

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 нового корпусу та площі перед ним)»

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

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

Белінського Олександра Олександровича

1. *Тема роботи:* Створення інтерактивного віртуального середовища НаУОА (екстер'єр нового корпусу та площі перед ним).

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

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

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

3. *Вихідні дані до роботи:* для створення проекту використовувались: Blender(для моделювання усіх 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:* зведення всіх конструктивних елементів, розробка екстер'єру нового корпусу «Національного університету Острозька академія», реалізація звуків, розробка принципу роботи годинника, відкривання-закривання шлагбаума, реалізація погодних ефектів.

5. *Перелік графічного матеріалу:* рисунки, 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/проекту	до 31.10.2023 р.	
2	Постановка технічного завдання	до 01.12.2023 р.	
3	Ознайомлення з документацією	до 10.12.2023 р.	
4	Написання розділу 1	до 01.02.2024 р.	
5	Написання розділу 2	до 01.04.2024 р.	
6	Тестування системи	до 20.04.2024 р.	
7	Виправлення помилок	до 01.05.2024 р.	
8	Попередній захист та перевірка на рівень унікальності кваліфікаційної роботи/проекту	до 31.05.2024 р.	
9	Здача кваліфікаційної роботи/проекту на кафедру	31.05.2024 р.	

Студент: _____ **Олександр БЕЛІНСЬКИЙ**
(підпис)

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

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

Тема: Створення інтерактивного віртуального середовища НаУОА (екстер'єр нового корпусу та площі перед ним)

Автор: Белінський Олександр Олександрович

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

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: 60 с., 40 рис., 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 екстер'єру нового корпусу, та площі перед ним, створення віртуального середовища, а саме реалізація годинника в реальному часі, підіймання і опускання шлагбаума, додавання звуків, та реалізація відтворення погодних умов таких як:ясне небо, хмарне небо, дощ та гроза.

Ця робота спрямована на ознайомлення абітурієнтів з новим корпусом Національного університету «Острозька академія», в якій вони зможуть детально розглянути, як він виглядає та функціонує.

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 new building and the square in front of it, create a virtual environment, namely, the implementation of a clock in real time, raising and lowering the barrier, adding sounds, and implementing the reproduction of weather conditions such as: clear sky, cloudy sky, rain and thunderstorm.

This work is aimed at familiarizing applicants with the new building of the National University "Ostroh Academy", in which they will be able to examine in detail how it looks and functions.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1.ТЕОРЕТИЧНА ЧАСТИНА СТВОРЕННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА НАУ ОА (ЕКСТЕР'ЄР НОВОГО КОРПУСУ ТА ПЛОЩІ ПЕРЕД НИМ) ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ BLENDER ТА ІГРОВОГО РУШІЯ UNREAL ENGINE 5	4
1.1. Постановка проблеми створення інтер'єру першого та другого поверхів нового корпусу, опис базових понять про Blender та Unreal Engine.	4
1.2. Постановка проблеми, опис базових понять про Unreal Engine 5	9
1.3. Аналіз аналогічних розробок.....	11
1.5. Інтерактивна частина розробки проекту	13
РОЗДІЛ 2.ПРИКЛАДНА РОЗРОБКА ЕКСТЕР'ЄР НОВОГО КОРПУСУ ТА ПЛОЩІ ПЕРЕД НИМ В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ BLENDER.	19
2.2. Опис технологічного стеку	22
2.3. Збір основної інформації для виконання завдання.....	25
2.4. Створення 3Д моделей в Blender	26
2.5. Інтеграція 3Д моделей в Unreal Engine.	30
2.6. Створення інтерактивної частини Нау ОА (екстер'єр нового корпусу та площі перед ним).....	37
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ	55

ВСТУП

У сучасному світі віртуальна реальність та інтерактивні технології перетинають межі реального та віртуального, створюючи нові можливості для взаємодії та навчання. Університети, як центри знань та інновацій, активно використовують ці технології для створення навчальних середовищ, що сприяють активному та ефективному навчанню. У цьому контексті Національний університет "Острозька Академія" розглядається як приклад використання віртуальних технологій для створення неперевершених навчальних та інноваційних середовищ.

Метою даного проєкту є: створення інтерактивного віртуального середовища НаУ ОА, а саме, що стосується моєї частини роботи, це екстер'єр нового корпусу, та площі перед ним. Ця робота була створена для ознайомлення студентів та абітурієнтів з новим корпусом в онлайн режимі, адже багато людей жодного разу не бували в Острозькій академії і навіть не уявляють який вигляд вона має. Саме для них ми створили наш проєкт.

Завдяки нашому проєкту, ми створили оригінальне віртуальне середовище, яке існує і має здатність розвиватись тільки у нашому університеті, та не має аналогів в жодному університеті нашої країни. За основу, як об'єкт, ми взяли Новий корпус Національного Університету "Острозька академія", який ми максимально реалістично відтворили в віртуальному середовищі такому як Unreal Engine 5.

Актуальність таких розробок як наша, на даний час є дуже популярною, адже з кожним роком ігрова індустрія розвивається шаленими темпами, а кількість користувачів збільшується в рази бажаючи протестувати нові продукти, які з'являються на ринку, тому на даний момент це є дуже актуальним.

Основними завданнями роботи були: розгортання своєї частини проєкту в Jira, створення погоди, годинника, звукових ефектів, шлагбаума, та оптимізацію усіх цих деталей для коректної роботи загального проєкту.

РОЗДІЛ 1

ТЕОРЕТИЧНА ЧАСТИНА СТВОРЕННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА НАУ ОА (ЕКСТЕР'ЄР НОВОГО КОРПУСУ ТА ПЛОЩІ ПЕРЕД НИМ) ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ BLENDER ТА ІГРОВОГО РУШІЯ UNREAL ENGINE 5

1.1. Постановка проблеми створення інтер'єру першого та другого поверхів нового корпусу, опис базових понять про Blender та Unreal Engine.

