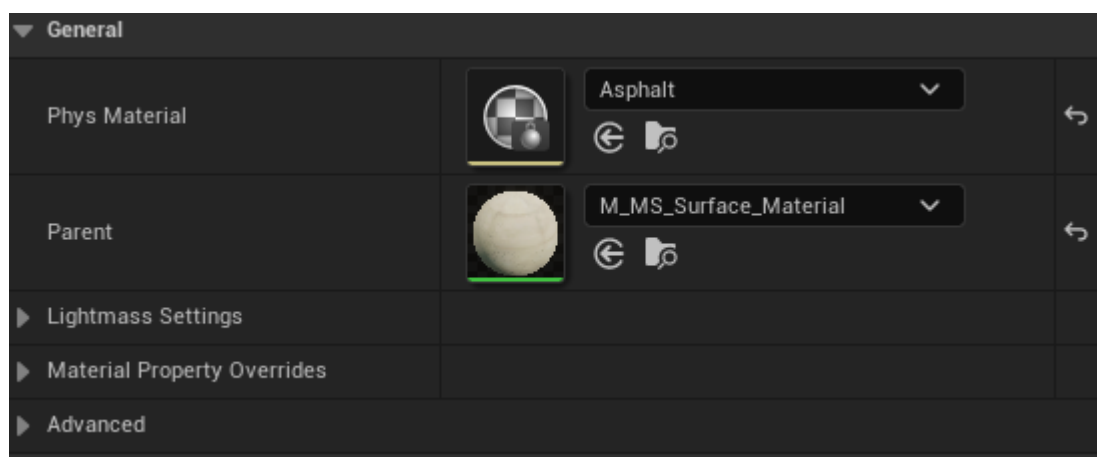


Рис 3.9. Додаємо фіз. матеріал на основний

*Джерело: Розробка автора*

Після чого, переходимо до анімації бігу нашого персонажу, створюємо Notify, та додаємо кілька подій, тобто наших звуків які поєднуються з кроками кожної ноги.

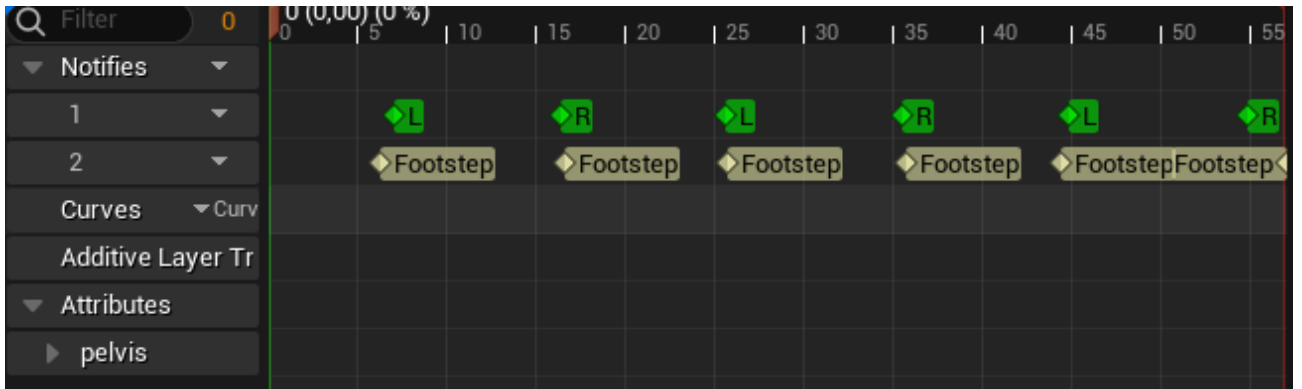


Рис 4.0. Створюємо Notify та присвоюємо йому звук

*Джерело: Розробка автора*

Далі створюємо функцію Footstep в якій робимо зв'язку нодів та присвоюємо звуки.

