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 НаУОА (інтер'єр першого та другого поверхів нового корпусу)»**

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

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

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

Кошубінського Павла Романовича

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 усіх меблів, взведення усіх конструктивних елементів, розробка інтер’єру першого та другого поверхів нового корпусу, реалізація звуків, розробка принципу роботи комп’ютера, відкривання-закривання дверей і вікон.

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

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 кафедру	31.05.2024 р.	

Студент: _____ **Павло КОШУБІНСЬКИЙ**
(підпис)

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

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

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

Автор: *Кошубінський Павло Романович*

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

Захищена «……»…… 20____ року.

Пояснювальна записка до кваліфікаційної роботи: *69 с., 41 рис., 2 табл., 1 додаток, 30 джерел.*

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

Короткий зміст праці:

Завданням кваліфікаційної роботи/проєкту, було створення інтер'єру першого та другого поверхів нового корпусу. Це інтерактивне середовище може служити засобом представлення загального враження для вступників, які мають можливість віртуально оглянути новий корпус. Використання Blender дозволило створити деталізований інтер'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, interior, Blender, Unreal Engine 5, modeling, compilation, creation, construction.*

Summary of the work: *The task of the qualification work/project was to create the interior of the first and second floors of the new building. This interactive environment can serve as a means of presenting an overall impression to entrants who have the opportunity to virtually tour the new building. Using Blender made it possible to create a detailed interior, and Unreal Engine 5 provided high-quality rendering and interactivity in this environment. The main users of our project will be interested participants and people who are interested in viewing the academy from the outside and inside. This project opens up new opportunities for using technology in the educational process and emphasizes virtual reality to promote universities and facilitate acceptance by future applicants.*

_____

ЗМІСТ

ВСТУП 3

РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА СТВОРЕННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА НАУОА(ІНТЕР’ЄР ПЕРШОГО ТА ДРУГОГО ПОВЕРХІВ НОВОГО КОРПУСА) ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ BLENDER ТА ІГРОВОГО РУШІЯ UNREAL ENGINE 5 5

1.1. Вибір середовища розробки 5

1.2. Аналіз аналогічних розробок..... 16

1.3. Використання системи управління проектами Atlassian JIRA 18

1.4. Інтерактивна частина розробки для взаємодії з різними об’єктами 20

РОЗДІЛ 2. ПРИКЛАДНА РОЗРОБКА ІНТЕР’ЄРУ ПЕРШОГО ТА ДРУГОГО ПОВЕРХІВ НОВОГО КОРПУСА НАУОА) В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ BLENDER 27

2.1. Використання Jira для управління та плануванням проектами..... 27

2.2. Опис технологічного стеку 30

2.3. Збір основної інформації для виконання завдання..... 31

2.4. Створення 3D моделей в Blender..... 32

2.5. Інтеграція 3D моделей в Unreal Engine..... 41

2.6. Інтерактивна частина створення НаУОА (інтер'єр першого та другого поверхів нового корпусу) 47

ВИСНОВКИ..... 61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 63

ДОДАТКИ..... 65

ВСТУП

У сучасному світі, галузь відеоігор стрімко розвивається, привертаючи все більше уваги користувачів. Розробники намагаються інтегрувати нові технології, оскільки ринок перенасичений схожими проектами. Віртуальна реальність та яскравий дизайн допомагають гравцям відпочити та відволіктися від буденних проблем.

Наш спільний проект демонструє нове та оригінальне, тому що такого рішення немає ні в одному університеті України – віртуальне середовище нового корпусу «Острозької академії та прилеглої території» і адаптована інтерактивність з елементами оточення на ігровому рушії Unreal Engine 5, тобто відкриття дверей, вікон, можливість перегляду камер охорони, реалізація комп'ютерів, включення/вимкнення світла в аудиторіях та коридорах, використання ліфту для переміщення між поверхами наближеними до реальності. Ці механіки дозволять усім людям, хто бажає вступити у Острозьку академію, віртуально погуляти по усім коридорам та аудиторіях, і цікавий стартап для нашого університету, який можна розвивати далі і далі.

Unreal Engine 5 є новим ігровим рушієм, розробленим компанією Epic Games. Основною метою його створення є надання геймдевелоперам ще більш реалістичного і вражаючого візуального досвіду, а також полегшення процесу розробки ігор.

Мета практики/кваліфікаційної – це створення інтерактивного віртуального середовища нового корпусу Національного університету «Острозька академія» (інтер'єр першого та другого поверхів нового корпусу) та основні завдання на практику:

1. Розробка звуків на різні поверхні всередині корпусу;
2. Реалізація комп'ютерів для відповідних аудиторій з функціоналом використання браузера для моніторингу інформації;
3. Реалізація закриття/відкриття дверей для аудиторій та туалетів;
4. Закриття/відкриття вікон.

5. Фінальна оптимізація усіх об’єктів та оточення.

Об’єктом дослідження цієї практики/кваліфікаційної є розробка інтерактивного віртуального середовища нового корпусу Національного університету «Острозька академія» (інтер’єр першого та другого поверхів нового корпусу) в середовищі Unreal Engine 5.

Предметом нашого дослідження є перелік програм та застосунків для реалізації нашої задумки, тобто: Blender, Unreal Engine 5, Adobe Photoshop**[4]**, Lidar.

РОЗДІЛ 1
ТЕОРЕТИЧНА ЧАСТИНА СТВОРЕННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА НАУОА(ІНТЕР'ЄР ПЕРШОГО ТА ДРУГОГО ПОВЕРХІВ НОВОГО КОРПУСА) ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ BLENDER ТА ІГРОВОГО РУШІЯ UNREAL ENGINE 5

1.1. Вибір середовища розробки

Порівняння проводили аналізуючи два найпопулярніші та безкоштовні рушії, тобто Unreal Engine 5 та Unity. На початку нашої роботи у нас виникало багато дискусій на рахунок вибору рушія, але все ж таки зупинилися на використанні Unreal Engine 5, тому що він наразі є більш актуальним та сучасним. Розглянемо що таке ігровий рушій та для чого він побудований.

Основні особливості Unreal Engine 5:

Nanite Virtualized Geometry: однією з ключових технологій є Nanite, яка дозволяє відображати велику кількість деталей без значного впливу на продуктивність. Геометрія обробляється в реальному часі, що дозволяє створювати деталізовані ігрові світи.

Lumen Global Illumination: Lumen**[5]** - це система глобальної ілюмінації, яка автоматично розраховує освітлення в реальному часі. Це дозволяє досягти неймовірно реалістичних світлових ефектів та тіней без необхідності ручного встановлення освітлення.

World Partition: ця технологія дозволяє створювати великі ігрові світи, розділяючи їх на краще оптимізовані "комірки". Гравці можуть пересуватися між комірками, і лише та частина світу, яка є в області видимості, обробляється в реальному часі.

Chaos Physics and Chaos Destruction: Unreal Engine 5 включає в себе систему Chaos Physics для реалістичної симуляції фізики та Chaos Destruction для вражаючих ефектів руйнування в ігрових оточеннях.

MetaHuman Creator: додатковим елементом є MetaHuman Creator, інструмент для створення високоякісних ігрових персонажів з вражаючою реалістичністю.

Game Engine — це програмне забезпечення, розроблене спеціально для створення та розробки відеоігор. Він містить набір інструментів, бібліотек і функцій, які дозволяють розробникам ігор створювати графіку, обробляти фізику, керувати штучним інтелектом, впроваджувати аудіо тощо.

Основні компоненти ігрового рушія:

1. Графічний механізм: відповідає за обробку графіки та відображення графічних об’єктів, ефектів освітлення та тіней.
2. Фізичний механізм: відповідає за імітацію руху та взаємодії об’єктів у грі, включаючи гравітацію, зіткнення та інші фізичні явища.
3. Audio Engine: відповідає за обробку звуку та його відтворення в грі.
4. Штучний інтелект (AI Engine): надає можливості розробки для моделювання інтелекту та поведінки штучних персонажів.
5. Інтерфейс користувача (UI): надає інструменти для створення графічних інтерфейсів користувача, таких як меню, елементи керування та інші інтерактивні елементи.
6. Сценарій гри: дозволяє програмістам реалізувати логіку гри за допомогою мови сценаріїв, щоб полегшити розробку та модифікацію гри.

Нами було вирішено розподілити структуру звіту ось так:
1. Опис предметного середовища

2. Аналіз наявних аналогів

3. Реалізація технічного завдання

4. Виконання основного функціоналу для взаємодії

5. Оптимізація та тестування проекту

Unreal Engine (UE) — тривимірний ігровий рушій комп’ютерної графіки, розроблений Epic Games і вперше представлений у 1998 році в шутері від першої особи Unreal. Спочатку розроблений для ПК-шутерів від першої особи, з тих пір він використовується в різних жанрах ігор і прийнятий в інших галузях, зокрема в кіно та телеіндустрії. Написаний на C++, Unreal Engine дуже портативний і підтримує широкий спектр настільних, мобільних, консольних і віртуальних

платформ. Наразі багато розробників хочуть переносити свої проекти на Unreal Engine 5, тому що вважають, що наразі це самий актуальний рушій, який стрімко розвивається. (Рис 1.1.)

Інтерфейс користувача:

- Viewport: основне вікно, де відображається віртуальна сцена.
- Редактори: різні редактори, включаючи редактор сцен, ресурсів, матеріалів, анімації та інших аспектів розробки.
- Панелі інструментів: місце, де доступні різні інструменти для роботи з об'єктами, матеріалами, світлом, анімацією тощо.

Система ресурсів та об'єктів:

- Актори: основні об'єкти на сцені, що можуть бути будь-якою сутністю від персонажів до світових об'єктів.
- Матеріали: визначають вигляд поверхні об'єктів.
- Текстури: зображення, які застосовуються на поверхні об'єктів для створення деталізації та реалістичності.

Робочі процеси:

- Level Design: створення та редагування рівнів (локацій) для ігор чи віртуальних середовищ.
- Lighting: налаштування освітлення для створення атмосфери та настрою сцени.
- Blueprints: візуальне програмування, яке дозволяє створювати складну логіку без писання коду.

Рендеринг та графіка:

- Real-time Rendering: можливість побудови візуально захоплюючих сцен в реальному часі.
- High-fidelity Graphics: здатність створювати високоякісні зображення та спеціальні ефекти.

Оптимізація та продуктивність:

- Performance Optimization: методи та інструменти для підвищення продуктивності та оптимізації гри чи віртуального середовища.
- Multi-platform Development: можлива розробка на різноманітних платформах(PC, Android, MAC).

UE5 відомий своєю потужністю та здатністю створювати вражаючі візуальні ігрові та віртуальні досвіди завдяки своїм інструментам для розробки, візуалізації та оптимізації.

Unreal Engine 5 (UE5) є однією з найновіших версій ігрового рушія, яка пропонує кілька важливих нововведень:

- Nanite: це нова технологія для обробки геометрії, яка дозволяє відтворювати надзвичайно деталізовані об'єкти без необхідності спрощення геометрії. Nanite дозволяє використовувати моделі високої деталізації, що робить візуальний досвід більш реалістичним.
- Lumen: це технологія глобального освітлення в реальному часі, яка автоматично розповсюджує світло по сцені та дозволяє створювати неймовірні ефекти освітлення без необхідності ручної настройки.
- Metahuman Creator: це інструмент для створення гіперреалістичних людських персонажів з використанням сканування людського обличчя. Це дозволяє швидко та ефективно створювати персонажів високої якості для ігор чи віртуальних просторів.
- Quixel Megascans Integration: це інтеграція з Quixel Megascans, що надає доступ до бібліотеки великої кількості фотореалістичних об'єктів, текстур та матеріалів для швидкого створення реалістичних сцен.
- Розширена підтримка мобільних пристроїв: UE5 демонструє розширену підтримку для мобільних пристроїв, дозволяючи розробникам створювати вражаючі ігри та віртуальні досліди для широкого спектру пристроїв.

Ці нововведення UE5 визначають новий стандарт для візуальної якості та продуктивності у світі розробки ігор та віртуальних середовищ. Їх потужність та

можливості роблять цей рушій відмінним інструментом для створення захоплюючих та реалістичних ігрових та віртуальних досвідів.

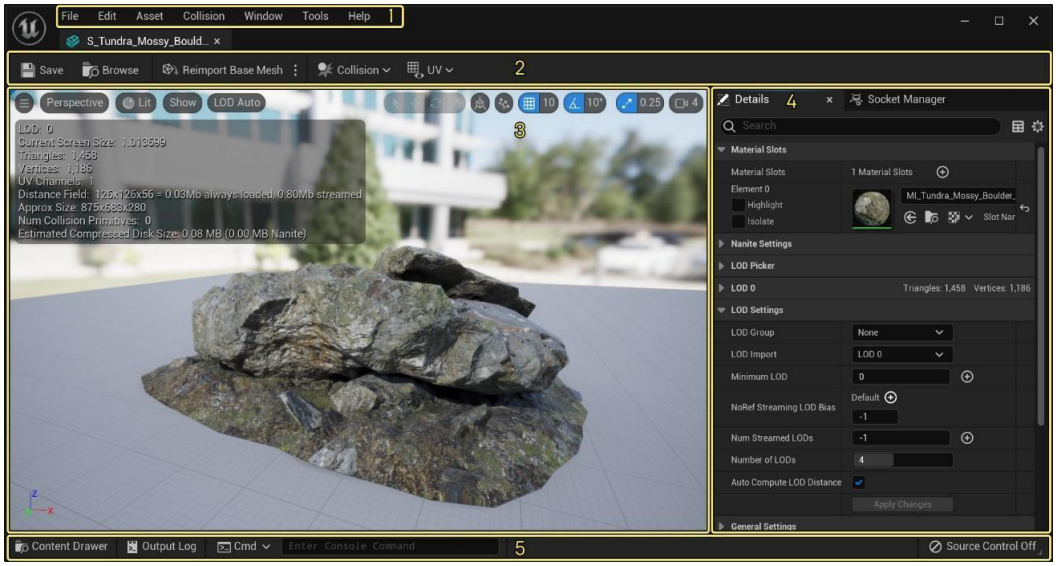

Рис 1.1. Інтерфейс Unreal Engine 5

Джерело: **[6]**

Unity - кросплатформний інструмент для розробки відеоігор і програм, а також двигунів, на яких вони працюють. Програми, створені за допомогою Unity, працюють на настільних комп’ютерних системах, мобільних пристроях та ігрових консолях із двовимірною та тривимірною графікою, а також на пристроях віртуальної чи доповненої реальності. Програми, створені за допомогою Unity, підтримують DirectX і OpenGL. Тут використовується мова програмування C# і має дуже велику спільноту. (Рис. 1.2)

Рис 1.2. Інтерфейс Unity

Джерело: **[7]**

Використовуючи таблицю, проведемо порівняння двох рушіїв.