Сучасний розвиток вищої освіти вимагає впровадження інноваційних підходів до організації навчального процесу, включаючи використання віртуальних середовищ для створення та візуалізації нових об'єктів архітектурного середовища. Однак створення віртуального екстер'єру нового корпусу та прилеглої до нього території стикається із значними труднощами та вимагає комплексного підходу до вирішення ряду проблем:

1. Аналіз та дослідження:

- Глибоке розуміння функціональних, естетичних та ергономічних аспектів: Необхідність аналізу архітектурних особливостей вимагає глибокого вивчення функціональних потреб нового корпусу та прилеглої території. Це включає розуміння того, як простір використовується для навчання, роботи та взаємодії. Естетичні аспекти стають важливими з точки зору забезпечення гармонії з навчально-дослідницькими цілями та створення приємного середовища. Ергономіка включає в себе вивчення того, як простір впливає на зручність та комфорт користувачів, забезпечуючи ефективність та задоволення від використання.
- Взаємодія з природним середовищем та динаміка простору: аналіз архітектурних особливостей також охоплює взаємодію з природним середовищем. Це включає розуміння впливу нового корпусу та прилеглої території на ландшафт, врахування екологічних аспектів та створення просторів, що сприяють взаємодії з природним оточенням. Динаміка простору включає

аналіз відкритих та закритих зон, зон для відпочинку та роботи, а також врахування потреб різних груп користувачів.

2. Технічні рішення та інструменти:

- Вибір програмного забезпечення та інтеграція інструментів: Вибір програмного забезпечення є важливим етапом для ефективності створення віртуального екстер'єру. Необхідно вибрати інструменти, які забезпечують високу якість графіки, швидкість обробки та можливість взаємодії з користувачем. Інтеграція інструментів також важлива для забезпечення сумісності та злагодженої роботи між різними аспектами віртуального середовища.

- Технічні рішення для оптимізації процесу створення та взаємодії: технічні рішення повинні бути спрямовані на оптимізацію процесу створення віртуального екстер'єру. Це включає в себе використання продуктивних та ефективних інструментів для моделювання, текстурювання та анімації. Питання сумісності та взаємодії інструментів повинні вирішуватися для плавного та безперебійного використання системи.

3. Валідація та віртуальна реалізація:

- Розробка Тестувальних та Валідаційних Процедур: Розробка валідаційних та тестувальних процедур є ключовою для забезпечення точності та реалістичності віртуальної моделі. Це включає створення стандартів для порівняння параметрів та характеристик віртуальної та реальної моделі.

Blender[1] - це потужний і вільний програмний пакет для 3D-моделювання, анімації, композитингу, рендерингу, обробки зображень і багато іншого. Ось деякі базові поняття, пов'язані з використанням Blender:

1. Інтерфейс користувача:

- 3D Viewport[23]: основне вікно, в якому ви працюєте з об'єктами в 3D-просторі.

- Tool Shelf: панель інструментів, яка зазвичай розташована ліворуч екрану і містить інструменти для моделювання, текстурювання, тощо.

- Properties Panel: панель властивостей для налаштувань об'єктів, сцени, матеріалів і багато іншого.

- Timeline: панель, що відображає часову шкалу для анімації.

2. Об'єкти:

- Meshes: геометричні об'єкти, такі як куби, сфери, які можна моделювати та маніпулювати.

- Cameras: об'єкти, які визначають точку зору для рендерингу сцени.

- Lights: Джерела світла, використовуються для освітлення сцени.

3. Моделювання:

- Edit Mode: режим, в якому ви можете редагувати вершини, ребра та поверхні мешу.

- Modifiers: функції, що додають або змінюють геометрію об'єкта (наприклад, Mirror, Subdivision Surface).

4. Матеріали і Текстури:

- Materials[24]: налаштування, які визначають вигляд об'єкта (колір, властивості відбиття світла і т.д.).

- Textures: зображення, які можна застосовувати до об'єктів для створення реалістичного вигляду.

5. Анімація:

- Keyframes: кадри, які визначають значення параметрів в певний момент часу.

- Dope Sheet і Graph Editor: панелі для редагування та візуалізації анімаційних ключів.

6. Рендеринг:

- Render Engine: алгоритм, який перетворює 3D-сцену в зображення (наприклад, Cycles або Eevee).

- Render Settings: налаштування, пов'язані з процесом рендерингу, такі як роздільність, освітлення, тощо.

7. Камера і Види:

- Camera View: вид з обраної камери.

- Orthographic і Perspective View: різниця в режимах відображення сцени.

8. Інші інструменти:

- Sculpt Mode: режим для творчого моделювання з використанням інструментів скульптування.
- Texture Paint і Weight Paint: режими для розфарбовування текстур і вагових картинок для анімації.

Blender демонструє високий рівень гнучкості та відкритості для інтеграції з іншими програмами та платформами. Нижче подано деякі аспекти його інтеграції:

- Імпорт та Експорт: Blender підтримує ряд форматів файлів для імпорту та експорту, включаючи OBJ, FBX, STL, Collada та багато інших. Це дозволяє обмінюватися моделями з іншими програмами для 3D-моделювання та рендерингу.
- Adobe After Effects[25]: є плагіни, які дозволяють імпортувати 3D-моделі з Blender безпосередньо в Adobe After Effects для створення складних візуальних ефектів.
- Unity[2]: Blender може експортувати моделі та анімації для використання в ігрових двигунах, таких як Unity. Також існують плагіни, що полегшують обмін даними між Blender та Unity.
- Unreal Engine[3]: аналогічно, Blender і Unreal Engine сумісні для обміну моделями та анімаціями.
- Blender-Maya Bridge: існують інструменти та плагіни, такі як Bifrost Bridge, що полегшують інтеграцію між Blender та Autodesk Maya для обміну даними та використання обох програм в одному проекті.
- Blender-3ds Max Bridge: аналогічно, існують інструменти для спрощення інтеграції між Blender та Autodesk 3ds Max.
- Natron та Blender: Natron, програма для композитингу та редакції відео, може взаємодіяти з Blender для імпорту руху та анімацій.

У своїй роботі я експортував з Blender деякі деталі декору та вікна саме в Unreal Engine 5, так як ми використовували саме цю програму для створення нашого проекту.