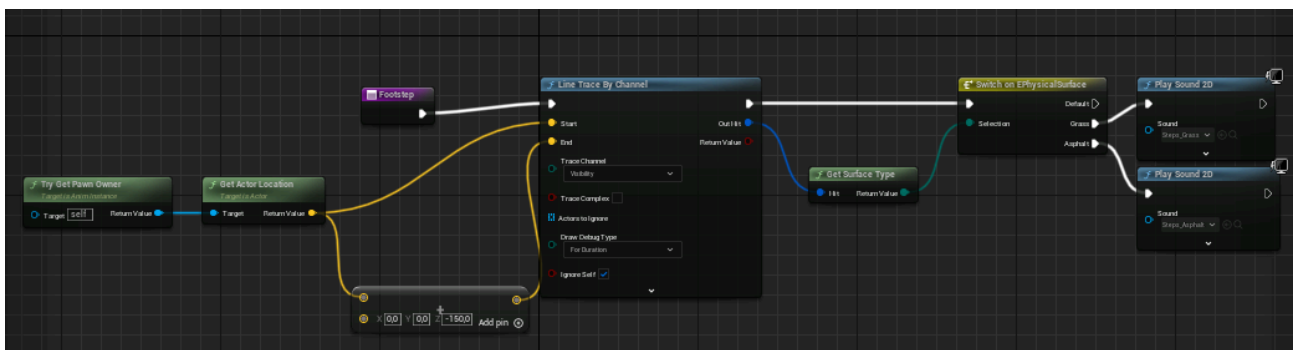


Рис 4.1. Функція FootStep

*Джерело: Розробка автора*

І також, в блюпринті основного персонажу, а саме в EventGraph створюємо подібну систему нодів.



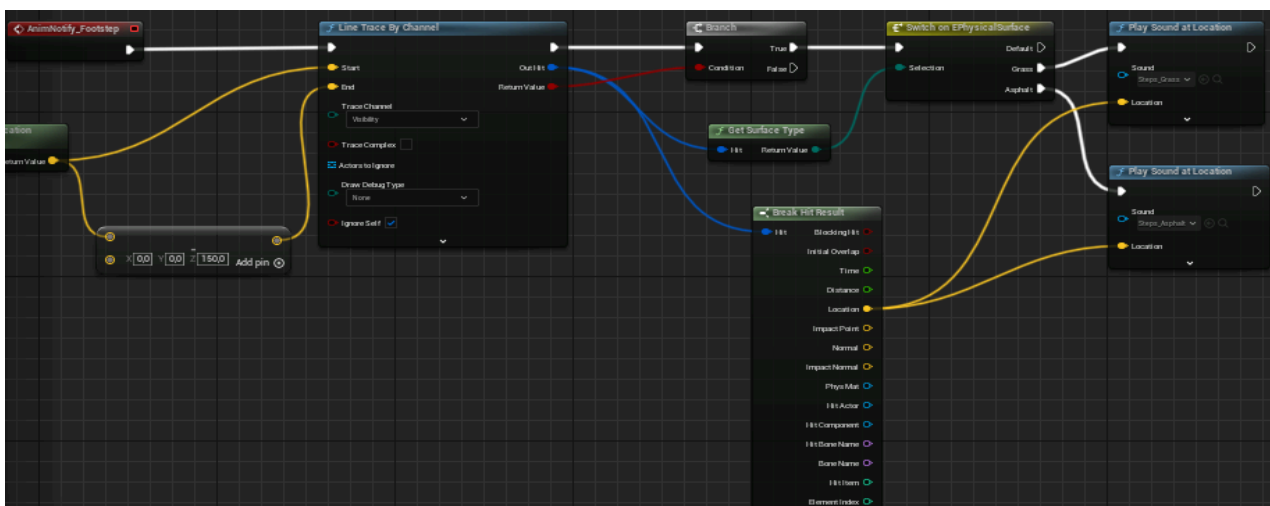


Рис 4.2. Об'єднання нодів для наших звуків

*Джерело: Розробка автора*

Додавання HDR карти:

- Завантажуємо або створюємо HDR карту, яка найбільш подобається. HDR картини часто використовуються для створення реалістичного освітлення та атмосфери[14].
- В Unreal Engine переходимо до вкладки Content Browser та обираємо "Import" та вибираємо нашу HDR карту для імпорту.