Таблиця 1.1

Таблиця порівнянь

Характеристика	Unreal Engine	Unity **[8]**
Мова програмування	C++, Blueprints	C#, UnityScript (в архіві), Boo
Графіка та візуалізація	Високоякісна графіка, Nanite Geometry	Зручна візуалізація, Shader Graph **[9]**, HDRP **[10]**
Фізика	Chaos Physics	Unity Physics Engine **[11]**, NVIDIA PhysX**[12]**

Мобільна розробка	Підтримка мобільних платформ	Сильна підтримка мобільних платформ
VR/AR розробка	Повна підтримка VR та AR	Сильна підтримка VR та AR
Ринок	Широкий, включає AAA індустрію та інді	Широкий, популярний серед інді-розробників
Спільнота та документація	Велика спільнота, детальна документація	Велика спільнота, добра документація
Вартість	Безкоштовний (роялті за прибуток), Unreal Engine 4 по GPL	Безкоштовний для особистого користування, платні плани для підприємств
Сумісність із платформами	PlayStation, Xbox, PC, VR, AR, і більше	Android, iOS, PC, Mac, VR, AR, і більше

Вибираючи між цими двома рушіями, ми обрали UE5, тому що він наразі є більш популярний та використовує високоякісну графіку і орієнтований в основному на персональні комп’ютери, а Unity більш адаптований під мобільні додатки та ігри.

Ми обирали серед двох графічних редакторів, це Blender та 3ds Max. Blender - програмний пакет для створення тривимірної комп'ютерної графіки, який включає інструменти моделювання, анімації, візуалізації та постобробки відео. До версії 2.80 він включав ігровий движок Blender для створення відеоігор. Цей пакет є безкоштовним програмним забезпеченням і розповсюджується за ліцензією GNU GPL. У Blender можна робити дуже якісні фото рендери та візуалізацію.

3ds Max - тривимірний графічний редактор, розроблений Autodesk, має багато функцій і включає систему створення та редагування об'єктів і візуалізацій. Він містить найновіші інструменти для дизайнерів, архітекторів, художників і фахівців у сфері мультимедіа. Він працює в 32-бітових і 64-бітових версіях операційних систем Microsoft Windows і Windows NT.

3ds Max використовується для візуалізації моделей будівель, комп'ютерних ігор, рекламних роликів і тривимірних анімаційних мультфільмів. Цього редактора використовували багато анімованих моделей для кінофільмів.

Таблиця 1.2

Таблиця порівнянь двох середовищ 3D моделювання між собою

Характеристика	Blender**[13]**	3ds Max**[14]**
Ліцензія	GPLv2 (безкоштовна та відкрита)	Комерційна, пропрієтарна
Операційні системи	Windows, macOS, Linux	Windows
Інтерфейс	На основі вкладених панелей і вкладок	Зручний інтерфейс з великою кількістю панелей

Рендерінг	Cycles**[15]**, Eevee (інтегровані)	Arnold**[16]**, ART, V-Ray, Mental Ray
Моделювання	Також має вбудовані інструменти	Розширені інструменти для 3D-моделювання
Анімація	Розвинена система ключових кадрів	Велика кількість інструментів для анімації
Симуляція	Підтримка частинок, рідини, тканин	Розширені інструменти для симуляцій
Сценарії та плагіни	Python API **[17]**, велика спільнота розробників	MaxScript, підтримка плагінів
Вартість	Безкоштовний **[18]**	Платний (ліцензія з обліковим періодом) **[19]**

У цих порівняннях, самим основним параметром було те, що ми вивчали середовище Blender раніше на навчальній дисципліні “3D Моделювання” та фактор безкоштовної ліцензії.

Blender - це потужний і універсальний інструмент для 3D-моделювання, який використовується для створення різноманітних об'єктів, включаючи інтер'єри будь-яких будівель. Ось декілька аспектів, що роблять Blender ключовим для моделювання інтер'єрів:

- Моделювання об'єктів: Blender дозволяє створювати складні моделі будь-яких елементів інтер'єру, від меблів до архітектурних деталей. Завдяки

різноманітним інструментам моделювання (включаючи скульптуру, моделювання за допомогою полігонів тощо), користувач може створювати деталізовані об'єкти.

- Матеріали та текстури: Blender надає широкі можливості для налаштування матеріалів та текстур, що дозволяє створювати реалістичні поверхні: від дерева до металу чи скла. Це важливо для відтворення реалістичного вигляду інтер'єру.

- Освітлення: Blender має різноманітні інструменти для керування освітленням, включаючи можливості налаштування різних джерел світла, тіней та ефектів освітлення, що дозволяє створювати атмосферні інтер'єри з різними настроями освітлення.

- Анімація: Blender може використовуватися для створення анімаційних сцен, що дозволяє відтворювати рух об'єктів, зміну освітлення тощо, що робить його корисним і для віртуальних турів або презентацій.

- Безкоштовність та спільнота: Однією з величезних переваг Blender є те, що він безкоштовний. Це відкриває доступ до програми для широкого кола користувачів. Також Blender має велику та активну спільноту, що сприяє обміну знаннями та підтримці.

Ці аспекти роблять Blender важливим інструментом для моделювання інтер'єрів, особливо у контексті створення віртуальних середовищ для університетських корпусів.

Базові аспекти Blender:

1. Інтерфейс користувача:

- Viewport: Основне вікно, де відображається 3D-сцена.

- Панелі інструментів: Місця для доступу до різних інструментів, які використовуються для моделювання, текстурування, анімації тощо.

- Робочі області: Включають області для моделювання, текстурування, анімації, налаштувань сцени та інше.

- Редактори: Можливість відкривати різні редактори, такі як редактор 3D-об'єктів, UV-редактор для текстур і багато інших.

2. Методи моделювання:

- Полігони, вершини та ребра: Основні елементи моделювання. Полігони складаються з вершин (точки) та ребер (лінії), які утворюють поверхні об'єкта.

- Скульптурування: Метод, що дозволяє створювати та формувати об'єкти, використовуючи інструменти, які дозволяють плавно змінювати форму об'єкта.

- Моделювання за допомогою форм: Використання базових форм (куби, сфери, циліндри тощо), які подальше трансформуються для створення складніших об'єктів.

3. Робота з матеріалами:

- Установка матеріалів: визначення вигляду поверхні об'єкта через налаштування параметрів, таких як колір, текстура, блиск тощо.

- Управління текстурами: додавання та редагування текстур для об'єктів, що дозволяє відтворювати реалістичний вигляд поверхонь.

- Налаштування освітлення та відблисків: встановлення параметрів освітлення та властивостей матеріалів для досягнення певних ефектів, наприклад, відблисків, прозорості або матовості.

Ці базові поняття в Blender є фундаментальними для розуміння та створення 3D-моделей та інтер'єрів.

Blender дуже добре інтегрується з ігровими рушіями, у нашому випадку ми експортовували усі об’єкти в Unreal Ungine 5, це доволі просто та цікаво.

Blender є потужним інструментом для 3D-моделювання, і порівнюючи його з іншими програмами, можна виділити кілька ключових переваг:

- Безкоштовність: Одна з найбільших переваг Blender – це те, що вона є повністю безкоштовною. Це значить, що вона доступна для всіх без обмежень, що робить її доступною для студентів, аматорів та професіоналів.

- Універсальність: Blender не обмежується лише моделюванням. Це комплексна програма, яка включає в себе інструменти для моделювання, текстурування, анімації, композитингу, відеомонтажу та багато іншого. Ця універсальність робить Blender цікавим вибором для різних проектів, включаючи створення інтер'єрів.

- Активна спільнота та підтримка: Blender має велику та активну спільноту користувачів. Це означає, що завжди є доступ до багато ресурсів, онлайн-курсів, туторіалів та форумів, де можна отримати допомогу та поради.

- Постійні оновлення та розвиток: Blender постійно оновлюється та розвивається завдяки активній команді розробників. Це означає, що користувачі можуть очікувати нові функції та покращення, що робить програму більш ефективною та потужною з часом.

- Спрощений процес інтеграції: Blender добре інтегрується з іншими програмами та платформами. Наприклад, імпорт і експорт файлів у форматах, які підтримуються більшістю інших програм, відбувається без проблем.

Вибір Blender для створення інтер'єрів може бути обгрунтований цими перевагами, оскільки вона не тільки безкоштовна та має потужний функціонал для моделювання, але й має активну спільноту та неперервний розвиток, що робить її привабливою для широкого кола користувачів у сфері створення віртуальних інтер'єрів.

1.2. Аналіз аналогічних розробок

Переглянувши усю сферу віртуалізацій, то ми не знайшли схожих аналогів у межах нашої країни і тим більше у межах студентських робіт.

Також ще ми знайшли розробку компанії Skeiron (Рис.1.1), які відсканували модель головного корпусу ОА, але це зовсім не той рівень, що плануємо зробити ми, тому що у нас має бути реалізована реальна модель з інтер’єром та екстер’єром, розставлені майже усі меблі, пухвіки та інші елементи декору, ще й до того всього, реалізувати це все на ігровому рушії Unreal Engine

5 з реалізації вільної ходьби персонажем по корпусу і всій прилеглій території, яка відноситься до академії та вільна камера руху. Тому тут порівняння будуть неправильними.

Рис 1.3. Вигляд віртуального середовища НауОА розробленого компанією “Skeiron”

Джерело: **[20]**

Цитата: “*Національний університет «Острозька академія» потрапив на онлайн-простір української культури на платформі Google Arts& Culture – «Україна поруч» (Ukraine is Here), який реалізується Google спільно з Міністерством культури та інформаційної політики України*.”

На нашу думку, ця компанія використовувала якесь програмне забезпечення для сканування поверхні, наприклад є така програма “Lidar”, вона використовує технологію отримання та обробки інформації про віддалені об’єкти за допомогою активних оптичних систем, що використовує явище відбиття світла і його розсіювання в прозорих і напівпрозорих середовищах. Лідар як прилад являє собою, як мінімум, активний далекомір оптичного діапазону.

Концепція LiDAR**[21]** існує з 1960-х років. Коротше кажучи, технологія дозволяє сканувати та картографувати навколишнє середовище, випускаючи лазерні промені, а потім визначаючи час, як швидко вони повертаються. Трохи схоже на те, як кажани "бачать" зі звуковими хвилями, лише з лазерами - що робить його навіть крутішим за Батаранг Бетмена.

Як і більшість футуристичних технологій, він розпочав своє життя як військовий інструмент на літаках, перш ніж став більш відомим як система, що дозволила місії Apollo 15 картографувати поверхню Місяця.

Зовсім недавно LiDAR (також відомий як lidar) був помічений на самохідних автомобілях, де він допомагає виявляти такі об'єкти, як велосипедисти та пішоходи. Можливо, ви також мимоволі зіткнулися з технологією у вакуумному роботі.

Але саме за останні пару років можливості LiDAR справді відкрились. Коли системи стають меншими, дешевшими та точнішими, вони стали життєздатними доповненнями до мобільних пристроїв, які вже мають такі речі, як потужні процесори та GPS - планшети та телефони.

1.3. Використання системи управління проектами Atlassian JIRA

Atlassian JIRA**[22]** — система відстеження помилок, призначена для організації спілкування з користувачами та управління проектами. Розроблено Atlassian у 2002 році. Є дві версії: «хмарна» і серверна. На даний момент JIRA містить три проекти: JIRA Software (для розробників), JIRA Service Desk (підтримка проекту) і JIRA Core (керування проектом). Назва системи (JIRA) була отримана шляхом модифікації японської назви Godzilla («Gojira»), яка, в свою чергу, натякала на назву продукту-конкурента — Bugzilla**[23]**. JIRA була створена як заміна Bugzilla і багато в чому дублює архітектуру останньої. На даний момент (2018) JIRA є однією з найпопулярніших систем управління проблемами.

Основними елементами JIRA є проблеми та робочі процеси. Проблема описує роботу, яку потрібно виконати, і її потрібно назвати й описати. Інші атрибути проблеми можуть бути її пріоритетом, хто її створив і хто має працювати над нею. Залежно від типу проблеми (помилка, завдання, інцидент) властивості можуть змінюватися. Важливим атрибутом є статус (англ. status), який показує, на якому етапі триває робота над питанням. Статус змінюється на основі робочого процесу, створеного для цієї проблеми під час роботи над нею. Робочий процес — це набір дій, які можна виконати над проблемою, щоб вирішити (або змінити) проблему. Найпростіший робочий процес: «Заплановано» - «Виконується» - «Виконано», але для кожного типу проблеми є свій робочий процес. Будь-які зміни проблеми реєструються в журналі активності. У JIRA обидві проблеми та їхні робочі процеси налаштовуються адміністратором, який може створювати власні елементи або змінювати стандартні елементи.

1. Хронології

Використовувати інтерактивну хронологію, щоб спробувати розмістити епіки та зіставити робочі завдання, залежності та релізи на шкалі часу. Хронології дозволяють синхронізувати дії команд та зацікавлених сторін.