Blender вирізняється на тлі інших програм для моделювання з численними перевагами, що забезпечують йому популярність та широкий спектр застосувань. Ось деякі з основних переваг Blender:

- **Безкоштовність та Відкритість Коду:** Blender є повністю безкоштовним та відкритим за розповсюдженням програмним забезпеченням. Це означає, що ви можете використовувати, змінювати та поширювати програму без обмежень.
- **Широкі Функціональні Можливості:** Blender надає вражаючий спектр функціональностей, включаючи моделювання, текстурування, рендеринг, анімацію, композитинг, вирізання відео, аудіо редагування та багато іншого. Він може використовуватися для створення від візуальних ефектів у кіноіндустрії до ігор, архітектурного дизайну та навіть наукових візуалізацій.
- **Спільнота та Підтримка:** Blender має активну та велику спільноту користувачів та розробників. Це означає доступність різноманітних плагінів, додатків, а також можливість обговорення та отримання допомоги від інших користувачів через форуми та інші ресурси.
- **Платформна Незалежність:** Blender підтримується на різних операційних системах, включаючи Windows, macOS та Linux, що робить його доступним для широкого кола користувачів.
- **Високоякісний Рендеринг:** вбудований рендерер Cycles у Blender надає високоякісні фотореалістичні зображення та підтримує відслідковування променів (ray tracing). Також є інші рендерери, такі як Eevee, який призначений для швидкого попереднього перегляду та інтерактивного рендерингу.
- **Надзвичайна Гнучкість та Розширюваність:** Blender надає гнучкість в налаштуванні та розширюваності завдяки вбудованій системі Python-скриптів. Користувачі можуть створювати власні інструменти та плагіни для задоволення своїх конкретних потреб.
- **Професійні Засвідчені Приклади Використання:** Blender використовується в ряді професійних проєктів, включаючи створення візуальних ефектів для фільмів, анімацію для відеоігор, моделювання архітектурних проєктів та багато інших галузей.

Ці переваги встановлюють Blender як важливий інструмент для творчого моделювання та візуалізації в різних областях.

1.2. Постановка проблеми, опис базових понять про Unreal Engine 5

Розвиток та вдосконалення ігрової індустрії вимагає постійного вдосконалення технологій, використовуваних для створення вражаючих віртуальних світів. Одним із ключових викликів у цьому контексті є необхідність високоефективних та потужних ігрових рушіїв, здатних забезпечити найсучасніші графічні та фізичні можливості. В цьому контексті, Unreal Engine 5 представляє собою обіцяний інструмент для розробників.

Список проблем на які можна натрапити:

- Оптимізація Продуктивності:

проблема: з високою якістю графіки та складними обчисленнями, проекти в Unreal Engine 5 можуть зазнавати низької продуктивності, особливо на менш потужних системах.

Завдання: розробка стратегій оптимізації для підвищення продуктивності, використання механізмів LOD[4] (рівні деталізації) та оптимізоване управління ресурсами.

- Використання Nanite та Lumen[5]:

Проблема: несправне використання технологій Nanite та Lumen може впливати на продуктивність та візуальну якість проекту.

Завдання: вивчення та правильне використання Nanite для деталізації геометрії та Lumen для досягнення реалістичного освітлення.

- Оптимізація завантаження світу (World Partition):

Проблема: завантаження великих гігабайтних ігрових світів може впливати на час завантаження та продуктивність гри.

Завдання: впровадження стратегій World Partition для ефективного завантаження та управління ресурсами, враховуючи зони зору гравців.

- Інтеграція та Робота з Зовнішніми Джерелами:

Проблема: взаємодія та інтеграція з зовнішніми програмами, такими як 3D-моделювальні інструменти, може призводити до проблем у форматів файлів та різноманітних технічних аспектах.

Завдання: Використання інструментів, таких як Datasmith[26], для ефективною інтеграції та оптимізації обміну даними з іншими програмами.

Опис базових понять про Unreal Engine 5

- Nanite - це технологія роботи з геометрією, що використовується в UE5. Вона дозволяє виводити дуже великі кількості деталей та текстур, забезпечуючи вражаючий рівень деталізації в іграх та віртуальних середовищах. Зазвичай використовується для створення складних об'єктів, таких як деталі ландшафту, текстуровані поверхні та деталізовані моделі.

- Lumen - це глобальна система освітлення в UE5, яка автоматично розповсюджує світло у великих віртуальних світах. Вона забезпечує реалістичне освітлення навіть у режимі реального часу, що покращує якість графіки та іммерсію. Використовується для створення реалістичного освітлення в іграх та віртуальних середовищах.

- World Partition[27] - це механізм, який автоматично розділяє великі ігрові світи на логічні "комірки". Він завантажує лише ті області, які знаходяться в полі зору гравця, що покращує продуктивність та зменшує час завантаження. Зазвичай використовується для оптимізації завантаження гігантських ігрових світів та полегшення роботи ігрового двигуна.

- MetaHuman Creator[6] - інструмент, що дозволяє створювати реалістичних цифрових людей за допомогою штучного інтелекту. Це значно спрощує та прискорює процес створення персонажів. Використовується для створення живих та реалістичних персонажів у віртуальних світах та іграх.

- Datasmith - інструмент для легкого імпортування даних з різних CAD-систем та інших джерел у формат, придатний для використання в UE5. Зазвичай використовується для архітектурної візуалізації та імпортування даних зовнішніх джерел.

1.3. Аналіз аналогічних розробок

При аналізі аналогічних проєктів нашого напрямку діяльності, виявилось, що лише декілька з них наближаються до складності та обсягу нашого проєкту. Основними викликами, які ускладнюють розробку, є великі витрати ресурсів та часу. Це в свою чергу робить наш проєкт унікальним і поза конкуренцією в даному сегменті.

Нещодавно ми також виявили 3D-модель ОА від Skeiron, яка, незважаючи на свою наявність, не може виправдати наших високих стандартів якості. Наш проєкт вирізняється завдяки можливості віртуального проходження по території Острозької академії, що є неперевершеним в контексті подібних ініціатив.



Рис 1.1. Вигляд віртуального середовища НаУ ОА, проєкту від “Україна поруч”

Джерело: Google Arts & Culture

Важливо відзначити, що наші основні конкуренти - інша група, яка працює над проєктом основного корпусу Острозької академії. Однак наше конкурентне перевага полягає в унікальності нашого підходу та можливостях, які ми надаємо.

1.4. Використання системи управління проектами Jira

Jira[7] - це інструмент управління проектами, який допомагає командам ефективно планувати, відстежувати та керувати проектами будь-якої складності. Вона надає широкі можливості для організації робочих процесів та сприяє ефективній комунікації в команді.

У Jira є ряд корисних функцій. Користувачі можуть створювати та організовувати різні типи задач, встановлювати терміни виконання, призначати відповідальних осіб та відстежувати прогрес виконання завдань. Крім того, Jira дозволяє спільно працювати над проектом, обмінюватися коментарями та документами, що сприяє покращенню комунікації в команді. Система має гнучкі можливості налаштування, які дозволяють адаптувати її під конкретні потреби та процеси проектування.