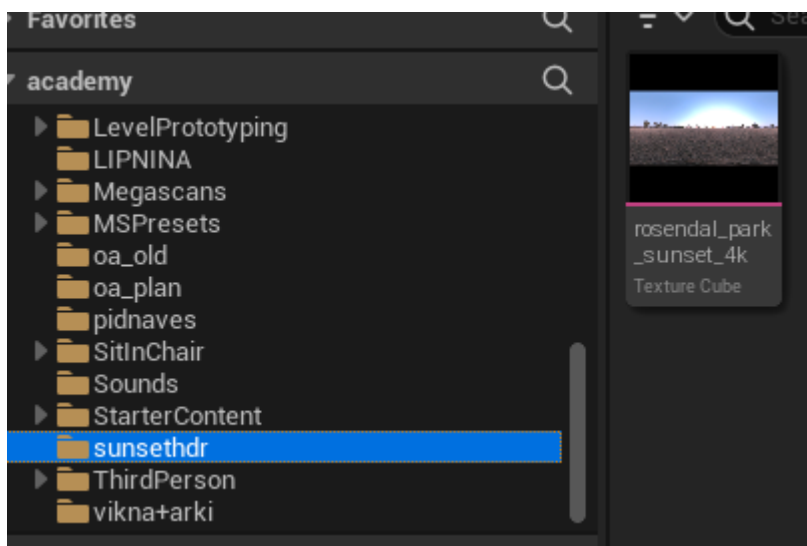


Рис 4.3. Додавання HDR карти

*Джерело: Простори інтернету*

Відкриваємо Blueprint Open level та переходимо до нодової системи в Event Graph.

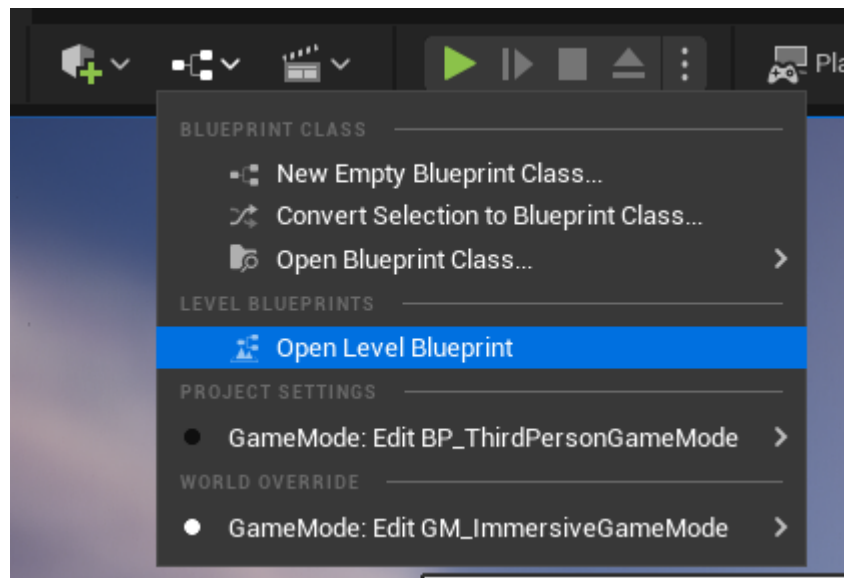


Рис 4.4. Open Level Blueprint

*Джерело: Розробка автора*

Тут у вкладці Variables створюємо нову змінну з типом Float та називаємо її speed(вона відповідатиме за швидкість зміни дня/ночі, тобто переміщення сонця по його осі).

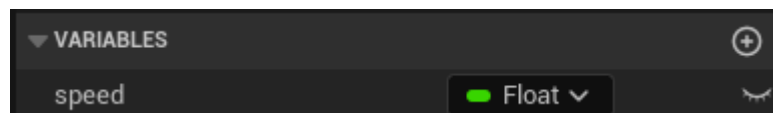


Рис 4.5. Змінна speed

*Джерело: Розробка автора*

Далі створюємо нодову систему за допомогою DirectionalLight.

Directional Light - це спеціальний тип світла, який імітує освітлення від далекого джерела, такого як сонце. Це використовується для моделювання основного джерела світла в сцені, яке проектує тіні на об'єкти[14]. Та за допомогою команди Add Actor Local Rotation, яка використовується для додавання локального обертання до актора, бачимо, що наша змінна Speed під'єднана до функції Make Rotator, а саме осі "Y", за допомогою якої сонце і обертається навколо неї, відповідно змінюючи пору доби з дня на ніч.