2. Керування роботою за допомогою потужних Agile-дощок

Дошки Scrum допомагають Agile-командам розбивати великі та складні проекти на зручні блоки, щоб вони могли зосередитись на важливому та швидше постачати нові версії під час спринтів.

3. Звіти та аналітика

Готові звіти та дашбоарди Jira Software дозволяють командам отримувати найважливішу аналітику в контексті роботи, щоб завжди залишатися в курсі подій та зберігати настрій на успіх.

4. Автоматизація простим перетягуванням

Зосередьтеся на важливому, а все інше зробить автоматизацію. Проста, але потужна автоматизація Jira дуже зручна у використанні.

1.4. Інтерактивна частина розробки для взаємодії з різними об’єктами

Для початку ми почали вивчати документацію для реалізації взаємодії персонажа з різними об'єктами, тому було прийнято рішення використовувати Blueprints.

Отже, розглянемо що таке Blueprints та як їх використовувати.

Blueprint, система візуальних сценаріїв у Unreal Engine, є комплексним рішенням для створення сценаріїв ігрового процесу через інтерфейс на основі вузлів у Unreal Editor. Подібно до багатьох популярних скриптових мов, він використовується для визначення об’єктно-орієнтованих класів або об’єктів у механізмі двигуна.

Ця система є неймовірно гнучкою та потужною, оскільки дозволяє дизайнерам використовувати практично повний спектр концепцій та інструментів, які зазвичай доступні лише програмістам. Крім того, спеціальна розмітка Blueprint, доступна в C++ реалізації Unreal Engine, дозволяє програмістам створювати фундаментальні системи, які можуть бути розширені дизайнерами.

Можливості Blueprints величезні. Незалежно від того, чи хочете ви створити базову гру чи згенерувати процедурний вміст, поекспериментувати з новими функціями чи вдосконалити наявні зусилля програмування – усіх цих цілей можна досягти за допомогою інтуїтивно зрозумілої системи візуальних сценаріїв Blueprint.

Креслення ефективні для конкретних завдань, особливо тих, які включають обробку подій. З іншого боку, якщо ваш Blueprint значною мірою залежить від складних математичних процесів і вимагає високопродуктивного виконання щоразу, може бути корисним замість цього перейти на рідний код C++. Варто зазначити, що функціональні можливості, керовані подіями, такі як керування введенням даних або пом’якшення пошкоджень, як правило, не

вимагають частих покадрових викликів; отже, вони краще підходять для реалізації за допомогою Blueprints, а не низькорівневих мов програмування, таких як C++.

Щоб отримати додаткову інформацію про кодування для креслень або детальну інформацію про компіляцію креслень, зверніться до Технічного посібника з Blueprint.

Цей посібник містить інформацію про прийняття обґрунтованих рішень щодо налаштування ваших Blueprints для забезпечення безперебійної роботи. Він охоплює типові рішення, які користувачі Blueprint мають зробити, і пропонує корисні поради та підказки.

Користувачі, які знайомі з Kismet UE3, можуть знайти креслення рівня впізнаваними, оскільки вони дозволяють вибирати об’єкти на рівні та виконувати над ними дії. Незважаючи на те, що вони ідеальні для створення швидких прототипів і звикання до використання Blueprint, ці креслення можна використовувати лише на призначених для них рівнях. Незважаючи на це, це обмеження дозволяє зручно налаштувати певну функціональність, призначену для Акторів або певного рівня, як-от запуск кінематографічних сцен, коли активуються певні тригери, або надання доступу через певні двері, коли всі вороги переможені.

Класи Blueprint є оптимальним підходом для досягнення повторюваної функціональності у вашому проекті. Включення класу Blueprint до будь-якого з ваших рівнів є можливим, коли ви його створюєте, і вам не потрібно дублювати сценарії, оскільки також можна без зусиль додати скільки завгодно відтворень.

Перенесення поведінки з плану рівня до класу плану є легким завданням. Просто скопіюйте функції, які працюють з акторами, наприклад виклик налаштування яскравості для освітлення, і вставити їх у новий клас Blueprint на основі актора світла. Функції автоматично налаштовуються відповідно до відповідного обсягу.

Зовні і функції, і макроси мають однакову мету, яка полягає в отриманні вхідних даних у вузол, виконанні дій над ним і виведенні результатів. Вони відрізняються підходами, але мають деякі спільні риси:

Кожна платформа має централізоване розташування, де можна редагувати функціональні можливості вузлів — макроси в їхньому мікрографії, а функції — у їхньому функціональному графіку. Будь-який вузол, який посилається на оновлений мікро або функцію, також отримає оновлення, щойно в цьому центральному розташуванні будуть внесені зміни. Вони обидва багаторазові.

Локальні змінні підтримуються як макросами, так і звичайними функціями. Однак у випадку макросів локальні змінні реалізуються анонімно без будь-якої конкретної ідентифікації, але можуть містити значення, як і інші іменовані локальні змінні.

Інкапсуляція функціональних можливостей для багаторазового використання та чіткості є винятковими методами.

Коли ви заглиблюється у складніші сценарії використання, певні принципові відмінності ставатимуть дедалі помітними.

Коли ви розміщуєте вузли для виклику функції, саме тоді функції активуються. Це дозволяє націлювати певні функції на інші об’єкти та полегшує зв’язок між Blueprints.

Під час компіляції Blueprint макроси дублюють усі вузли з макрографу та замінюють їх замість початкового вузла макросу. Це призводить до реплікації всіх підключених вузлів для заміни кожного екземпляра макросу в скомпільований кодовій базі.

Через їх основну роботу існують певні розбіжності в доступних функціях.

Ви маєте свободу використовувати будь-який вузол для класу, до якого прив’язані ваші макроси, а саме той, який ви вибрали під час створення макросу або визначений у локальному макросі. Таким чином, макроси забезпечують додаткову гнучкість щодо вибору вузлів.

Основна відмінність між функцією та макросом полягає в тому, що хоча ви можете вставляти сплячі вузли в останній, така опція недоступна для першого.

Ви можете змінювати поведінку функцій у дочірніх Blueprints, надаючи їм унікальні функції. Візьмемо, наприклад, автомобіль Blueprint із власною функцією «PlayerInteractedWithMe», яка відтворює звуковий сигнал під час виклику. Якщо ви створите двох дочірніх елементів із цього плану — один представляє поліцейську машину, а інший — пожежну машину, — кожен із них може мати свою версію PlayerInteractedWithMe: поліцейська машина відтворювати звуки сирени під час увімкнення світла; натомість шаблон «Пожежна машина» може викидати воду. Однак макроси не підтримують такі основні можливості, як описано тут!

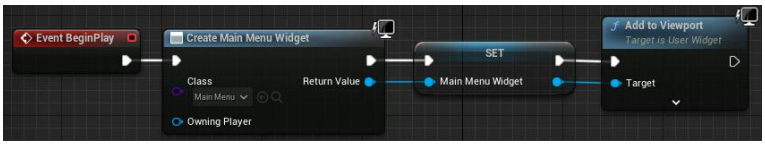

Рис 1.4. Приклад додавання віджета використовуючи Events

Джерело: **[24]**

Сам віджет не повинен бути видимим. Наприклад, віджет Grid Panel рівномірно розподіляє простір між вмістом. Користувач не може бачити панель сітки, але може бачити її ефект.

Віджети також можуть містити інші віджети.

Ми навіть можемо створити віджет як цілий інтерфейс, наприклад екран меню. Нижче наведено приклад віджета, який виглядає як титульний екран. Усі елементи інтерфейсу також є віджетами та містяться у віджеті титульного екрана.

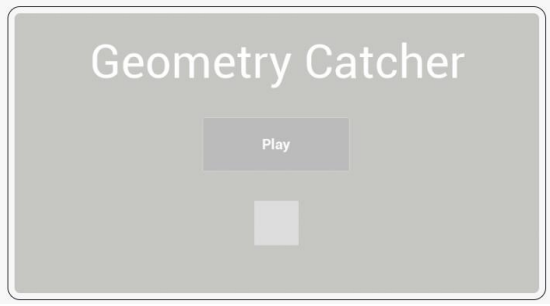

Рис 1.5. Приклад вигляду інтерфейса, за допомогою Widgets

Джерело: **[25]**

Після відкриття віджета, маємо наступні робочі області:

1. Дизайнер: ця область містить візуальне представлення вашого віджета. Панорамуйте, утримуючи праву кнопку миші та переміщаючи мишу. Масштабуйте, прокручуючи колесо миші.
2. Деталі: властивості будь-якого вибраного вами віджета відображатимуться тут.
3. Палітра: список усіх віджетів, які можна використовувати. Тут також відображатимуться будь-які створені користувачем віджети.
4. Ієрархія: список усіх віджетів, які ви зараз використовуєте.
5. Анімація: у віджетах можуть бути анімовані певні властивості, наприклад положення та розмір. Ця панель містить усі ваші анімації.
6. Часова шкала: коли ви вибираєте анімацію, на цій панелі відображатимуться властивості анімації та ключові кадри.
7. Режим редактора: тут ви можете переключатися між режимами конструктора та графіка. Режим Graph майже ідентичний режиму Event Graph Blueprint.

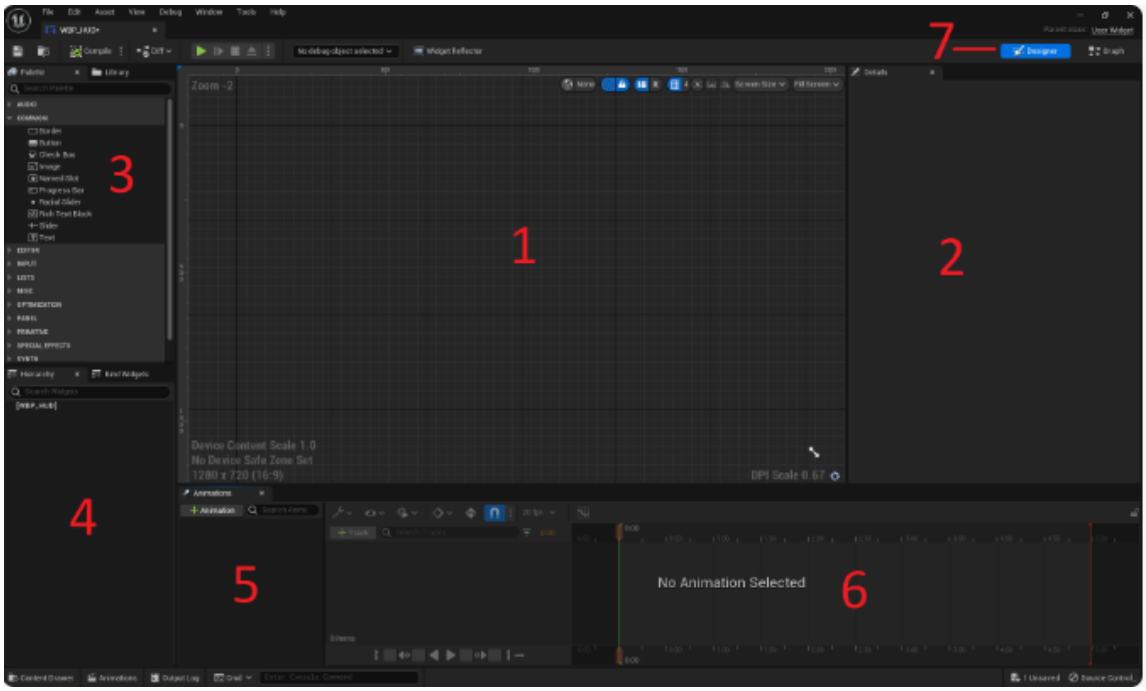

Рис 1.6. Показані робочі області

Джерело: **[26]**

Nodes в EventGraph, які називаються Events, ініціюються з ігрового коду для активації унікальної мережі. Вони дозволяють Blueprints виконувати певні дії у відповідь на певні ігрові події, наприклад, коли гра починається, рівні скидаються або гравці отримують пошкодження.

Blueprints дозволяють реалізувати нову функціональність або розширити функціональність за замовчуванням за допомогою доступу до подій. Кілька подій можуть бути використані в одному EventGraph; однак одночасно можна використовувати лише один тип події.

Якщо ви бажаєте ініціювати кілька дій за допомогою однієї події, їх потрібно зв’язати лінійно, оскільки подія може виконувати лише один об’єкт за раз.

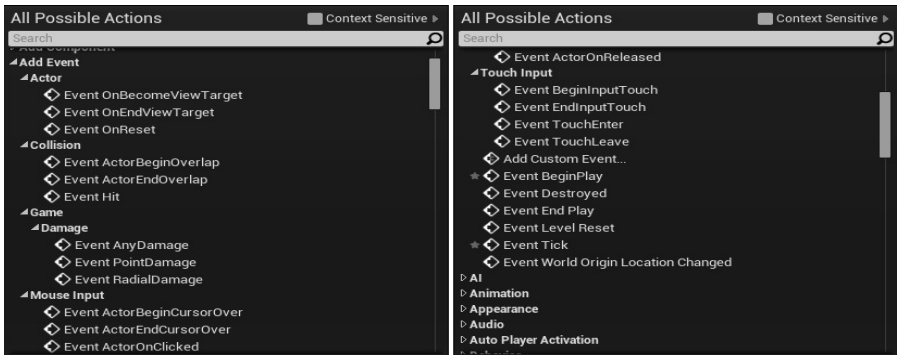

Рис 1.7. Доступні події

Джерело: **[27]**

Видно що є велика наявність подій, як поділені на категорії, тобто присутні ігрові події, події на колізію, на пошкодження, навіть є події для взаємодії із мишкою та її клавішами, існують ще на дотик, анімації та аудіо.