Впровадження Jira в практичну діяльність передбачає проведення навчання щодо основ користування системою, налагодження інтеграції з іншими інструментами та розробку стандартів використання системи в команді.

Хоча Jira має багато переваг, включаючи зручний інтерфейс та ефективне управління проектами, вона також має свої недоліки, такі як висока вартість ліцензії та складність в освоєнні для нових користувачів.

Мій власний практичний досвід використання Jira підтверджує її ефективність у плануванні та відстеженні прогресу проектів. За допомогою Jira ми змогли покращити комунікацію в команді та забезпечити більш ефективне управління ресурсами.

У висновках можна зазначити, що Jira є потужним інструментом для управління проектами, який може допомогти командам досягти більшого успіху в їхніх проектах. Однак важливо бути готовим до викликів і складнощів, які можуть виникнути під час впровадження та використання системи.

1.5. Інтерактивна частина розробки проекту

Загальна інформація про Blueprints в Unreal Engine 5:

Blueprints [8] в Unreal Engine 5- це дивовижний інструмент, що відкриває безмежні можливості для розробки ігор. Ця візуальна система програмування дозволяє розробникам будувати складні ігрові системи, використовуючи лише графічний інтерфейс, без необхідності глибоких знань у програмуванні. Це робить їх доступними для широкого кола користувачів, включаючи художників, дизайнерів і навіть тих, хто має обмежений досвід у програмуванні.

Одна з ключових переваг Blueprints - це їх візуальний інтерфейс, який дозволяє представляти програмну логіку у вигляді блоків та з'єднань. Це полегшує розуміння коду та прискорює процес розробки. Розробники можуть легко створювати різноманітні ігрові ефекти, системи управління, штучний інтелект та багато іншого, просто перетягуючи та з'єднуючи блоки відповідно до потреб проекту.

Ще одна важлива перевага Blueprints - це їх гнучкість та розширюваність. Завдяки ним розробники можуть легко змінювати та вдосконалювати готові рішення, а також створювати нові, відповідно до потреб проекту. Швидкість розробки також є великим плюсом, оскільки Blueprints дозволяють швидко створювати прототипи та експериментувати з ідеями без необхідності в глибоких знаннях програмування.

Інтеграція Blueprints з C++[28] є ще однією важливою функцією. Розробники можуть поєднувати візуальні блоки зі складним кодом на C++, отримуючи при цьому кращі можливості програмування та зберігаючи зручність використання візуального підходу Blueprints.

Одним з ключових аспектів використання Blueprints є можливість командної роботи. Розробники можуть спільно працювати над проектом, використовуючи системи контролю версій та спільного доступу до коду. Це дозволяє ефективно вести проект та обмінюватись ідеями з колегами.

Широка база документації та активна спільнота Unreal Engine додаються до переваг Blueprints, надаючи необхідну підтримку та допомогу для вивчення та використання цього інструменту.

У підсумку, використання Blueprints в Unreal Engine 5 відкриває безмежні можливості для створення складних інтерактивних ігрових середовищ, дозволяючи розробникам з різних сфер легко і ефективно створювати захоплюючі ігри.

Тепер розглянемо як можна використовувати Blueprints в Unreal Engine 5:

Створення логіки гри: Одним з основних способів використання Blueprints є створення логіки гри. Розробники можуть використовувати блоки та з'єднання, щоб визначити реакції персонажів на різні події в грі, такі як зіткнення з об'єктами, взаємодія з іншими персонажами, зміни в стані гравця тощо.

Створення UI ігрового інтерфейсу: Blueprints можна використовувати для створення інтерактивних елементів інтерфейсу користувача (UI) в грі. Розробники можуть створювати кнопки, текстові поля, списки та інші елементи, які реагують на дії гравця за допомогою Blueprints.

Створення анімацій і персонажів: Blueprints дозволяють розробникам створювати складні анімації для персонажів у грі. Вони можуть визначати рухи, взаємодію та інші аспекти поведінки персонажів за допомогою візуального інтерфейсу.

Реалізація штучного інтелекту (AI)[29]: Blueprints можна використовувати для реалізації штучного інтелекту для NPC (некерованих персонажів) у грі. Розробники можуть створювати різноманітні алгоритми поведінки для NPC, визначаючи їх реакції на різні ситуації та взаємодію з іншими об'єктами у грі.

Створення спеціальних ефектів та взаємодій: Blueprints можна використовувати для створення різноманітних спеціальних ефектів та взаємодій в грі, таких як вибухи, поглиблення, ефекти частинок, зміни погодних умов тощо.

Розширення функціональності гри: Крім того, Blueprints можуть використовуватися для розширення функціональності гри, додавання нових функцій та можливостей, удосконалення існуючих систем та багато іншого.

Загалом, Blueprints в Unreal Engine 5 є потужним інструментом для створення складних ігрових систем та ефектів без необхідності у глибоких знаннях програмування. Вони надають розробникам можливість швидко та ефективно реалізовувати свої ідеї та втілювати їх у життя.

Для збільшення функціоналу Blueprints використовуються такі елементи як:

Графи (Graphs):

У контексті Blueprints у Unreal Engine 5, термін "графи" (Graphs) відноситься до візуальних представлень логіки програмування. Графи використовуються для створення різноманітних ігрових систем, від управління персонажами та анімації до обробки подій та взаємодії об'єктів у грі.

У Blueprints можуть бути різні типи графів, такі як:

Event Graph: Це один з основних типів графів у Blueprints, який використовується для обробки подій, таких як натискання клавіш, колізії об'єктів, анімації персонажів тощо. В Event Graph зазвичай реалізується логіка відгуку об'єктів гри на різноманітні події.

Construction Script: Цей граф використовується для налаштування параметрів та створення об'єктів на сцені під час їх побудови або редагування. Construction Script зазвичай використовується для автоматизації процесів, таких як створення складних мап або налаштування властивостей об'єктів.

Function Graph: Це граф, де можна створювати та зберігати функції для подальшого використання. Функції можуть бути використані для виконання певних завдань або обчислень у різних частинах вашої програми.

Material Graph: Цей тип графа використовується для створення матеріалів, які використовуються для текстуровання об'єктів у грі. У Material Graph ви можете створювати складні матеріали, які реагують на світло, тіні та інші ефекти.

Кожен з цих графів має свої власні особливості та призначення, але вони всі дозволяють розробникам створювати складні ігрові системи за допомогою візуального програмування без необхідності писати код. Це робить Unreal Engine 5 та Blueprints доступними для широкого кола користувачів, включаючи тих, хто не має глибоких знань програмування.