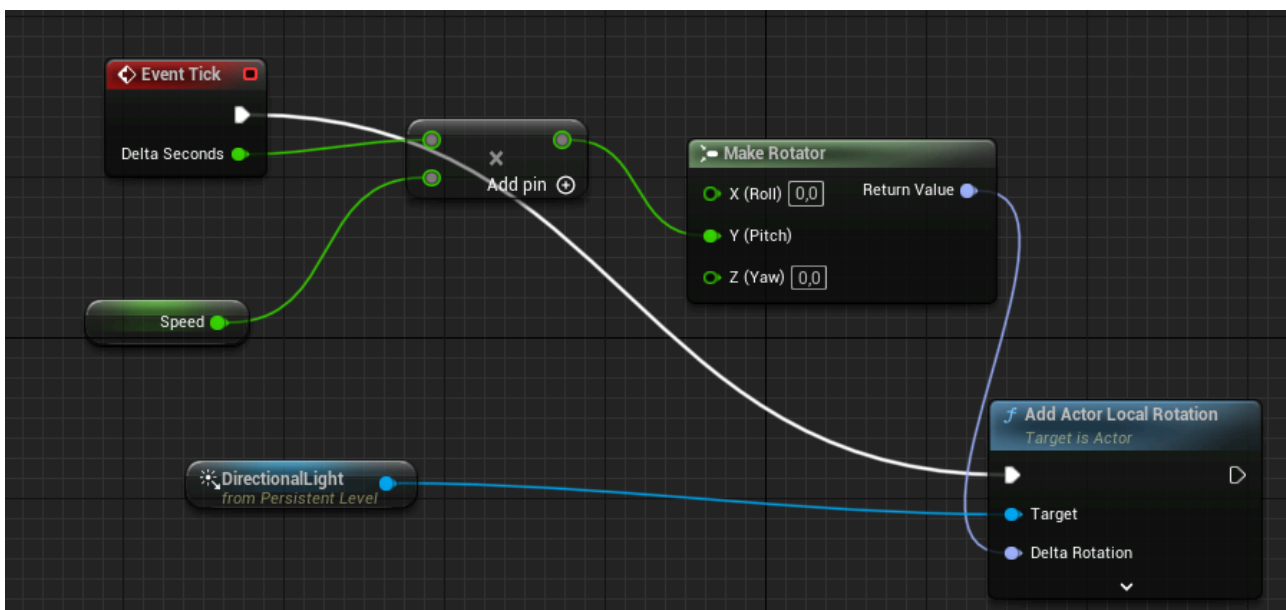


Рис 4.3 Фрагмент нодів який використовується

*Джерело: Розробка автора*

Для створення анімації сидіння в Unreal Engine 5 нам знадобиться виконати кілька кроків.

- Створення анімації сидіння: Створюємо саморуч або завантажуюмо вже готову анімацію сидіння з інтернету для нашого персонажу. Це може бути анімація, яка відтворюється, коли персонаж сідає або встає зі сидіння, а також анімація, яка відтворюється під час самого сидіння.
- Додавання анімацій у контролер анімації: Підключаємо анімацію сидіння до контролера анімацій. У цьому контролері ми можемо визначати, коли відтворюватиметься анімація сидіння.
- Управління анімаціями через Blueprints: Створюємо або відкриваємо Blueprint нашого персонажу. В Blueprint ми можемо налаштувати умови, при яких персонаж почне або закінчить сидіти.

Працювати це повинно наступним образом: коли персонаж підходить до лавки, або об'єкту на який хоче сісти, на екрані з'являється підказка, яку клавішу потрібно натиснути для взаємодії, в нашому випадку це буква E.

Для початку створимо blueprint class з назвою BP\_Chair це буде наша поверхня, на яку ми будемо сідати.

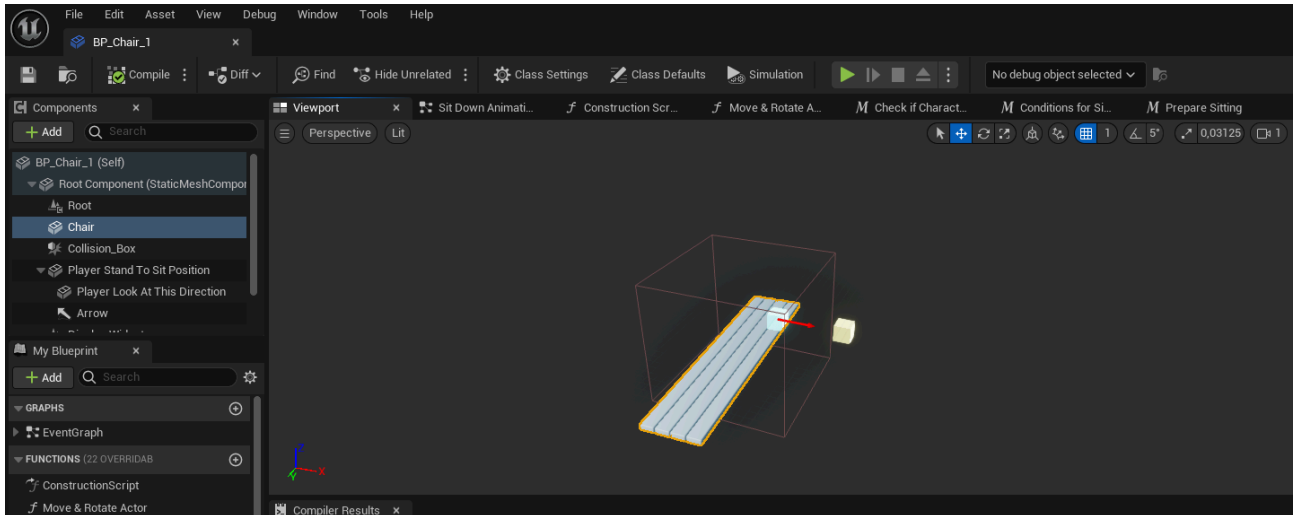


Рис 4.4. BP\_Chair

*Джерело: Розробка автора*

Створюємо 3 змінні з типом Anim Montage:

- коли персонаж сідає
- коли персонаж встає
- коли персонаж сидить

Анімаційний монтаж (Animation Montage) - це один з основних інструментів у Unreal Engine для управління анімацією персонажа. Він дозволяє створювати композиції з різних анімаційних кліпів та керувати їх відтворенням з точністю до кадру.

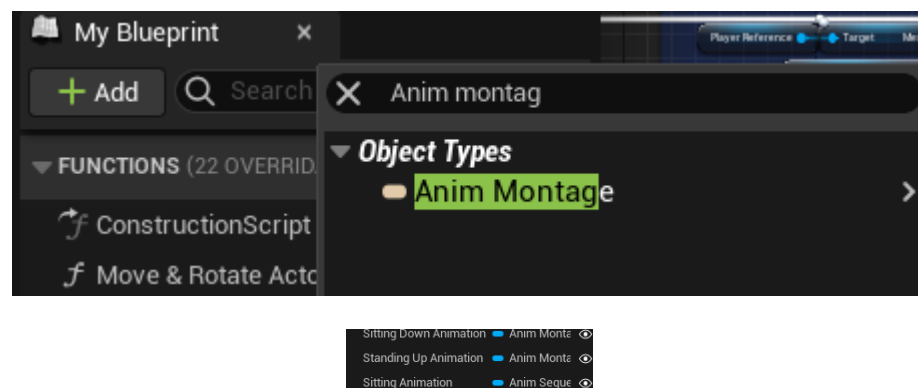


Рис 4.5. Створення змінних

*Джерело: Розробка автора*

Далі у вікні Details в параметрі Default присвоюємо раніше нами заготовлені анімації.



Рис 4.6. Присвоєння самих анімацій

*Джерело: Розробка автора*

Далі розглянемо покрокові макроси, дії яких потрібно провести аби все працювало коректно.

Макрос Prepare Sitting:

Коли наш персонаж підходить до об'єкту, над ним викликається підказка, яку клавішу потрібно натиснути аби провести дію.



Рис 4.7. Приклад підказки для взаємодії

*Джерело: Розробка автора*

Далі розглянемо детально нодову систему.

Checks if character is playing animation. If is, will wait until it finishes - цей вузол перевіряє, чи в даний момент персонаж відтворює певну анімацію. Якщо анімація відтворюється, умова вважається виконаною, і ви можете виконати певні дії відповідно до цього.

Remove "Interact" Text - ми звертаємось до текстового елемента і викликаємо метод або дію, яка видаляє цей елемент зі сцени або приховує його від користувача, тобто коли персонаж вже сів на лавку, текст буде прибрано з поля зору.

Stop player movement input (camera will still work) - це дія, яка призупиняє обробку введення, що контролює рух гравця у грі. Ця дія призупиняє переміщення гравця, але не впливає на рух камери, тобто камера може продовжувати працювати незалежно від зупинки руху гравця, тобто ми можемо вільно оглядати місцевість коли персонаж сидить.

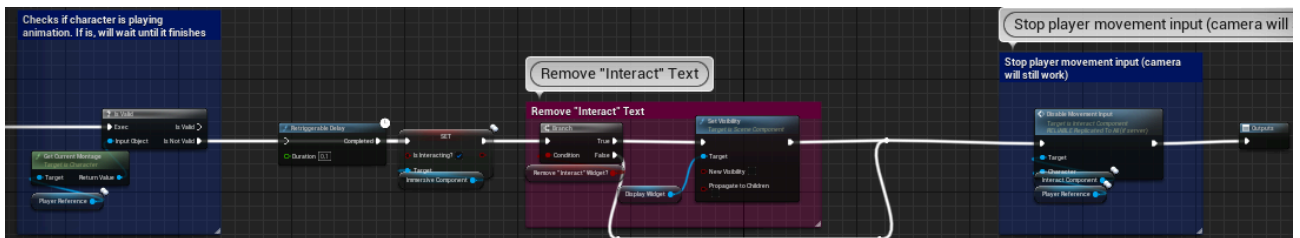


Рис 4.8. Нодова система Prepare Sitting

*Джерело: Розробка автора*

Далі розглянемо нодову систему Character will play sit down animation and move out of chair(коли персонаж сідає та встає).