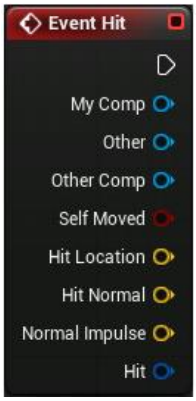

Рис 1.8. Подія EventHit(Удар)

Джерело: **[28]**

РОЗДІЛ 2
ПРИКЛАДНА РОЗРОБКА ІНТЕР'ЄРУ ПЕРШОГО ТА ДРУГОГО ПОВЕРХІВ НОВОГО КОРПУСА НАУОА) В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ BLENDER

2.1. Використання Jira для управління та плануванням проектами

На початку роботи в Jira Software ми розподілили обов'язки, тобто для кожного учасника команди створили спринти, і вже у них були додані відповідні завдання для виконання (Tasks). У кожного спринта є свій визначений час для виконання та поставлений статус, у якій фазі розробки завдання.

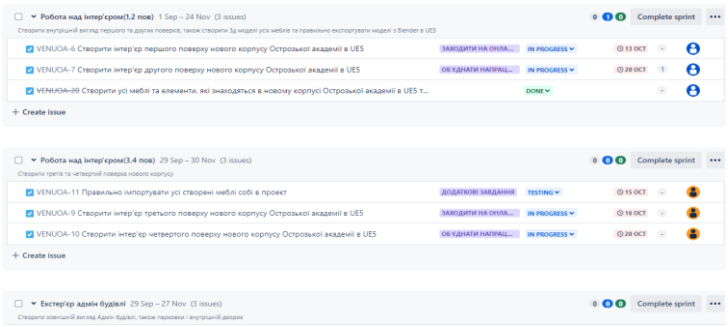

Рис 2.1. Показ наявних спринтів у вкладці «Backlog» **[29]**

Джерело: створено автором

Після натиснення на конкретне завдання, відкривається вікно з детальною інформацієї та описом

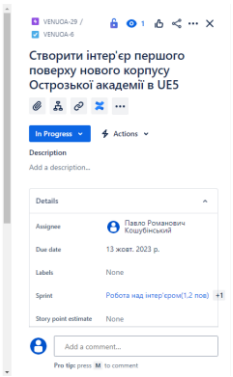

Рис 2.2. Вікно конкретного завдання

Джерело: створено автором

На дощі відображається в стовпцях набір задач. Кожен стовпець представляє етап виконання роботи командою.

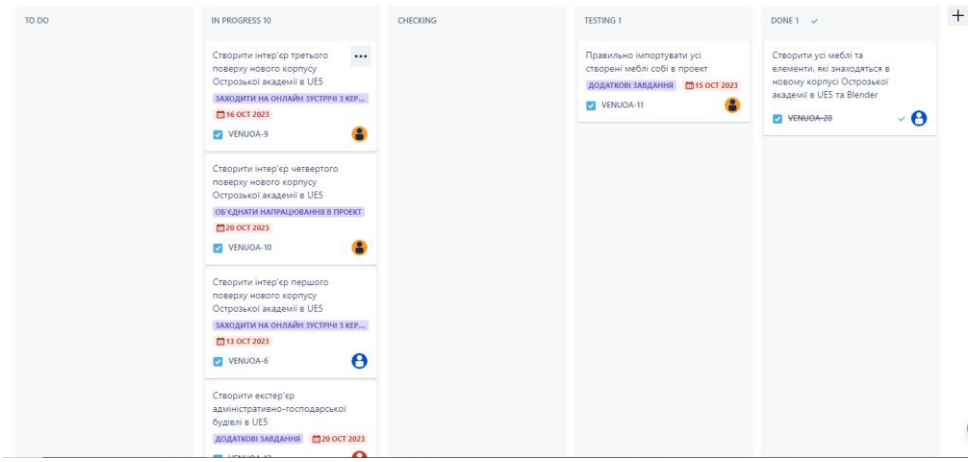

Рис 2.3. Демонстрація дошки Scrum

Джерело: створено автором

Як видно на фото, ми створили 5 стовпців, кожний з яких це етапи проходження кожного завдання та перевірки.

Перший стовбець – це TO DO**,** тобто що потрібно зробити у майбутньому.

Другий стовбець – це IN PROGRESS, тобто завдання у процесі виконання.

Третій стовбець – це CHECKING, тобто перевірка завдань після виконання.

Четвертий стовбець – це TESTING, тобто тестування усіх механік для знаходження усіх багів, для подальшого покращення.

П’ятий стовбець – це DONE, тобто завдання, які уже пройшли усі минулі етапи та повністю виконані.

У Jira Software робочим процесом називається шлях, який проходять завдання від створення до завершення. Кожен робочий процес складається з набору статусів та переходів між ними, які робить завдання протягом життєвого циклу. Зазвичай робочий процес є відображенням процесів у створенні.

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

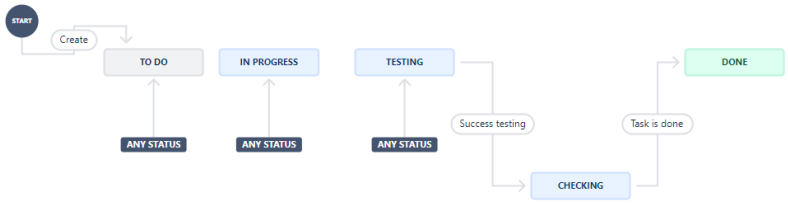

Рис 2.4. Налаштований робочий процес під наші потреби

Джерело: створено автором

Тут на скріншоті наглядно видно, які процеси проходить конкретне завдання, щоб отримати статус «DONE». Наприклад, завдання не може напряму перейти з статуса «TESTING» до статуса «DONE», тому що йому перше треба перейти у «CHECKING», а потім вже в статус «DONE»

Також для кожного випуску є свої типи:

Epic – Epics відстежує 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 завдання.

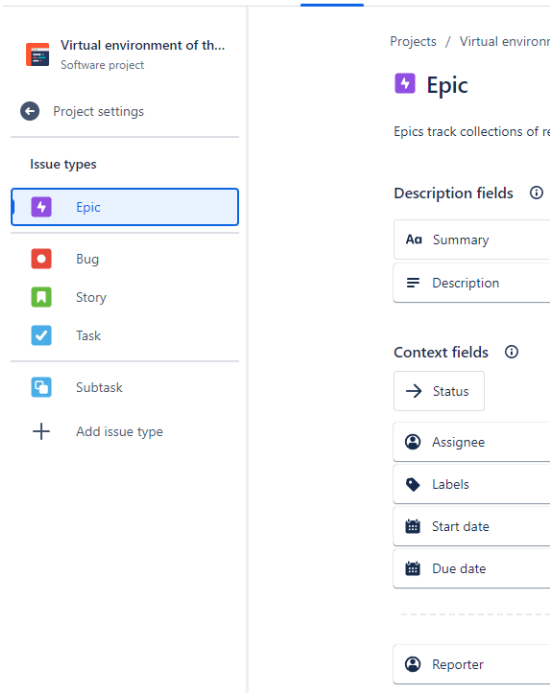

Рис 2.5. Демонстрація наявних типів

Джерело: створено автором

2.2. Опис технологічного стеку

Нами було використано програми:

1. Blender для моделювання усіх основних елементів декору;
2. Unreal Engine 5 для реалізації вільної ходьби в віртуальному просторі;
3. Adobe Photoshop було використано для підведення усіх планів для оптимальних розмірів.

В програмі Blender також використовували аддони:

Bool Tool- дає можливість зручно та бистро вирізати отвори в об’єктах;

BlenderKit- це велика платформа з безкоштовними та платними матеріалами, моделями, HDRI картами;

Також аддони для зображень, щоб зручно було перенести план у наш проект.

В Unreal Engine 5 також розробляли основні елементи інтерактивності, для взаємодії ігрового персонажа з іншими предметами, тобто взаємодія із дверима та вікнами, можливість піднімати елементи на сцені.

Також в UE5 моделювали неважкі об’єкти, які не вимагають складних технологій та використовували стандартні об’єкти, які вже адаптовані під взаємодію з персонажем для інтерактивності.

2.3. Збір основної інформації для виконання завдання

1. Знайти детальні схеми першого та другого поверхів нового корпусу;

Отримали в провідного інженера усі плани академії та взяли дозвіл (Рис.3) на фотографування корпусів.

2. Детальне фотографування аудиторій та коридорів для моделювання усіх елементів інтер’єру та розробки матеріалів.

Спочатку ми зробили багато фото та відео внутрішньої та зовнішньої частини нового корпусу НаУОА.

Далі було створено проект на своєму комп’ютері, де зберігалися усі зібрані дані та процес розробки.

Усі фото робили самостійно і також використовували знімки з дрона, які були у вільному доступі, щоб краще було видно різні деталі.

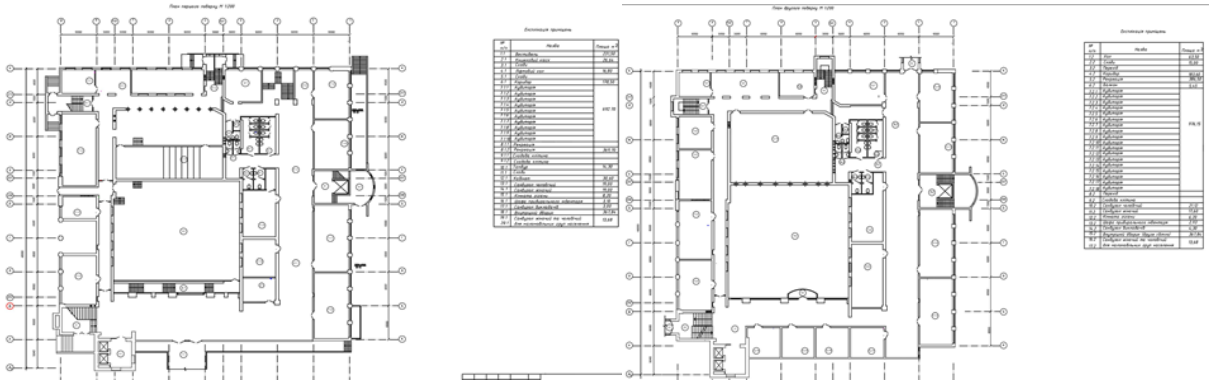

Рис 2.6. План першого та другого поверхів

Джерело: Національний університет «Острозька академія»

Детальні фотографії коридорів та аудиторій першого поверху нового корпусу старалися робити самі найважливіші та щоб було зрозуміло усі маленькі деталі на фото.

Рис 2.7. Фото коридорів та аудиторій нового корпусу

Джерело: створено автором

2.4. Створення 3D моделей в Blender

За допомогою планів, виведення конструктивних елементів у 3D вигляді

Хід роботи:

1. Створення стін у 3D для першого поверху;
2. Створення стін у 3D для другого поверху.

Для початку я добавив фотографію плану першого поверху у свій проєкт і по ньому почав зводити стіни та розмітку по розмірам у 3D форматі використовуючи звичайні куби, далі перейшли в EditMode і там вже витягували стіни як потрібно та модифікатор BoolTool для вирізання отворів як показано на фото (Рис.2.6). Так само зробив і з другим поверхом.

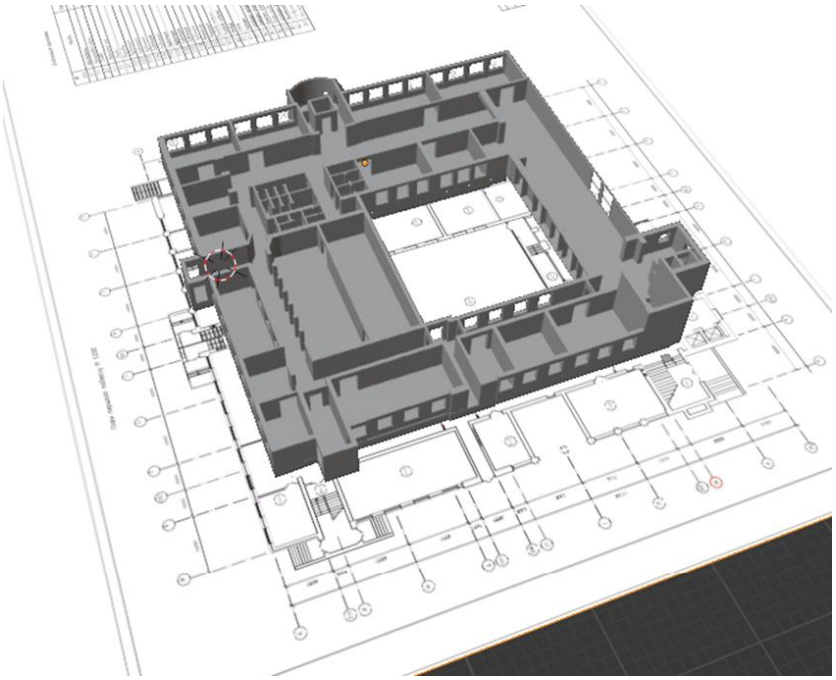

Рис 2.8. Перший поверх нового корпусу зведений у Blender

Джерело: створено автором

Деталізація усіх елементів інтер’єру по наявних фото.

Хід роботи:

1. Вибір кольору стін, підлоги, дверей, вибір меблів та декору;
2. Вибір світла (лампи, стеля і тд.);
3. Постановка дрібних деталь (вазони, коври і тд.)

Спочатку для вибору кольорів було використано ColorPicker, за допомогою нього можна дізнатися приблизний колір по фото.

Далі були зроблені ламки, за допомогою параметра “Emission” та звичайного куба

Дрібні деталі моделювали у окремому проекті за допомогою модифікаторів “Mirror, Array, Subdivision Surface”, за допомогою них економиться багато часу та зручніше.

Рис 2.9. Колаж інтер’єру у Blender

Джерело: створено автором

Моделювання усіх елементів інтер’єру.

Хід роботи:

1. Створення усіх меблів, тобто: стільці в аудиторії, стільці в комп’ютерні аудиторії, парти двох видів, комп’ютерні миші, клавіатури, монітори, системні блоки, дошка, батарея, вікна, двері, усі елементи в туалети, стеля, підвіконня;
2. Створення матеріалів до меблів, стін, підлоги та стелі;
3. Створення сходової клітини та перил.