В Unreal Engine 5 існують різні типи графів, такі як Event Graph (для обробки подій), Construction Script (для ініціалізації об'єктів), Function Graphs (для організації функцій), Material Graphs (для створення матеріалів), Animation Graphs (для анімації персонажів) та інші.

Функції (Functions):

У контексті Blueprints у Unreal Engine 5, "функції" (Functions) є важливим елементом візуального програмування, який дозволяє розробникам створювати власні блоки логіки та перевикористовувати їх у різних частинах гри. Функції використовуються для групування певного фрагменту коду в один блок, який може бути викликаний з інших частин програми.

Основні характеристики та використання функцій у Blueprints в Unreal Engine:

Перевикористання коду: функції дозволяють створювати блоки логіки, які можна використовувати багаторазово в різних місцях вашої гри. Це полегшує розробку та підтримку коду, оскільки одноразово створений фрагмент можна використовувати у багатьох місцях.

Організація коду: функції допомагають організувати ваш код, роблячи його більш структурованим та зрозумілим. Ви можете групувати схожі функції разом, що полегшує їх знаходження та редагування.

Параметри і повернення значень: функції можуть приймати вхідні параметри та повертати значення. Це дозволяє створювати більш універсальні та гнучкі функції, які можуть працювати з різними даними та виконувати різні завдання.

Виклик функцій[30]: функції можуть бути викликані з інших частин програми, що дозволяє використовувати їхню логіку та функціональність у потрібних місцях. Це спрощує структуру програми та полегшує взаємодію між різними елементами гри.

Загалом, функції у Blueprints дозволяють розробникам створювати більш організований, перевикористовуваний та гнучкий код для Unreal Engine 5, що полегшує процес розробки гри та збільшує продуктивність.

Макроси (Macros)[9]:

У контексті Blueprints у Unreal Engine 5, "макроси" (Macros) є інструментом, що дозволяє згрупувати послідовність дій або логічний блок коду в один блок, який може бути повторно використаний у різних частинах програми. Макроси спрощують розробку та підтримку коду, особливо коли одні й ті ж дії потрібно виконувати декілька разів.

Основні характеристики та використання макросів у Blueprints в Unreal Engine:

Перевикористання коду: макроси дозволяють створювати блоки логіки або послідовності дій, які можна повторно використовувати в різних місцях вашої програми. Це полегшує розробку та підтримку коду, оскільки одноразово створений блок може бути використаний у багатьох місцях.

Організація коду: макроси допомагають організувати ваш код, роблячи його більш структурованим та зрозумілим. Ви можете групувати схожі функції разом, що полегшує їх знаходження та редагування.

Параметри макросів: макроси можуть мати параметри, які дозволяють їм приймати вхідні дані та змінювати свою поведінку в залежності від цих даних. Це робить макроси більш універсальними та гнучкими.

Підтримка умов та циклів: макроси можуть містити логічні умови та цикли, що дозволяє їм виконувати різні дії в залежності від умови або повторювати певні дії кілька разів.

Виклик макросів: макроси можуть бути викликані з інших частин програми, що дозволяє використовувати їхню логіку та функціональність у потрібних місцях. Це дозволяє спростити структуру програми та полегшити взаємодію між різними елементами гри.

Загалом, макроси у Blueprints дозволяють розробникам створювати більш організований, перевикористовуваний та гнучкий код для Unreal Engine 5, що полегшує процес розробки гри та збільшує продуктивність.

Змінні (Variables):

У контексті Blueprints у Unreal Engine 5, "змінні" (Variables) - це об'єкти, які зберігають та представляють дані, такі як числа, рядки, об'єкти, вектори тощо. Вони використовуються для збереження інформації та передачі даних між різними частинами програми.

Основні характеристики та використання змінних у Blueprints в Unreal Engine:

Зберігання даних: Змінні дозволяють зберігати дані, що використовуються у вашій грі, такі як позиція гравця, кількість життів, стан різних об'єктів тощо. Вони можуть зберігати різні типи даних, включаючи числа, рядки, вектори, булеві значення тощо.

Поширення даних: Змінні можуть бути використані для передачі даних між різними частинами вашої програми. Наприклад, значення змінної, що зберігає позицію гравця, може бути передане у функцію, яка відповідає за переміщення гравця на карті.

Зміна значень: значення змінних можуть змінюватися під час виконання програми. Наприклад, змінна, що зберігає кількість життів гравця, може зменшуватися, коли гравець отримує ушкодження від ворогів.

Локальні та глобальні змінні: у Blueprints можуть бути як локальні, так і глобальні змінні. Локальні змінні існують тільки в межах певного об'єкта або функції, тоді як глобальні змінні можуть бути доступні з будь-якої частини програми.

Імена та типи змінних: кожна змінна має своє унікальне ім'я та тип даних, який вона зберігає. Ім'я змінної використовується для її ідентифікації та виклику у вашій програмі, а тип даних визначає, який вид інформації може бути збережений у цій змінній.

Загалом, змінні у Blueprints дозволяють розробникам ефективно управляти та передавати дані у своїй грі, що дозволяє створювати складні та інтерактивні ігрові системи.

РОЗДІЛ 2

ПРИКЛАДНА РОЗРОБКА ЕКСТЕР'ЄР НОВОГО КОРПУСУ ТА ПЛОЩІ ПЕРЕД НИМ В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ BLENDER

2.1. Використання Jira для управління проектом

Під час запуску проекту в Jira Software ми провели структурування обов'язків, розподіливши їх між учасниками команди через створення окремих спринтів. В кожному спринті вже були зазначені відповідні завдання для виконання (Tasks). При цьому кожен спринт мав свій визначений часовий проміжок для завершення, а також було встановлено певний статус для кожного завдання, що вказував на його фазу у процесі розробки. Такий підхід дозволяє кожному учаснику команди чітко уявити свої завдання та визначити їх прогрес у контексті загальної роботи над проектом.