Основними нодами є:

Sit Down Timeline - це таймлайн, який створений для анімації процесу сидіння персонажа. Він містить ключові точки (keyframes), які визначають рухи та зміни параметрів, необхідні для того, щоб персонаж сів. Наприклад, це може включати зміну положення і обертання персонажа, зміну положення рук і ніг тощо, все це змінюється поступово з часом, щоб створити візуально зручну та реалістичну анімацію сидіння.

Server is sitting - мова йде про сервер та сидіння персонажа, це може означати, що гравець, який взаємодіє з грою через мережу, здійснює дію сидіння на сервері. Тобто, гравець відправляє повідомлення серверу про свою

дію, а сервер обробляє це повідомлення та відтворює анімацію сидіння на всіх клієнтських пристроях для забезпечення синхронізації гри.

Всі інші ноди, які використовуються в даній схемі є додатковими з'єднувачами для коректної роботи даної анімації, тобто (Get, Set, Get Actor Location, Player Reference, Get World Location, Set Actor Location та ін.)



Рис 4.9. Нодова система sit down/move out

*Джерело: Розробка автора*

Наступна і остання функція Character is sitting(коли персонаж сидить).

Розглянемо детально дану нодову систему.

Is character sitting? - у нас є змінна стану "Is Sitting", тому ми перевіряємо її значення за допомогою вузла "Branch" або у Blueprint. Якщо значення "IsSitting" дорівнює true, то персонаж сидить, і ми можемо виконати відповідні дії.

Has animation played? - використовується для перевірки того, чи анімація вже відтворилася, вам знадобиться встановити змінну, яка вказуватиме на те, чи анімація вже відтворювалася. Після відтворення анімації ви можете встановити цю змінну, щоб позначити, що анімація була відтворена.

Is interact button pressed? - перевіряє стан введення кнопки взаємодії за допомогою Blueprint.

Is character falling? - функція "Is Falling" використовується для перевірки, чи персонаж не контактує з поверхнею. Якщо ця функція повертає значення true, це означає, що персонаж перебуває у стані падіння, в нашому випадку сідає на лавку.

Is chair interactable? - використовуємо функцію, перевіряємо її значення за допомогою вузла Branch, що означає, чи крісло є інтерактивним об'єктом, та чи підходить воно для того аби персонаж контактував.

Uncrouches your character if they're crouching - ми використовуємо систему персонажа Character.

Реалізуємо підхід до цієї функціональності наступним чином:

1. Отримуємо доступ до управління рухом персонажа.
2. Перевіряємо, чи персонаж перебуває у стані сидіння (crouching), перевіривши відповідну змінну (наприклад, "IsCrouching").
3. Якщо персонаж сидить, буде викликана функція, щоб змінити стан сидіння на "false" або викличемо функцію "UnCrouch" для того аби персонаж встав, в нашому випадку використовується саме функція UnCrouch, яка передає параметр на наступний нод Outputs, який в свою чергу з'єднується саме через Sit Down та передає далі інформацію чи може персонаж встати - Can Stand Up?



Рис 4.10. Нодова система Character is sitting

*Джерело: Розробка автора*



## ВИСНОВКИ

Протягом періоду моделювання я почав зі створення корпусу та паркової зони Національного університету "Острозька академія" відразу в UE5. Під час роботи над проектом я успішно освоїв роботу з інтерфейсом та механіками в Unreal Engine 5. Цей досвід дозволив мені вивчити потужні можливості даного інструменту та використовувати його для створення вражаючої візуалізації. За час розробки виникли численні труднощі, особливо пов'язані з моделюванням фігур та визначенням правильних розмірів. Вирішення цих проблем вимагало від мене не лише технічної експертизи, але й творчого підходу. Я здатен був ефективно знаходити виходи з складних ситуацій та адаптуватися до змін у процесі роботи. Завдяки роботі з геймдев рушієм, я отримав можливість вільного переміщення персонажа та огляду всіх елементів. Це створило основу для подальшого розвитку проекту та легкої імплементації нових механік фізики. Моя здатність розширювати проект у різних напрямках надає йому потенціал для подальшої інновації. Робота над цим проектом дала мені цінний досвід у візуалізації просторів. Виклики, що виникали при моделюванні, покращили мої навички та вчили ефективно вирішувати проблеми, пов'язані з дизайном та пропорціями. Процес переробки проекту з нуля кілька разів свідчить про мою відданість якості та систематичному вдосконаленню. Навіть у випадках неуспіху я знаходив цінні вивчення, що допомагають мені уникати подібних проблем у майбутньому. Загалом, цей проект не лише став важливим етапом у моєму професійному розвитку, а й підкреслив мою здатність ефективно працювати з великими та складними проектами, забезпечуючи їхню успішну реалізацію. Під час проходження практики, мною було успішно додано три складові у проект, а саме:

1. Загальні анімації та анімації сидіння персонажу: у проекті успішно додані та реалізовані анімації, що дозволяють персонажу взаємодіяти з лавкою. Це додає реалістичності та іммерсивності в гру, створюючи можливість для гравців відчувати більш глибоку взаємодію з оточуючим середовищем.

2. Зміна циклу дня і ночі: Впроваджена зміна освітлення та атмосфери відповідно до часу доби надає імітованому світові реалістичності та динамічності. Ця функціональність створює унікальну атмосферу в грі, що змінюється разом із часом.
3. Додані звуки ходьби по різних поверхнях: У проєкті вдало впроваджено звукові ефекти, які реагують на ходьбу персонажа по різних типах поверхонь.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Blender Foundation. (2023). Blender 3.5 Manual. URL: <https://docs.blender.org/manual/en/latest/> (дата звернення: 5.03.2024 р.).
2. Unreal Engine Documentation. (2023). Unreal Engine 5 Documentation. URL: <https://docs.unrealengine.com/5.0/en-US/> (дата звернення: 13.03.2024 р.).
3. Mullen, T. (2011). Mastering Blender. Wiley Publishing. (дата звернення: 17.03.2024 р.).
4. Sanders, W., & Langley, A. (2022). Learning Unreal Engine: The Official Guide. Packt Publishing. (дата звернення: 24.03.2024 р.).
5. Gahan, A. (2022). 3D Game Textures: Create Professional Game Art Using Photoshop. CRC Press. (дата звернення: 28.03.2024 р.).
6. Interface Unity URL: <https://unity.com/releases/2019-3/editor-tools> (дата звернення: 03.04.2024 р.).
7. Kuckuck, S., Lauter, U., & Weinberg, V. (2018). Game Development with Unreal Engine 4: A Practical Approach. Springer. (дата звернення: 07.04.2024 р.).
8. Adobe Photoshop URL: <http://surl.li/tsatl> (дата звернення: 11.04.2024 р.).
9. Hummel, B., & Resh, P. (2023). Introduction to Unreal Engine 5. Independently published. (дата звернення: 15.04.2024 р.).
10. Introduction to Jira URL: <http://surl.li/taajd> (дата звернення: 18.04.2024 р.).
11. Steele, K., & Derakhshani, D. (2020). Introducing Autodesk Maya 2020: Autodesk Official Press. Sybex. (дата звернення: 21.04.2024 р.).
12. Murphy, J. (2017). Beginner's Guide to Blender. Independently published. (дата звернення: 24.04.2024 р.).
13. Using Widgets in Unreal Engine 5 URL: <http://surl.li/tsaqq> (дата звернення: 27.04.2024 р.).
14. Doull, R., & Shaw, R. (2021). Unreal Engine 5 Shaders and Effects Cookbook. Packt Publishing. (дата звернення: 01.05.2024 р.).
15. Blender Guru. (2023). Blender Beginner Tutorial Series. YouTube. URL: <https://www.youtube.com/playlist?list=PLjEaoINr3zgFX8ZsChQVQsuDSjEqdWMA D> (дата звернення: 04.05.2024 р.).