Наприклад, розробка стільця з комп’ютерної аудиторії велася у три етапи:

1. Створення ніжок використовуючи в “EditMode” параметр “Extrude” та модифікатор “Mirror”;
2. Створення ручок та м’якої частини використовуючи розтягування об’єкта “Plane” та циліндрів, і також для згладжування кутів використовували модифікатор “Bevel”;
3. Підбір відповідних матеріалів на всі частини.

Рис 2.10. Меблі без матеріалів та текстур розроблені у Blender

Джерело: створено автором

Нанести усі матеріали та текстури на меблі, стіни, підлогу та стелю

Хід роботи:

1. Нанесення матеріалів на всі меблі;
2. Нанесення текстур та матеріалів на стіни та підлогу;
3. Розробка матеріалу на стелю через нодову систему.

Створення матеріалів розроблялося через нодову систему, яка добре реалізована в Blender. На чорний стілець з аудиторій було розроблено два матеріали, один на м'яку частину, а інший на метал.

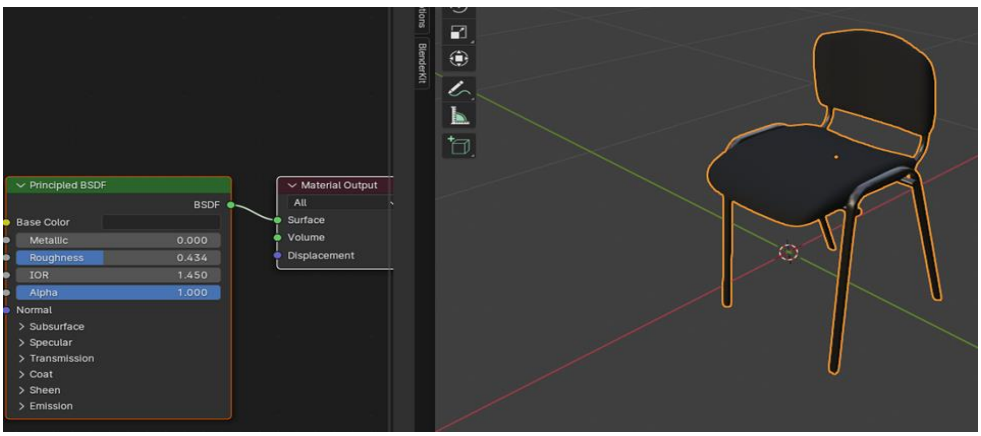

Рис. 2.11. Приклад створення м’якої частини стільця в Blender

Джерело: створено автором

Для його створення потрібно паметр “Metallic” викрутити на значення 0, це надасть об'єктові ефект сивості, і трішки “Roughness” для регулювання шороховатості. Далі просто наношу раніше розроблені матеріали по окремості, щоб вони коректно застосовувалися.

У Blender ми використовували нанесення текстур та матеріалів на стіни та підлогу наступними методами:

Нанесення текстур на об'єкт: створення матеріалу: У вкладці "Shader Editor" створили матеріал для стіни або підлоги, натиснувши кнопку "New" для створення нового матеріалу.

Додавання текстури: додали текстуру до матеріалу, використовуючи вузли. У вкладці "Shader Editor" використовуються вузли, такі як "Texture Coordinate", "Image Texture" та "Mapping", щоб додати та налаштувати текстуру.

Налаштування параметрів текстури: після додавання текстури почали редагувати її параметри, такі як масштабування, зміщення, обертання та інші налаштування за допомогою вузлів "Mapping" або шляхом редагування самої текстури через вузли.

Нанесення текстур на поверхню: редагування UV-координат: Вибрали об'єкт, на якому будемо наносити текстури, перейшли у режим редагування (Edit Mode), вибрали UV-редактор та розмістили точки у редакторі так, щоб вони відповідали тому, як текстура буде наноситися на об'єкт.

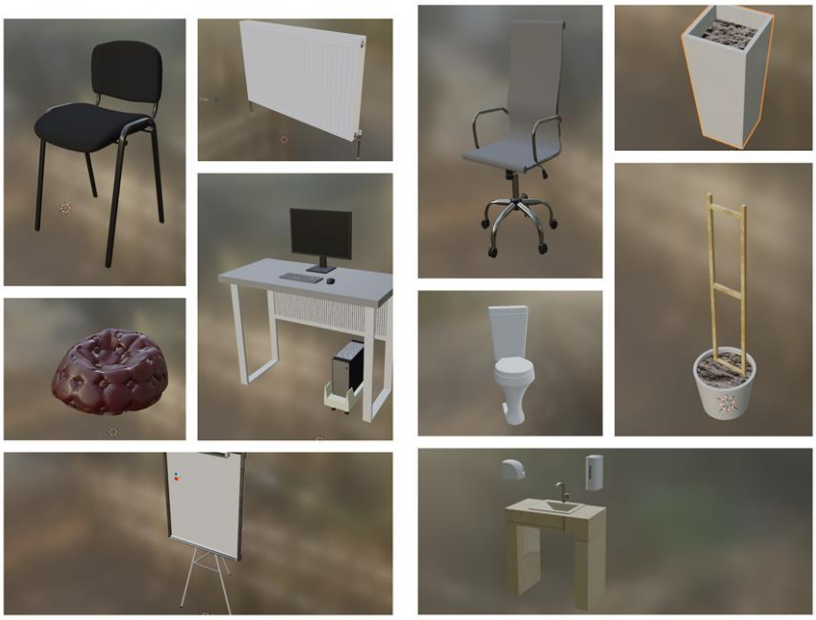

Рис 2.12. Меблі з матеріалами та текстурами нового корпусу розроблені у Blender

Джерело: створено автором

Стіл у комп’ютерних класах, там де нижня решітка, розробляли через матеріали, а не вирізалося фізично, тому що це дуже нагружало систему.

Кран було створено моделюванням “Subdivision”, ми доволі часто користувалися цим методом, коли ним робиш, моделі виходять доволі згладжені та звичні.

Пуфік було створено за допомогою анімації, тобто спочатку створив звичайний куб та сферу, потім кубові задав фізику тканини, а сфері задав колізію, наступним кроком, сферу розмістив вище куба та за допомогою анімації в Blender, сфера почала падати донизу та зробила випадкову форму у кубові.

Саме більше в мене часу пішло на матеріал стелі, через те що я робив її через нодову систему, а не через текстуру.

Було використано багато різних нодів, наприклад “Texture Coordinate” у поєднанні з “Mapping” для позиціонування та масштабування текстури

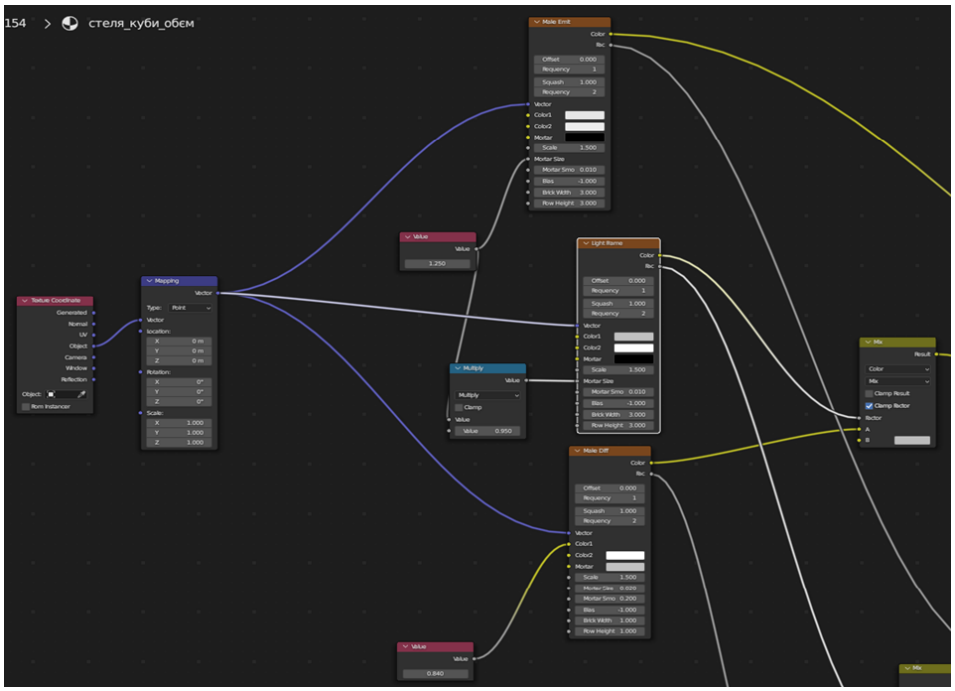

Рис 2.13. Матеріал стелі розроблений через нодову систему у Blender

Джерело: створено автором

Використовували ось такі елементи:

Multiply: використовується для множення значень між собою у вузлах Math або Color (Рис. 2.11). Це дозволяє нам змішувати, модифікувати або налаштовувати значення для досягнення певних ефектів у матеріалі.

Musgrave Texture: це одна з багатьох вбудованих шумових текстур, доступних 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:

Dimensions (Розміри): визначає масштаб шуму (Рис. 2.8). Чим більше значення, тим більші будуть деталі текстури.

Lacunarity (Лакунарність): коефіцієнт, що визначає "нерівномірність" текстури. Змінюючи цей параметр, ви можете контролювати рівномірність чи нерівномірність текстури.

Octaves (Октави): визначає кількість октав шуму, що додаються разом для створення текстури. Більше октав може призвести до більшої деталізації текстури.

Strength (Сила): контролює інтенсивність текстури.

Текстура Musgrave може бути використана для створення різних ефектів у матеріалах, таких як структуровані поверхні. Її можна змішувати з іншими текстурами, налаштовувати параметри.

Bump: відноситься до техніки, що використовується для створення візуального ефекту випуклості або виступів на поверхні об'єкта, без зміни фактичної геометрії моделі. Це досягається за допомогою карт Bump (Рис. 2.8) або Normal.

Рис 2.14. Демонстрація результату створення матеріалу стелі нового корпусу у Blender

Джерело: створено автором

Правильно імпортувати всі меблі та матеріали до основного файлу де моделювалися два поверхів.

Хід роботи:

1. Імпортувати меблі та інші елементи у основний (інший) файл з розширенням .blend;

2. Імпортування матеріалів у основний файл.
3. Розстановка усієї меблі по аудиторіях

Для початку я перемістив меблі та матеріали з одного проекту, за допомогою комбінації клавіш CTRL+C в файл з виведеними стінами по планах за допомогої клавіш CTRL+V. Потім понаносив всі матеріали на стіни, підлогу та все інше.

Матеріали також перенеслися без проблем в основний проект, тому що вони були зразу нанесені на меблі.

Підлогу ми робили за допомогою текстури, наклавши її, далі редагували UV-розгортку для приведення нормальний розмірів.

Перший поверх

Рис 2.15. Рендери першого поверху нового корпусу у Blender

Джерело: створено автором

Використовувалося світло з створених ламп за допомогою матеріалу і також налаштовувалося глобальне світло на проект.

Жалюзі були зроблені за допомогою накладанням текстури на Plane (Рис. 2.11) та розтягував за допомогою UV розгортки

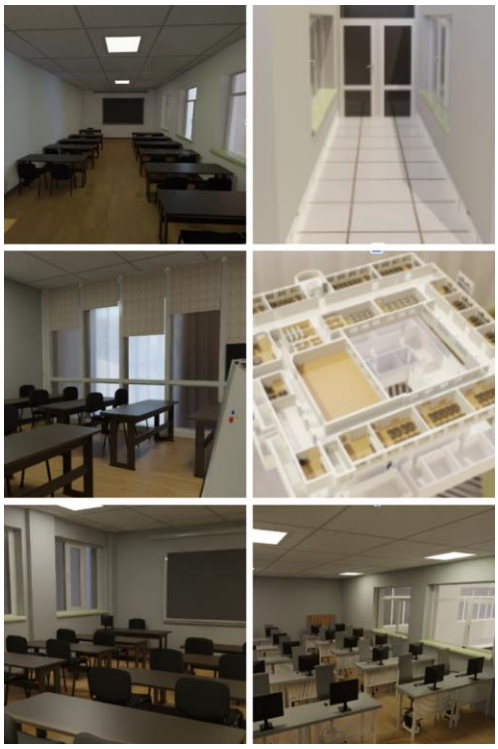

Рис 2.16. Другий поверх нового корпусу створений у Blender

Джерело: створено автором

2.5. Інтеграція 3D моделей в Unreal Engine

Через певні особливості інтеграції 3D моделей з Blender в Unreal Engine 5 і для покращення оптимізації, спільним-колективним рішенням було прийнято рішення, частину елементів 3D моделей переносити з середовища Blender, так як у нас вже було багато пророблено роботи там, а іншу частину робити в Unreal Engine 5, використовувати вбудовані інструменти для розробки. Це все було зроблено для найоптимальнішого покращення оптимізації.

Це розумне рішення, яке дозволяє нам ефективно поєднувати переваги обох середовищ для оптимізації та поліпшення якості проекту.

Це також дозволить оптимізувати робочий процес: використання вбудованих інструментів Unreal Engine 5 може спростити і прискорити розробку окремих елементів, особливо якщо вони пов'язані з функціями або властивостями саме цієї платформи. Водночас, вже готові елементи з Blender

можуть бути використані без додаткового перероблення, зберігаючи їх якість та характеристики.

Це також може позитивно вплинути на процес розробки, забезпечуючи гнучкість і можливість пристосування до змін у проекті, а також економію часу. Правильний розподіл завдань між двома середовищами може сприяти збереженню часу та ресурсів, а також полегшити співпрацю між різними частинами команди, які спеціалізуються на певних аспектах розробки.

Важливо також враховувати синхронізацію та збереження цілісності проекту при переміщенні та редагуванні елементів між цими двома середовищами. Це може потребувати уважного планування, документації та стандартизації процесів, але правильно налаштований потік роботи допоможе забезпечити ефективну і спільну роботу нашої команди.