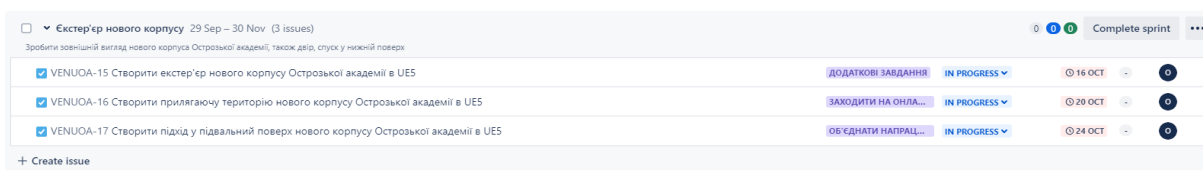


Рис. 2.1. Показ наявних спринтів у вкладці «Backlog»

Джерело: створено автором

Щоб подивитися детальну інформацію, потрібно натиснути на поле, яке вам потрібне і вся інформація, яка вам доступна висвітлиться в меню з права.

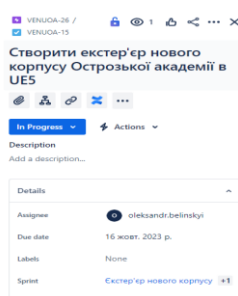


Рис. 2.2. Вікно конкретного завдання

Джерело: створено автором

У нашій системі управління завданнями, яка подібна до дошки, відображаються різні стовпці, що представляють етапи виконання роботи командою. Кожен стовпець має свої унікальні задачі, які переміщуються вздовж дошки залежно від їхнього стану виконання. Такий підхід дозволяє чітко відслідковувати прогрес роботи та забезпечує ефективне управління проектом.

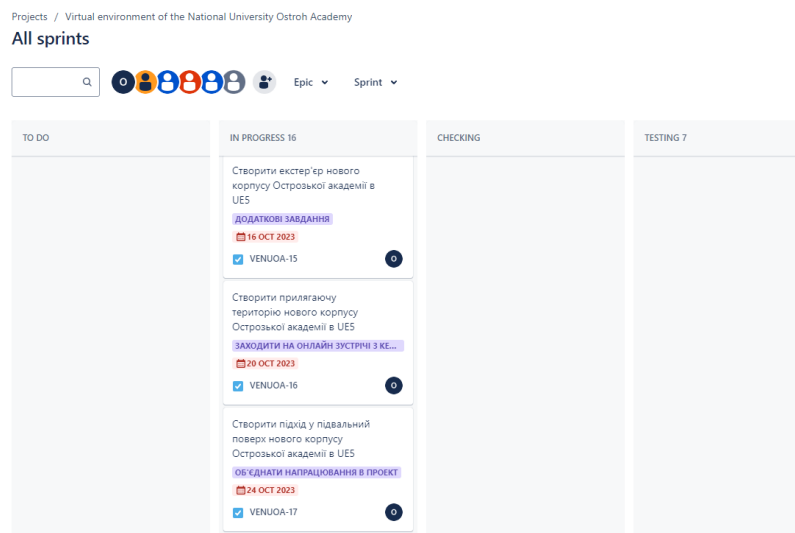


Рис. 2.3. Демонстрація дошки Scrum

Джерело: створено автором

На цьому знімку видно, що ми створили п'ять стовпців, кожен з яких відображає етапи проходження кожного завдання та його перевірки. Перший стовпець - це "TO DO", що вказує на завдання, які мають бути виконані у майбутньому. Другий стовпець - "IN PROGRESS", означає завдання, які знаходяться у процесі виконання. Третій стовпець - "CHECKING", позначає етап перевірки завдань після їх виконання. Четвертий стовпець - "TESTING", це етап тестування механіки для виявлення багів і подальшого їх виправлення. П'ятий стовпець - "DONE", відображає завдання, які успішно пройшли всі попередні етапи та вважаються завершеними.

У Jira Software[10] робочим процесом називається послідовність дій, яку завдання проходять від створення до завершення. Кожен робочий процес складається з набору статусів та переходів між ними, які відображають рух завдання протягом його життєвого циклу. Робочий процес Jira відображає

процеси роботи команди з керування завданнями та відображається у структурі дошки завдань.



Рис. 2.4. Налаштований робочий процес під наші потреби

Джерело: створено автором

На цьому знімку чітко демонструється послідовність процесів, яку проходить кожне завдання, щоб досягти статусу «DONE». Наприклад, відомо, що завдання не може прямувати зі статусу «TESTING» безпосередньо до «DONE». Спочатку воно має пройти через статус «CHECKING», а лише потім може бути переведене в «DONE». Цей процес гарантує, що кожне завдання проходить всі необхідні етапи перевірки та тестування перед завершенням.

Крім того, для кожного випуску визначені свої типи завдань. Наприклад, "Еріс" використовується для відстеження колекцій пов'язаних помилок, історій та завдань. "Bug" використовується для відстеження конкретних проблем або помилок. "Story" використовується для відстеження функціональності або функцій, які виражені як цілі користувача. "Task" використовується для відстеження невеликих окремих частин роботи. "Subtask" використовується для відстеження невеликих фрагментів роботи, які є частиною більшого завдання. Кожен тип завдання відповідає конкретним потребам та функціям у межах проекту.

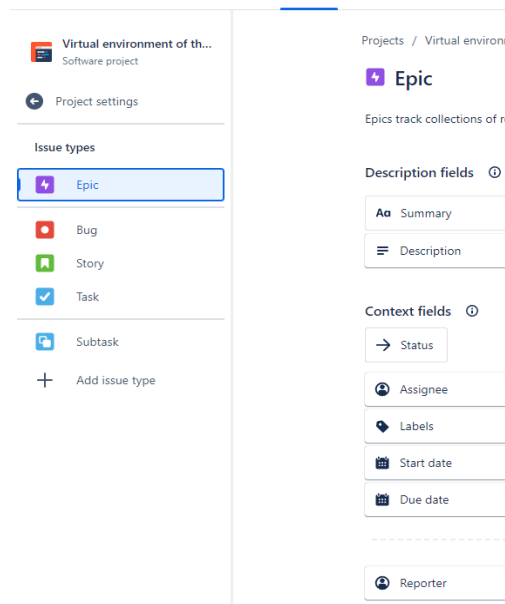


Рис. 2.5. Демонстрація наявних типів

Джерело: створено автором

2.2. Опис технологічного стеку

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

- Photoshop[11]:

Призначений для редагування та обробки растрових зображень.

Застосування:

Ретушування фотографій: видалення дефектів, корекція кольорів, зміна експозиції.

Створення текстур: додавання текстур до фотографій або створення нових.

- Illustrator[12]:

Призначений для створення векторних ілюстрацій та графіки.

Застосування:

Векторний дизайн: створення логотипів, ілюстрацій, векторних малюнків.