16. Unreal Engine. (2023). Початок роботи з Unreal Engine 5. YouTube. URL: <https://www.youtube.com/watch?v=I5g1qGI1f64>. (дата звернення: 07.05.2024 р.).
17. Wright, P. (2023). Unreal Engine 5 for Beginners. Independently published. (дата звернення: 10.05.2024 р.).
18. Farkas, B. (2022). Blender 3D Basics: Beginner's Guide. Packt Publishing. (дата звернення: 13.05.2024 р.).
19. Felicia, J. (2022). Architectural Visualization with Unreal Engine 5. Packt Publishing. (дата звернення: 16.05.2024 р.).
20. Specialized Blueprint Node URL: <http://surl.li/tsalz> (дата звернення: 18.05.2024 р.).
21. Cross, D. (2023). Blender Cycles: Materials and Textures Cookbook. Packt Publishing. (дата звернення: 20.05.2024 р.).
22. Physical Material User Guide URL: <http://surl.li/tsaqj> (дата звернення: 10.03.2024 р.).
23. Mitchell, R. (2023). Creating Game Environments in Unreal Engine 5. CRC Press. (дата звернення: 09.04.2024 р.).
24. Creative Shrimp. (2023). Blender Secrets. YouTube. URL: <https://www.youtube.com/c/CreativeShrimp>. (дата звернення: 13.04.2024 р.).
25. Quixel. (2023). Megascans in Unreal Engine 5. YouTube. URL: <https://www.youtube.com/watch?v=Hn7y8rByX1A>. (дата звернення: 20.04.2024 р.).
26. Omura, G. (2022). Blender 3D: Noob to Pro. Independently published. (дата звернення: 17.04.2024 р.).
27. Park, M. (2023). Unreal Engine 5 Blueprints Visual Scripting. Independently published. (дата звернення: 18.05.2024 р.).
28. XRGuru. (2023). Unreal Engine 5 Beginner Tutorial. YouTube. URL: <https://www.youtube.com/watch?v=Ou28IOF3pyY>. (дата звернення: 19.05.2024 р.).
29. BlenderNation. (2023). Blender Tips and Tricks. URL: <https://www.blendernation.com/>. (дата звернення: 23.05.2024 р.).
30. Gamedev.tv. (2023). Unreal Engine 5 C++ Developer. Udemu. URL: <https://www.udemy.com/course/unrealcourse/>. (дата звернення: 25.05.2024 р.).

31. CG Geek. (2023). Blender Quick Tips. YouTube. URL: <https://www.youtube.com/c/CGGeek>. (дата звернення: 20.03.2024 р.).
32. Hough, S. (2023). Unreal Engine 5 for Filmmakers. CRC Press. (дата звернення: 14.03.2024 р.).
33. Blender Cookie. (2023). Introduction to Blender. YouTube. URL: <https://www.youtube.com/c/BlenderCookie>. (дата звернення: 17.03.2024 р.).
34. Epic Games. (2023). Unreal Engine Marketplace. URL: <https://www.unrealengine.com/marketplace/en-US/store> (дата звернення: 14.04.2024 р.).

## ДОДАТКИ

### ТЕХНІЧНЕ ЗАВДАННЯ

#### проєкту

#### **Створення віртуального середовища НаУ “Острозька академія”**

Метою цього проєкту є створення деталізованої і достатньо точної віртуальної копії території корпусів Національного університету “Острозька академія”, яка може бути використана в розробці ігор, віртуальній реальності та WEB-інтерфейсах. Фінальна модель повинна відображати реальний вигляд території, включаючи всі будівлі, споруди, відкриті простори та топографію.

Виконання цього проєкту 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 змодельовані 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, куші 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. На поточному етапі потрібно реалізувати переміщення по створеному середовищі.

- Результати:

Надайте фінальний проект в форматі ueproject, 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 здачі курсової роботи, тому складають до 4 грудня 2023 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 будівлі:

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 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;
  2. Синиця Дмитро: інтер'єр третього та четвертого поверху;
  3. Лаптев Богдан: фасад адмінбудівлі, частина прощі перед корпусом, територія за корпусом;
  4. Белінський Олександр: фасад корпусу, частина прощі перед корпусом.;
  5. Мосійчук Василь: фасад академічного дому та визначених будівель біля нового корпусу.
- Основний корпус
  6. Даніель Демчук: фасад бібліотеки, бібліотечний парк;
  7. Ілля Гупалюк: інтер'єр першого поверху;
  8. Максим Семенюк: фасад корпусу, паркова зона перед корпусом;
  9. Пучко Олександр: стадіон, тенісні корти, фасад господарської частини;
  10. Черевко Вікторія: інтер'єр академічної бібліотеки.

Уточнення щодо певних особливостей виконання проекту:

- в новому корпусі не працюєте з рельєфом біля адміністративної частини;
- в паркових зонах та інтер'єрах статуї не робите, можна зробити лише прості візуально схожі;
- в бібліотеці не відтворюєте живопис купола зсередини;

- при неможливості відтворення ландшафтних елементів в паркових зонах можна їх пропустити, але все інше має бути зроблене;
- будь-які елементи які не можна зараз зробити в точності якими вони є не робите або замінюєте простими аналогами.

Наведені вище уточнення пов'язані з великим обсягом таких робіт або недостатньою наявною інформацією для її виконання. Певні роботи вимагають фізичної присутності на об'єкті моделювання та зняття додаткових замірів. Тому вони будуть виконуватись наступним етапом.