За допомогою планів, виведення конструктивних елементів у ігровому рушії Unreal Engine 5

Хід роботи:

1. Створення стін, перестінків, стель, підлоги у Unreal Engine 5 для першого поверху;

2. Створення стін, перестінків, стель, підлоги у Unreal Engine 5 для другого поверху.

Спочатку я добавив фотографію плану першого поверху у свій проєкт і по ньому почав зводити стіни, використовуючи вбудовані стіни, розміром 400x400, підлогу за допомогою вбудованої Floor 400x200 в Unreal Engine 5 та інші конструктивні елементи по розмірам.

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

Щоб все точно спозиціонувати , розміщував усі об’єкти, за допомогою параметру “Location” по вісі X,Y,Z

Матеріали брав з платформи Quixel Bridge, далі наносив їх на об’єкт, та за допомогою інструмента “UV Project” вирівнював та розтягував матеріал по потрібним масштабам.

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.

Аналогічні дії були розроблені і з другим поверхом (Рис. 2.12).

Рис 2.17. Результат виведення двох поверхів у Unreal Engine 5

Джерело: створено автором

Деталізація усіх елементів інтер’єру по раніше зібраних фото та відео.

Хід роботи:

1. Імпортування усіх меблів(парти, монітори, вікна, двері, крісла, дошки) з Blender;
2. Вибір кольору стін, підлоги, дверей, вибір меблів та декору;
3. Вибір світла (лампи, стеля і тд.);
4. Постановка дрібних деталь (вазони, коври і тд.)
5. Розстановка усіх меблів по аудиторіях

Для імпортування меблів з Blender, зберігали спочатку моделі в форматі .fbx, це для того, щоб якісно перенісся об’єкт в Unreal Engine 5. Також ми ще використовували експортування в формат .obj, нам він більше був до вподоби.

Кольора стін ми підбирали схожі по фото та відео, так само двері, підлога, декор. Розстановкою меблів також робилася приблизно по фотографіям, які нас вдалося зібрати.

Світло використовували не стандартне, а з використанням підбору різних HDRI карт, так був набагато кращий ефект сонячних лучів та якісніше освітлення, також була змога регулювання кута наклону сонця, таким чином робили день або вечір. Всередині додавали додаткове освітлення, за допомогою інструмента “RectLight”. Великим плюсом цього інструмента є те, що він прозорий при рендері карти.

Вже після імпорту усіх дрібних деталей, була розстановка горщиків, вазонів і тд. і також з Quixel Bridge додав невибагливу рослинність.

Quixel Bridge є можливість скачувати разом з Unreal Engine 5, і він йде як аддон до рушія.

При скачуванні матеріалів, їх зразу можна буде перетягнути в вікно Unreal Engine 5 і вільно нанести на об’єкт.

Тисячі 3D-ресурсів, плиткових поверхонь, недоліків, рослинності, атласів і наклейок з усього світу є у цій програмі. Присутні фотореалістичні або стилізовані проекти з найвищою якістю 3D-контенту в галузі. Новий вміст Megascans записується та додається до бібліотеки щодня.

Через Bridge можна отримати доступ до індивідуальних MetaHumans, створених у MetaHuman Creator, а також до понад 50 готових MetaHumans. Маючи доступні як UAssets, так і вихідні ресурси, можна використовувати Bridge для завантаження та експорту своїх цифрових людей для безпосереднього використання в Unreal Engine або подальшого редагування в Blender.

Результати роботи першого поверху

Рис 2.18. Перший поверх розроблений у Unreal Engine 5

Джерело: створено автором

На цьому етапі саме найдовший термін проводиться робота над поставленням світла та створення стелі. Також затратним по часовим рамках виявилося виведення жовтих ліній та квадратів, які розташовані на підлозі. Для покращення світла, ми використовували додаткові джерела освітлення та вмонтовані фонарі в стелі.

Рис 2.19. Результат другого поверху нового корпусу розроблені у Unreal Engine

Джерело: створено автором

Не мало важливим елементом є розстановка усіх меблів через комбінацію клавіш Alt+D, щоб оптимізувати проект, тому що, якщо робити просто копії об’єкта, то дуже заб’ється оперативна пам’ять.

Один з важливих результатів- це розробляти 206 аудиторію, і ще мені потрібно було розробляти в Blender нові парти та крісла, так як вони відрізняли від усіх інших меблів.

Було зроблено картини максимально приближені до реальних, за допомогою нанесення текстури на Plane. Нанесення відбувалося також за допомогою нодової системи, використовували “Texture Sample” і вже у нього завантажувалась фотографія у форматі .png та проводилося налаштування колірної гами та прозорість, далі трішки поправлена UV розгортка для покращення якості самого матеріалу.

Вікна моделювалися у Blender, тому що у цьому середовищі дуже зручно розробляти моделі, кожне вікно було поділене на три окремі частини, тобто сам каркас із склом, частина яка відкривається для провітрювання та ручка, це все було зроблено, щоб зручніше робити інтерактивну взаємодію із ними, відкривати або закривати.

Рис 2.20. Результат візуалізації аудиторії №206 нового корпусу розроблена у Unreal Engine 5

Джерело: створено автором

У цій аудиторії вікна розроблялися за допомогою двох частин.

Перша частина- це пластикова рамка;

Друга частина- це скло. Матеріал скла був взятий стандартний, який уже йде при створенні проекту.

Підлога була зроблена максимально приближена до реальності, сходини були пораховані та відтворені, також реалізовано сходи, там де інший вихід з аудиторії.

Також були реалізовані м’які крісла та столики, максимально приближені до реальності.

Парти розроблялися трішки інші, переробляли колір та трішки форму.

2.6. Інтерактивна частина створення НаУОА (інтер'єр першого та другого поверхів нового корпусу)

На початку реалізації звуків, ми знайшли безкоштовні звукови для поверхонь: (метал, трава, дерево) та імпортували їх у наший робочий проект. Далі було створено Cue. Туди були додані усі наші звуки.

У Unreal Engine Sound Cue — це аудіоресурс, який інкапсулює складні завдання звукового дизайну в графі вузлів. Sound Cues надають звуковим дизайнерам свободу динамічно змінювати частини дизайну звукового ефекту шляхом упорядкування та модифікації звукових вузлів для створення складного та цікавого аудіовиходу.

Потім потрібно було створити також три фізичних матеріали(метал, дерево, трава) і додали їх в Physical Surfaces**.**

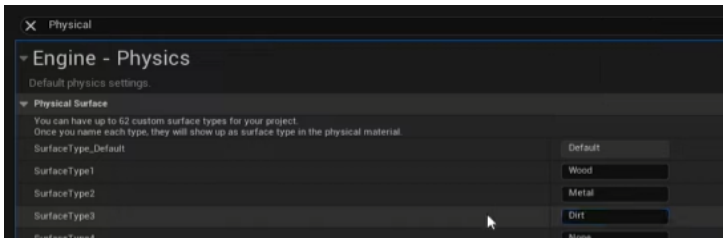

Рис 2.21. Додавання назви для Physical Surfaces**[30]**

Джерело: створено автором

Додавання раніше створеної властивості для фізичного матеріала

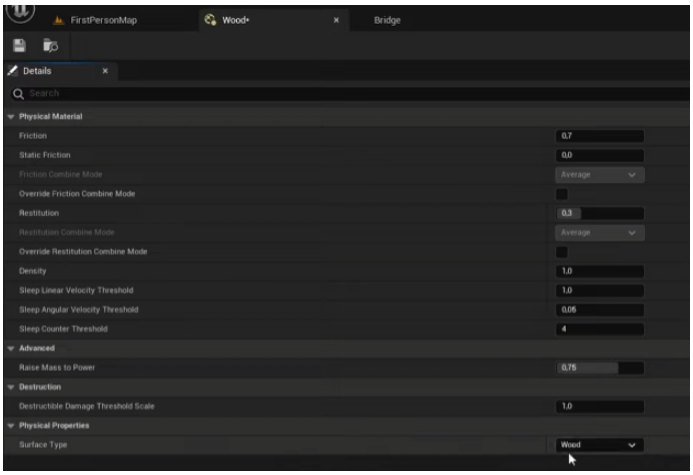

Рис 2.22. Змінення властивості Surface Type на потрібну

Джерело: створено автором

Для інших матеріалів проводимо аналогічні операції. Після цих дій, було створено функцію Trace Footsteps, де вже ця функцію викликається в блюпрінті нашого основного персонажа.

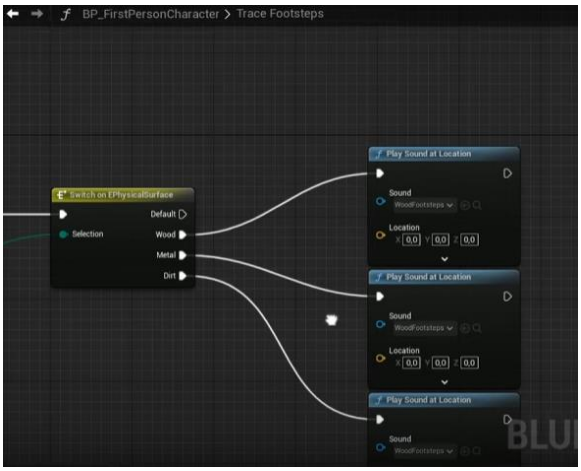

Рис 2.23. Об’єднання нодів для наших використаних звуків

Джерело: створено автором

Наступний крок, це додавання наших фізичних матеріалів для наших основних матеріалів, які використовуються на потрібних нам поверхнях

Рис 2.24. Застосування фізичних матеріалів

Джерело: створено автором

Завершальний крок створення звуків, це налаштування радіуса та дистанції дії наших використаних звуків на поверхнях

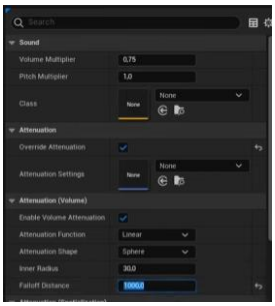

Рис 2.25. Встановлення потрібних налаштувань на радіус на дистанцію

Джерело: створено автором

Основна логіка роботи полягає в тому, що при натисканні клавіш для руху персонажа (WASD) відтворюється звук, який прив'язаний за допомогою фізичного матеріала до відповідної поверхні, і цей звук зациклений в звуковій доріжці.

Почавши розробляти цю механіку, розпочався пошук документації та моніторинг, в результаті було використано спеціальний експериментальний плагін WebBrowser.

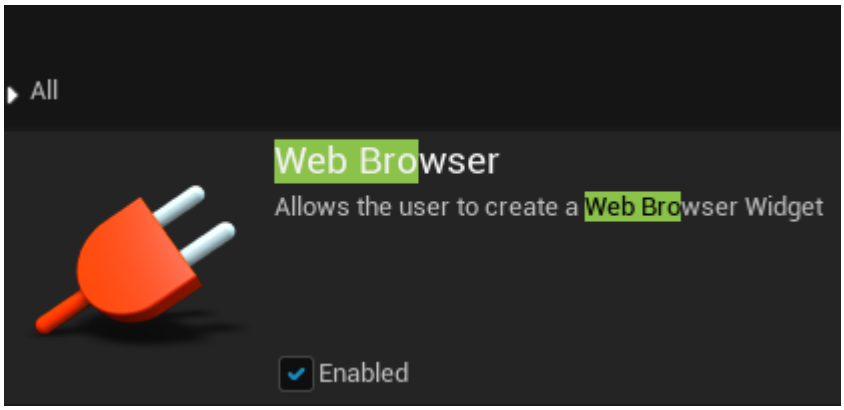

Рис 2.26. Вигляд плагіна Web Browser

Джерело: створено автором

Рушій unreal надає плагін для веб-браузера, який дозволяє створювати інтерфейси за допомогою браузера. Це дозволяє користувачам отримувати доступ до віддалених або локальних URL-адрес, подібно до звичайного Інтернет-браузера; таким чином полегшуючи дизайн інтерфейсу, ніби це розробка веб-сайту. Основна технологія, що використовується, базується на Chromium Embedded Framework (CEF) версії 59 – інструменті з відкритим вихідним кодом для швидкого вбудовування браузерів у різні додатки та програми сторонніх розробників.

Наступний крок, це створення блюпринта та додавання віджета WebBrowser, тобто за допомогою нього, можна ввівши відповідне посилання, зайти на любу інтернет сторінку та взаємодіяти з нею.

Далі для симуляцію робочого стола комп’ютера, операційної системи Windows 10, було створено задній фон, додані кнопки для подальшої взаємодії з ними, за допомогою подій, також використані зображення та звичайні лінії, для відокремлення різного контенту між собою. Функціонал виконаний наступний:

1. Коли персонаж підходить до монітора, комп’ютер автоматично вмикається і відображається робочий стіл;

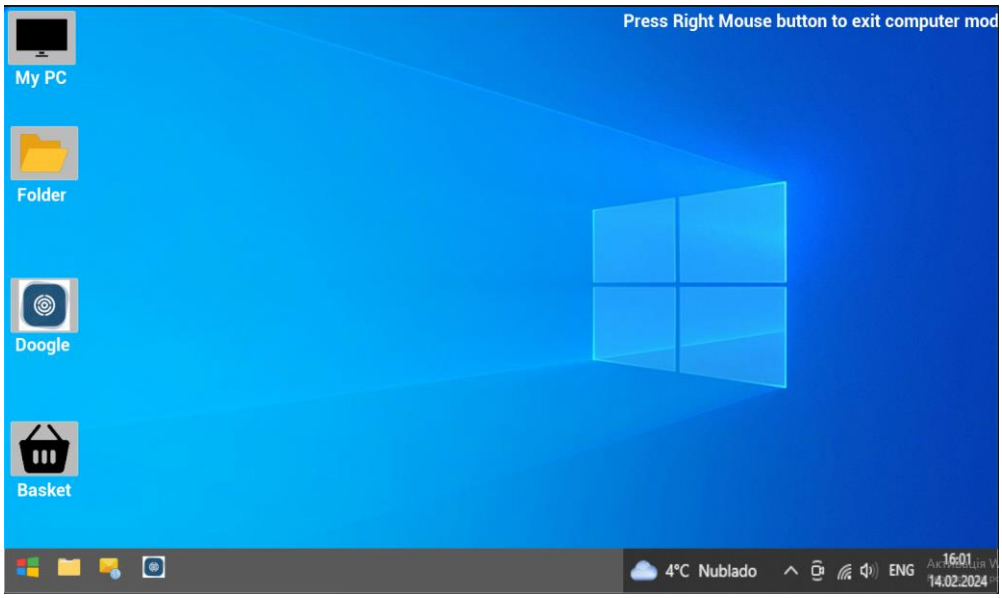

Рис 2.27. Вигляд робочого стола

Джерело: створено автором

2. На робочому столі при натисканні на іконку “Doogle Browser”, відбувається перехід на сторінку з симуляцією браузера, де можна задати будь-яке посилання для переходу на сайт, або на панелі обрати вже існуючі, які показані відповідними іконками.