Типографіка: Робота з текстовими елементами та шрифтами для створення друкованих матеріалів.

- Blender:

Призначений для моделювання, анімації та рендерингу.

Застосування:

Моделювання: створення тривимірних об'єктів та сцен.

Анімація: створення рухомих об'єктів та персонажів.

Рендеринг: генерація високоякісних зображень з 3D-сцен.

Віртуальна реальність (VR): використання для розробки VR-проектів та віртуальних середовищ.

Також було використано багато доповнень для Blender такі як :

- BlenderKit:

BlenderKit - це онлайн-ресурс, який надає доступ до бібліотеки ресурсів, таких як 3D-моделі, текстури, матеріали та інші об'єкти, які можна використовувати у ваших проектах у Blender.

Застосування: Зручний доступ до різноманітних ресурсів для використання в 3D-проектах безпосередньо з Blender.

- Booltool[13]:

Booltool - це додаток, який полегшує використання булевих операцій (об'єднання, віднімання, перетин) при моделюванні об'єктів у Blender.

Застосування: Спрощує і поліпшує процес створення складних форм та об'єктів за допомогою булевих операцій.

- Image addon:

Image addon - може надавати додаткові інструменти для редагування та обробки текстур або зображень.

- Real Landscape:

Real Landscape - це аддон, який надає інструменти для створення природних ландшафтів та рельєфів у Blender.

Застосування: Допомогає швидко та ефективно створювати реалістичні ландшафти та географічні форми.

- Real Texture:

Real Texture - це аддон для роботи з текстурами, що надає додаткові функції для редагування та використання текстур в Blender.

Застосування: Покращення процесу роботи з текстурами для отримання більш реалістичних та якісних результатів.

Unreal Engine 5 - це високопродуктивний ігровий рушій, розроблений компанією Epic Games. Він є потужним інструментом для створення ігор, симуляцій, віртуальної реальності та інтерактивних візуальних додатків. UE5 відомий своєю передовою графікою та включає у себе нові технології, такі як Nanite Virtualized Geometry та Lumen Global Illumination, що підвищують реалістичність та динаміку візуального відображення в реальному часі.

Також було використано багато доповнень для Unreal Engine 5 такі як :

- Quixel Bridge:

Quixel Bridge-це платформа для швидкого та зручного вибору, перегляду та імпортування текстур, 3D-моделей та інших ресурсів в ігровій розробці.

Застосування: використовується для ефективного управління та імпортування ресурсів, наданих Quixel, в інші ігрові рушії чи програми для створення контенту.

- Real Sky:

Real Sky- це текстурний або графічний ресурс для створення реалістичного вигляду неба у віртуальних просторах.

Застосування: використовується для надання реалістичного фону та атмосферного освітлення в ігрових сценах чи візуальних проектах.

- Real Wall:

Real Wall - це аддон для створення реалістичних текстур стін або 3D-модель, яка імітує стіну.

Застосування: використовується для створення деталей архітектурних об'єктів та реалістичного дизайну стін в іграх чи візуальних проектах.

- Texture:

Текстури - це графічні зображення, які використовуються для покриття поверхонь 3D-моделей чи створення візуальних ефектів.

Застосування: Використовуються для призначення кольорів, деталей та реалістичності об'єктів у грі чи візуальному контенті. Можуть бути використані в Quixel Bridge або інших програмах для роботи з текстурами.

2.3. Збір основної інформації для виконання завдання

Хід роботи:

1. Отримати схеми поверхів нового корпусу для того щоб зробити каркас нового корпусу для подальшого створення фасаду;
2. Отримати фото та відео для правильної розстановки усіх деталей на площі та фасаді.

Для початку отримали схему(Рис.2.1) по якій можна буде зробити контури для нового корпусу для подальшої роботи.

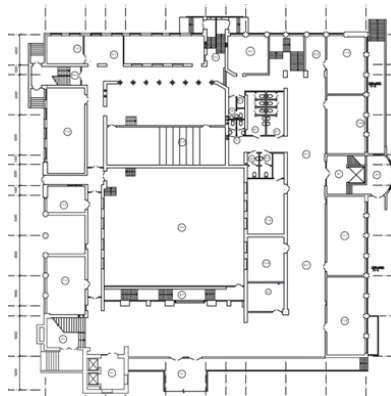


Рис. 2.6. Схема корпусу Національного університету «Острозька академія»

Джерело: розроблено автором

Також підібрав фото які можна буде використовувати для роботи з фасадом та розстановкою усіх бордюрів та поглиблень, а також для розмітки площі для подальшого розміщення елементів декору, та моделей Нового корпусу, та адмін-будівлі.

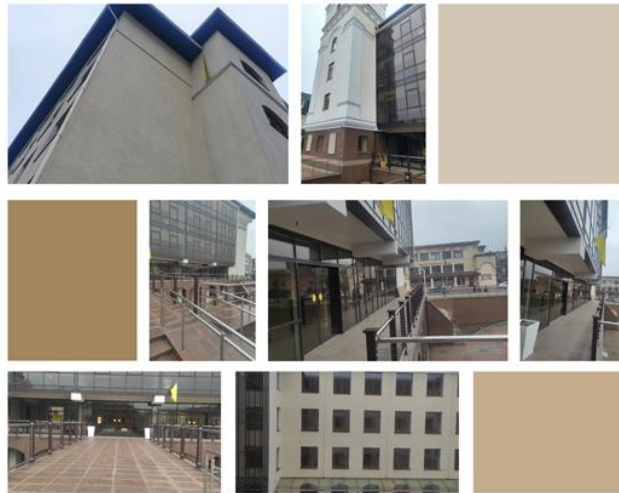


Рис. 2.7. Фасад Національного університету «Острозька академія»

Джерело: розроблено автором

2.4. Створення 3Д моделей в Blender

Початок роботи в Blender (створення двору та каркасу будівлі)

Хід роботи:

1. Створення моделі будівлі;
2. Створення двору та допоміжних деталей.

Завдяки планам будівлі та фото двору вдалося зробити 3D-модель з усіма заглибленнями та розставленими бордюрами щоб можна було візуально підігнати деталі на місця в яких вони розташовані в реальному житті щоб досягти максимально реалістичного вигляду, а щодо будівлі, то в подальшому ми відокремили вежі для їх деталізації, адже для більш комфортної роботи та деталізації не тільки зовні, а й в середині, об'єднане все в одне ціле буде заважати, тому було вирішено розділити все на окремі деталі з якими і проводилась більш детальна робота, саме завдяки цьому ми можемо перенести деякі деталі в ігровий рушій Unreal Engine 5 і це набагато спрощує роботу, адже з окремими деталями набагато простіше працювати і не потрібно перенавантажувати систему запускаючи весь проект.