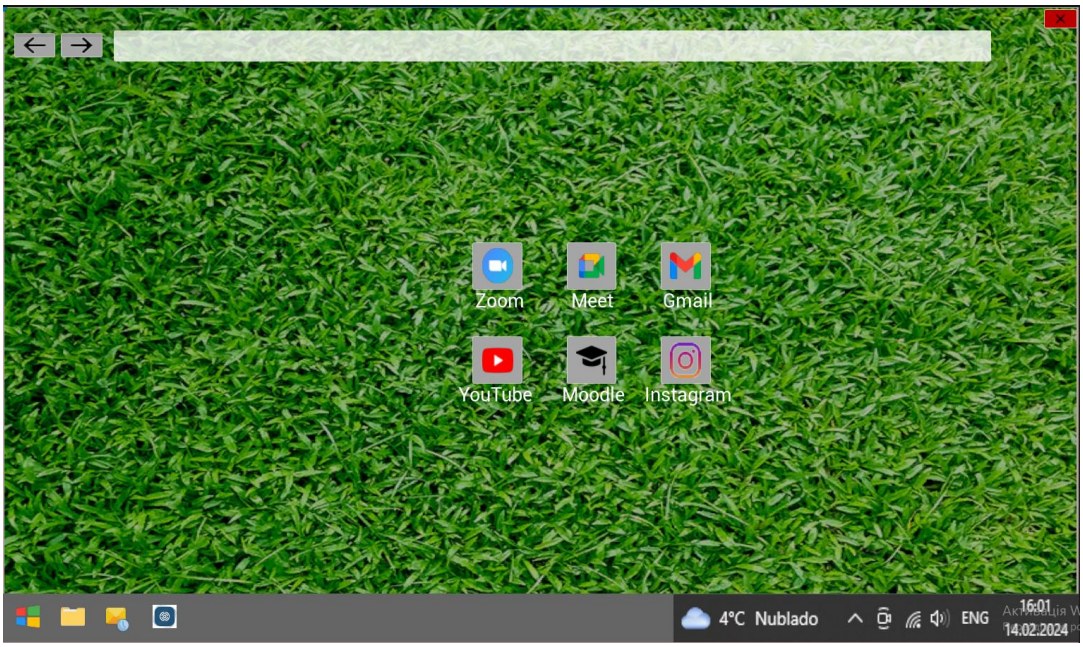

Рис 2.28. Вигляд симулятора браузера

Джерело: створено автором

3. Обравши іконку під назвою “Folder”, відбудеться перехід на сторінку, у якій наявна симуляція провідника файлів, які знаходяться на комп’ютері.

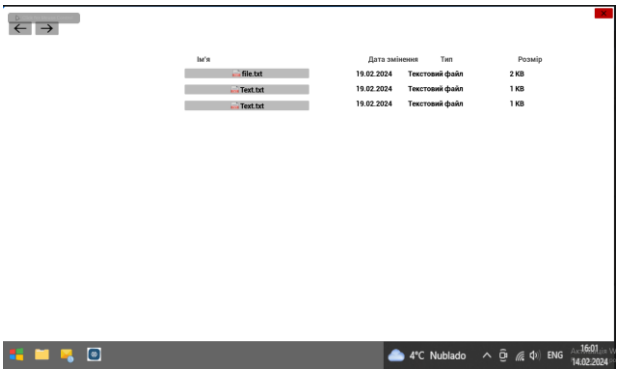

Рис 2.29. Віджет симуляції відтворення провідника Windows 10

Джерело: створено автором

Ці всі вікна інтерфейсу були створені за допомогою віджетів та їхніх стандартних елементах, і на кожну клавішу за допомогою використання Events Button, на кожну клавішу прив'язаний відповідний віджет для переходу.

Використовуючи різну інформацію для пошуків, також було вирішено робити взаємодію із дверима та вікнами за допомогою блюпринтів та використання віджетів для взаємодію з користувачем.

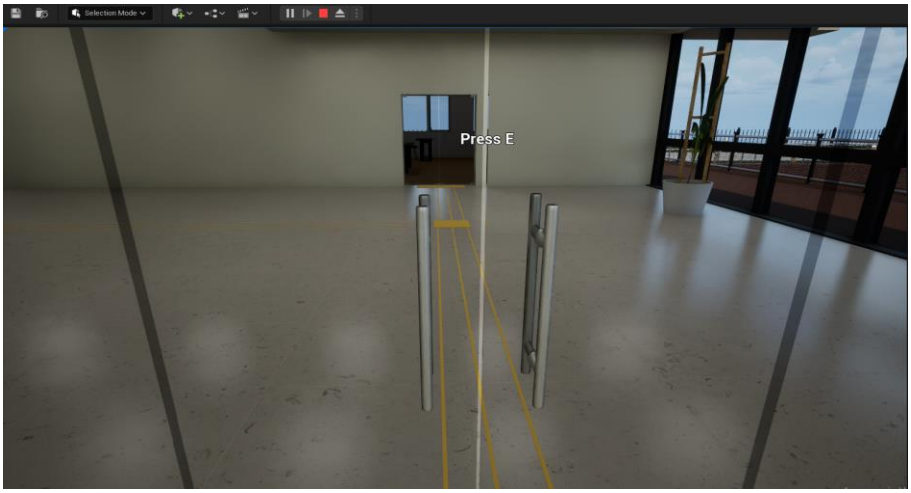

Рис 2.30. Показ роботи блюпринта

Джерело: створено автором

Основна логіка така, коли персонаж підходить до дверей, йому на екрані з’являється підказка, яку клавішу потрібно натиснути і тоді відкриваються двері і також закриваються при натисканні на клавішу, а основні двері, які знаходяться на головному вході в корпус, відкриваються автоматично, коли персонаж підходить до них.

Після відкривання, персонаж як проходить може закрити їх за собою і підказка зникне.

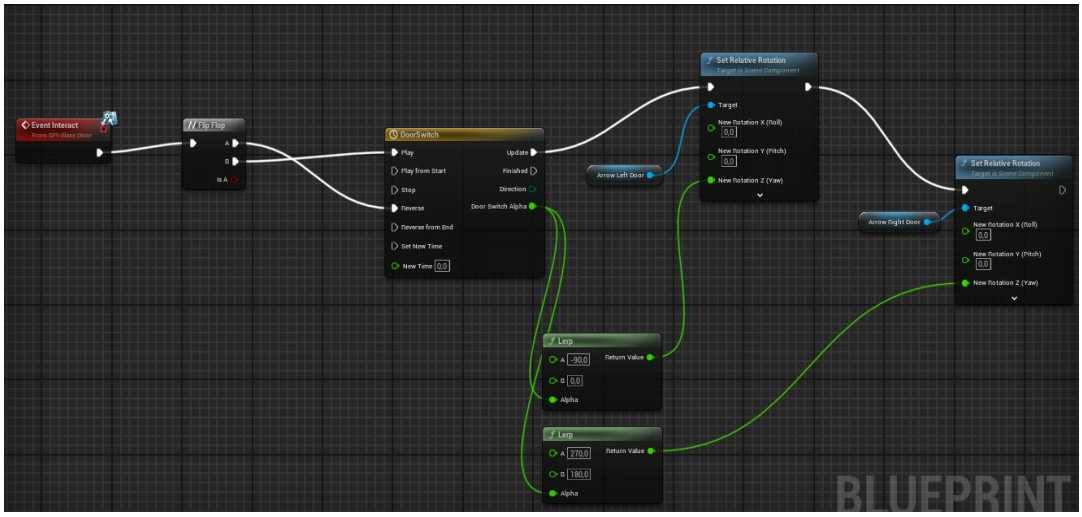

Рис 2.31. Фрагмент нодів, які використовуються у блюпринті для дверей і аналогічно для вікон

Джерело: створено автором

Розглянемо ноди, які використовуються

FlipFlop - під час кожного виконання вузла FlipFlop, він чергує вихід між контактами A та B. Статус того, який контакт зараз активний, можна визначити, перевіривши логічний параметр під назвою IsA. Якщо значення IsA має значення True, це означає, що поточний робочий штифт дорівнює A; тоді як False вказує на те, що використовується контакт B.

Timeline - вузли Timeline служать спеціальними вузлами, які полегшують швидке проектування та відтворення анімації на основі часу. Ці вузли можуть бути викликані подіями, плаваючими елементами, векторами або кольорами в ключових кадрах, розташованих уздовж шкали часу.

У редакторі Blueprint часові шкали можна безпосередньо змінити, подвійним клацанням миші на них у вкладці Graph або My Blueprint. Ці інструменти розроблено спеціально для керування такими простими завданнями, як зміна освітлення та відчинення дверей, а також внесення тимчасових коригувань акторів під час сцени. Подібно до послідовностей рівнів, вони пропонують такі значення, як плаваючі елементи, вектори та кольори, які інтерполюються між різними ключовими кадрами на часовій шкалі.

Set Relative Rotation - налаштування обертання компонента відносно його батьківського об’єкта.

Lerp - лінійно інтерполюючий, результат лежить між A і B, де альфа визначає пропорцію кожного використаного значення (A використовується повністю, коли альфа дорівнює 0, тоді як B використовується повністю при альфа-оцінці 1).

Робота цього блюпринта полягає в наступному, коли підходить персонаж до ліфтових дверей, вони автоматично відкриваються, коли персонаж трішки проходить вперед, на екрані з’являється курсор і користувач може обрати на який поверх потрібно переміститися, після натискання, ліфт рухається на відповідний поверх і двері автоматично відкриваються, після того як персонаж вийшов, двері також автоматично закриваються.

Рис 2.32. Показ роботи вибору номера поверху

Джерело: створено автором

Коли персонаж зайшов в ліфт і закрилися двері, курсором миші може обрати потрібний поверх і ліфт переміститься на поверх і тільки персонаж виходить з ліфта, курсор автоматично пропадає, і натиснувши ліву клавішу миші, персонаж далі може рухати камерою у різні сторони.

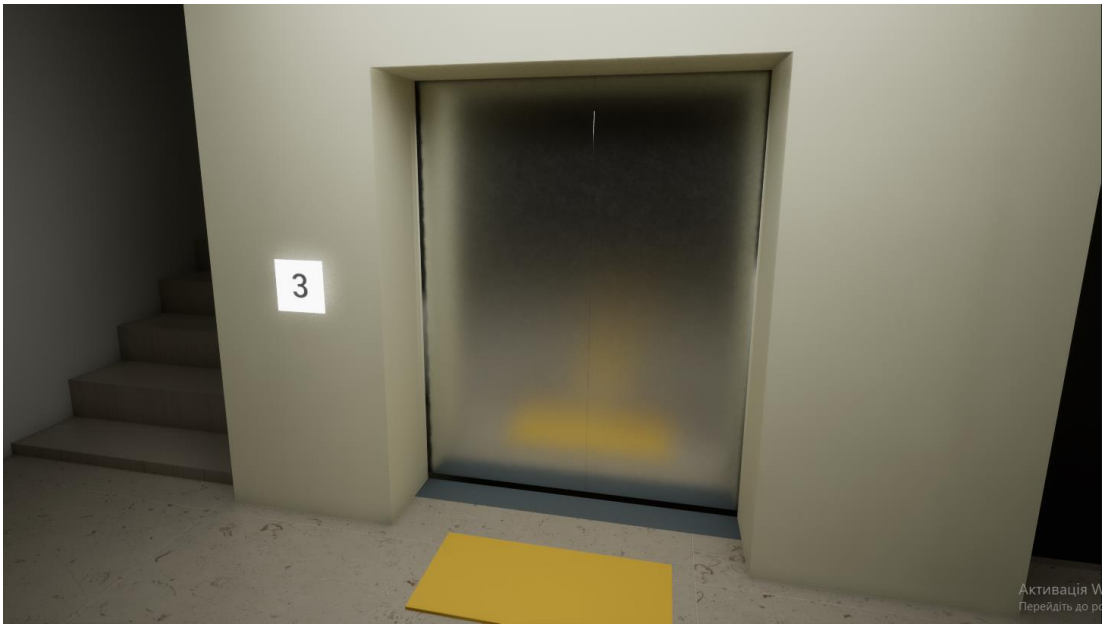

Рис 2.33. Показ роботи блюпринта

Джерело: створено автором

Тут ми бачимо, що ліфт спрацював і персонаж перемістився куди йому було потрібно, щоб не підніматися по сходах.

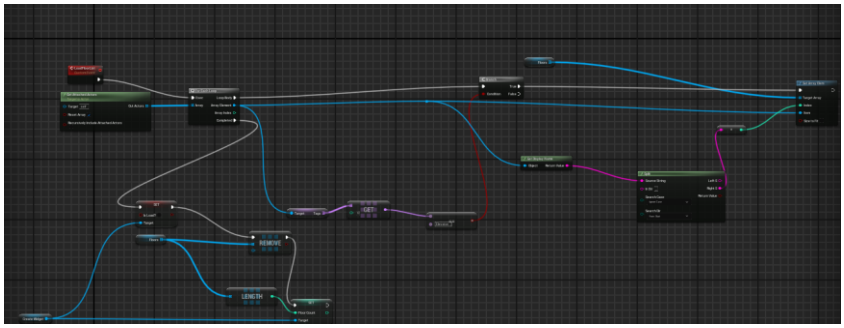

Рис 2.34. Фрагмент нодів, які використовуються у блюпринті

Джерело: створено автором

Тут показано виконання переміщення ліфта у вказані координати, які розміщені на BoxCollision у відповідних місцях для чотирьох поверхів, і прив'язані до відповідно своєї клавіші. які показано вище.

Також реалізовано вмикання та вимикання ламп в коридорах, туалетах та аудиторіях, тобто коли персонаж переміщується до вимикача, йому показується відповідна підказка, яку клавішу потрібно натиснути, і після цього, світло вимикається і якщо потрібно, вмикається.

Рис 2.35. Демонстрація підказок для користувача

Джерело: створено автором

Коли персонаж відходить від вимикача, вона автоматично приховується, це реалізовано використовуючи BoxCollision, тобто коли наш ігрок заходить в відповідну невидиму зону, спрацьовує подія, яка викликає відповідний віджет із потрібно підказкою для користувача, а коли виходить із неї, підказка зникає.

Ось ці підказки, що показані на фото (Рис. 2.30.) реалізовані за допомогою User Interface, використовуючи зображення, щоб користувачу було інтуїтивно зрозуміліше та текст для написання, яку клавішу потрібно використовувати.

Рис 2.36. Виконання вимкнення світла

Джерело: створено автором

Працює цей блюпринт наступним чином, коли світло ввімкнено, і як герой натискає ліву клавішу миші, відбувається заміна матеріалів на кожній лампі, та виключається додаткове підсвічування під назвою RectLight, і аналогічні дії відбуваються, коли персонаж знову натискає відповідну клавішу.

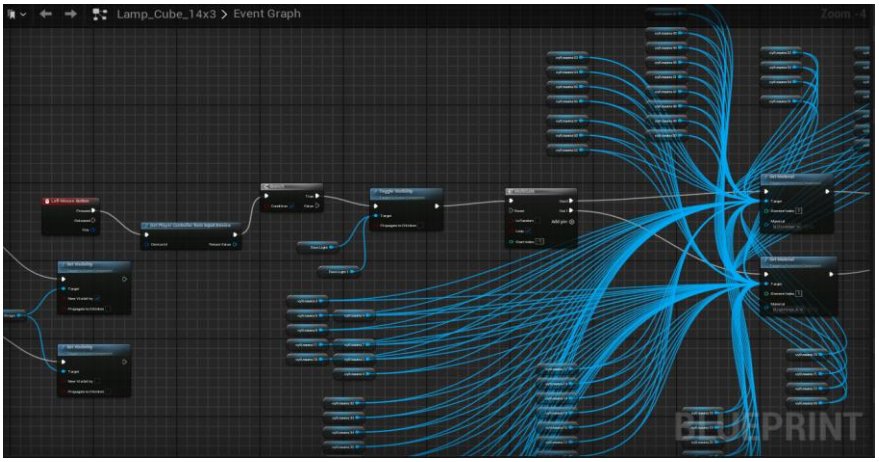

Рис 2.37. Фрагмент нодів, які використовуються в блюпринті

Джерело: створено автором

Було вирішено додати ще таку функцію, як камери безпеки, вони допомагають переглядати усю територію та спостерігати на в'їзді за шлагбаумом.

Коли ігрок підходить до великого монітора, що знаходиться в холі, він одразу запускається і при натисканні на клавішу K на клавіатурі, поле зору переходить у режим перегляду у монітор, та активується робота камер, які можна переключати за допомогою клавіші R на клавіатурі, якщо потрібно вийти у звичайний режим, потрібно знову ж натиснути клавішу K, і персонаж вийде з режиму перегляду камер та коли він відійде, монітор автоматично вимкнеться.

Рис 2.38. Вигляд роботи камер, коли персонаж підійшов

Джерело: створено автором

Як видно на зображенні (рис. 2.38), тільки персонаж заходить в режим камер, активовуються перша камера, яка має вид на головну площу та видно адмінбудівлю і трішки шлагбаум.

Рис 2.39. Вигляд з камери у 206 аудиторії

Джерело: створено автором

Як персонаж переключає камеру на клавішу R, з’являється вид з іншої камери, яка розставлена у 206 аудиторії, на другому поверсі, з якої можна дивитися на всю площу аудиторії.

Рис 2.40. Вихід із режиму камер за допомогої клавіши K на клавіатурі

Джерело: створено автором

Після того, як персонаж натиснув клавішу K, поле зору з основної камери персонажа відводиться назад, та деактивовується дія блюпринта камер безпеки і пероснаж дальше може вільно переміщувати по всьому корпусу академії.

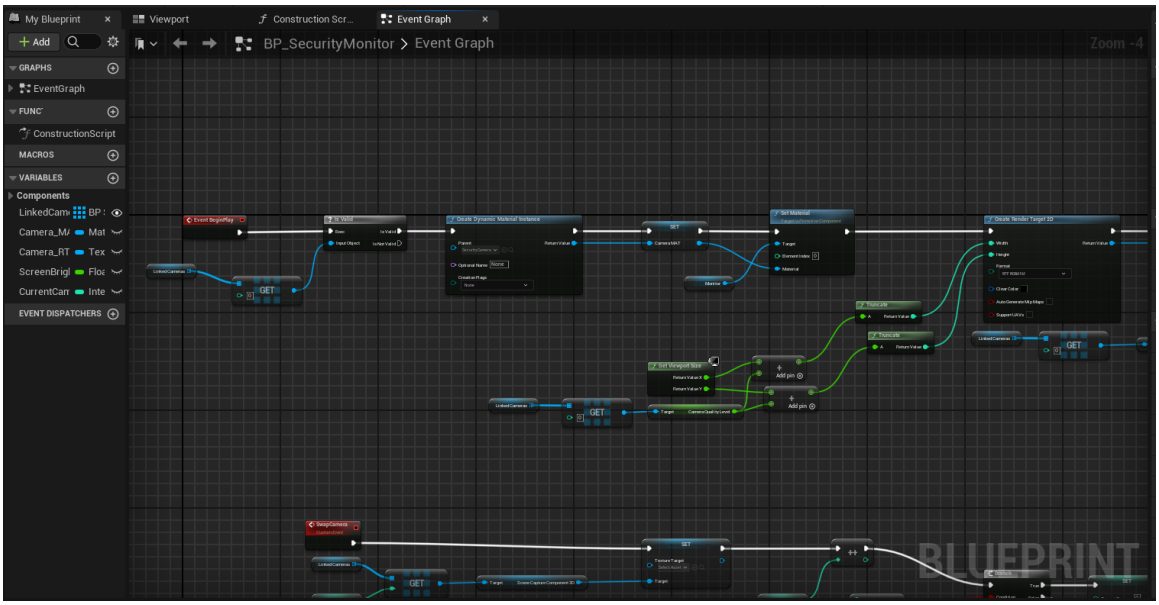

Рис 2.41. Фрагмент використання нодів у блюпринті камер

Джерело: створено автором

IsValid - якщо об’єкт придатний для використання, поверніть true. Не очікує на знищення та не нульовий

Create Dynamic Material Instance - для заданого індексу елемента створюється екземпляр динамічного матеріалу, необов’язково з наданого матеріалу.

Set Material - змінює матеріал, який наноситься на елемент сітки.

Create Render Target 2D - створює нову ціль візуалізації та встановлює її початкові розміри.

Get Viewport Size - помічник для визначення розміру полотна HUD цього контролера гравця, дає 0 за відсутності HUD.

ВИСНОВКИ

Для створення інтерактивного віртуального середовища першого та другого поверхів академії було проведено аналіз та зібрано інформацію. За допомогою програмного забезпечення Blender та Unreal Engine 5 було розроблено моделі стін, перестінків та меблів. Використані плани стали основою для створення моделей у Blender, включаючи вирізання вікон та дверей. Текстурування проводилося на основі реальних фотографій та матеріалів у Blender. Після цього всі деталі було імпортовано у середовище Unreal Engine 5 для подальшого доопрацювання.

Інтерактивність була реалізована за допомогою використання Blueprints та Widgets, розробляючи взаємодію з об'єктами, за основу були взяті моделі, які ми робили у середовищі Blender, і вже з ними комбінували різні механіки

Було зроблено відкривання/закривання дверей та вікон в аудиторіях, коридорах, туалетах, у холі, реалізували звуки на ходьбу персонажа по різним поверхням, також була відтворена механіка роботи комп’ютера, з можливістю заходити в браузер та моніторити різні сайти. Для зручного та бистрішого переміщення між поверхами корпуса, був зроблений ліфт, з можливістю вибору переміщення на будь-який поверх з інтуїтивно зрозумілим інтерфейсом.

На кінець роботи, реалізовані камери спостереження за територією та аудиторіями, які можна вільно переключати, а також було проведення детальної оптимізації усіх об’єктів, тобто дальність промальовування об’єктів на сцені та робота над зменшенням детальності світла, задля поліпшення продуктивності. Також, було нами використана технологія Nanite, використовуючи цю технологію, ПК не малює кожної моделі, щодо кількості її полігонів. Завдяки ньому зображення виводиться роздільною здатністю один полігон на піксель, тобто чим вища роздільна здатність монітора, то тим і вища кількість полігонів (деталізація моделі).

Отже, цей проект успішно досягнув своєї мети - створення інтерактивного віртуального середовища для НаУОА, що включає інтер'єри першого та другого

поверхів нового корпусу, з використанням Blender та Unreal Engine 5 та в подальшу оптимізацію. Це інтерактивне середовище може служити засобом представлення загального враження для вступників, які мають можливість віртуально оглянути новий корпус. Використання Blender дозволило створити деталізований інтер'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.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Брито А. Blender 3D Architecture, Buildings, and Scenery / Allan Brito. – 2015. – Режим доступу: http://surl.li/ueofc (дата звернення: 01.12.2023 р.).

2. Брито А. The Blender 3D Architect: Design and Modeling for a Sustainable Built Environment / Allan Brito. – 2018. – Режим доступу: http://surl.li/ueofk (дата звернення: 03.12.2023 р.).

3. Кодрон Р. Blender 3D: Designing Objects / Romain Caudron. – 2019. – Режим доступу: http://surl.li/ueofs (дата звернення: 04.12.2023 р.).

4. Adobe Photoshop URL: http://surl.li/tsatl (дата звернення: 06.12.2023 р.).

5. Lumen Global Illumination and Reflection URL: http://surl.li/tsaue (дата звернення: 09.12.2023 р.).

6. Official site Unreal Engine URL: http://surl.li/tsaqw (дата звернення: 10.12.2023 р.).

7. Interface Unity URL: https://unity.com/releases/2019-3/editor-tools (дата звернення: 11.12.2023 р.).

8. Unity documentation URL: https://docs.unity.com/ (дата звернення: 12.12.2023 р.).

9. Using shader graph URL: https://docs.unity3d.com/Manual/shader-graph.html (дата звернення: 13.12.2023 р.).

10. High Definition Render Pipeline URL: http://surl.li/tsarm (дата звернення: 14.12.2023 р.).

11. Physics URL: https://docs.unity3d.com/Manual/PhysicsSection.html (дата звернення: 16.12.2023 р.).

12. NVIDIA PhysX URL: https://unity.com/solutions/programming-physics (дата звернення: 18.12.2023 р.).

13. Introduction Blender URL: http://surl.li/hhlbz (дата звернення: 22.12.2023 р.).

14. Introduction 3ds Max URL: http://surl.li/trzxk (дата звернення: 28.12.2023 р.).

15. Blender render Cycles URL: https://docs.blender.org/manual/ru/dev/render/cycles/index.html (дата звернення: 02.01.2024 р.).

16. Autodesk 3ds Max render Arnold URL: http://surl.li/trzzp (дата звернення: 02.01.2024 р.).

17. Using Python API in Blender URL: https://docs.blender.org/api/current/index.html (дата звернення: 05.01.2024 р.).

18. License Blender URL: https://www.blender.org/about/license/ (дата звернення: 07.01.2024 р.).

19. Price Autodesk 3ds Max Software URL: http://surl.li/gyybe (дата звернення: 09.01.2024 р.).

20. Офіційний сайт НаУОА URL: www.oa.edu.ua (дата звернення: 15.01.2024 р.).

21. Lidar URL: https://icoola.ua/blog/sho-take-lidar-scanner/ (дата звернення: 16.01.2024 р.).

22. Introduction to Jira URL: http://surl.li/taajd (дата звернення: 18.01.2024 р.).

23. About Bugzilla URL: https://www.bugzilla.org/about/ (дата звернення: 22.01.2024 р.).

24. Events in Unreal Engine 5 URL: http://surl.li/tsamh (дата звернення: 25.01.2024 р.).

25. Using Widgets in Unreal Engine 5 URL: http://surl.li/tsaqq (дата звернення: 28.01.2024 р.).

26. UMG UI Designer URL: http://surl.li/tsalm (дата звернення: 29.01.2024 р.).

27. Specialized Blueprint Node URL: http://surl.li/tsalz (дата звернення: 30.01.2024 р.).

28. EventHit URL: http://surl.li/tsany (дата звернення: 01.03.2024 р.).

29. Creating your backlog URL: http://surl.li/tsaor (дата звернення: 10.03.2024 р.).

30. Physical Material User Guide URL: http://surl.li/tsaqj (дата звернення: 15.05.2024 р.).

ДОДАТКИ

ДОДАТОК 1.

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 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: змоделюйте 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 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 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. Лаптєв Богдан(фасад адмінбудівлі, частина прощі 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 прощі 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 ме 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 різним поверхням.

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 номер