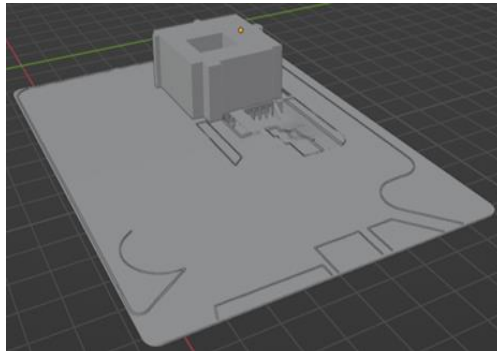


Рис. 2.8. Фасад, двір в програмі Blender Національного університету
«Острозька академія»

Джерело: розроблено автором

Створення дрібних деталей

- 1) Створення 3D моделей вікон та дверей а також підбір кольорів та матеріалів до них.
- 2) Створення ліпнини та віконець на ній ,також підбір кольору та матеріалу.

Для початку було створено об'єкт, та за допомогою комбінацій клавіш Ctrl+B, було зроблено заокруглені виступи, після того на них був нанесений матеріал ,який надав віконній рамі більш реалістичний вигляд, потім для створення вирізів під вікна було використано BoolTool для охайного вирізу без дефектів. віконні рами були створені за допомогою комбінації клавіш Ctrl+W, що дозволяє більш коректно заокруглити верхню частину вікна, в кінці було створено скло тобто об'єкт на який було нанесено матеріал який дає візуальний вигляд скла.



Рис. 2.9. Дрібні деталі вікон в програмі Blender Національного університету
«Острозька академія»

Джерело: розроблено автором

Створення матеріалів та нанесення їх на об'єкти

Хід роботи:

1. Розробка матеріалів;
2. Нанесення матеріалів на об'єкти ;

Найважче мені було з створенням матеріалу, а саме з мармуровою темно-червоною плиткою яка нанесена на вежу та на вхід до цокольного поверху, а також з матеріалом для фасаду нового корпусу через те що вона робиться через нодову систему яка використовується в комп'ютерній графіці, та програмуванні візуальних ефектів, також використовують нодові системи для створення та налаштування матеріалів, освітлення, текстур, анімації та інших аспектів візуальної графіки. Нам вона була потрібна для створення максимально реалістичних текстур і матеріалів, для надання максимально реалістичного вигляду для об'єктів.

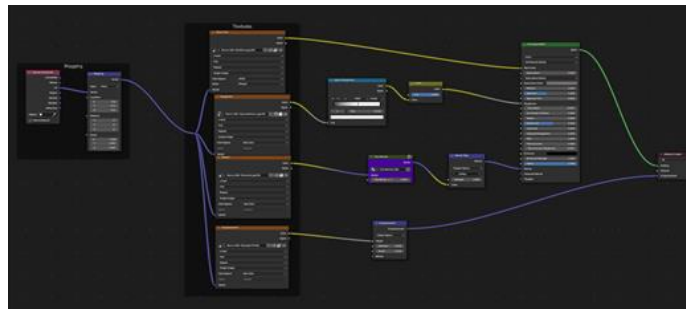


Рис. 2.10. Нод до матеріалу граніту

Джерело: розроблено автором

А ось так воно виглядає на об'єкті після нанесення матеріалу, та налаштування UV-розгортки[14], для правильного розміщення текстури граніту на об'єкті:



Рис. 2.11. Нанесення текстур на центральну вежу Національного університету

«Острозька академія»

Джерело: розроблено автором

При подальшій роботі, по такому ж прикладу робились і інші матеріали, для різних об'єктів таких як: вікна, двері, трава, стіни, поручні, і тд. На ці всі деталі застосовувались матеріали розроблені через нодову систему.

Створення головної вежі її деталізація та додавання матеріалів

Хід роботи:

Для початку роботи ми підготували усе нам необхідне та створили різні дрібні деталі такі як, вікна, годинники, шпиль вежі. Які нам були потрібні для роботи а далі приступили до надання форм і виступів на вежі для більш реалістичного вигляду, тобто почали з розмітки вежі та приступили до надання їй форм. Після того як ми надали вежі форми ми почали робити вирізи під вікна за допомогою Booltool, та розставляти їх, після того я перейшов до накладання матеріалів на цьому робота була закінчена, залишалось лише підігнати вежу на своє місце.



Рис. 2.12. Готова модель з текстурами головної вежі Національного університету «Острозька академія»

Джерело: розроблено автором

Створення моделі нового корпусу його деталізація та накладання матеріалів

Хід роботи:

Для початку роботи ми підготували усе необхідне та створювали різні дрібні деталі(вікна, декоративні виступи на 4 поверсі, двері і тд) які нам були потрібні для роботи, а далі приступили до створення об'ємів тобто почали з розмітки моделі корпусу імпортувавши план будівлі та вивівши стіни по зазначених в плані будівлі та приступили до надання йому форм за допомогою об'єктів які є в самому Blender. Після того як ми надали будівлі форми ми почали

робити місця під вікна та розставляти їх, після того ми перейшли до накладання матеріалів. На цьому робота була закінчена.



Рис. 2.13. Готова модель Національного університету «Острозька академія»

Джерело: розроблено автором

2.5. Інтеграція 3D моделей в Unreal Engine.

Підготовка до інтеграції

Інтеграція 3D моделей[15] в Unreal Engine - це важлива частина розробки візуальної частини нашого проекту. Unreal Engine надає зручні інструменти для роботи з 3D-об'єктами та їх інтеграції в наш проект. Ось кілька кроків, які ми виконали для цього:

Створення 3D моделі:

Використали програми для моделювання, такі як Blender, Autodesk Maya, або 3ds Max, для створення 3D об'єктів.

Переконалися, що наші моделі зберігаються в підтримуваних Unreal Engine форматах, таких як FBX або OBJ.

Імпорт моделей в Unreal Engine:

У Unreal Engine відкрили відповідний проект.

Вибрали вкладку "Content Browser".

Клацнули правою кнопкою миші в області вмісту, та вибрали "Import" і вибрали наші 3D-моделі.

Настроювання матеріалів:

Unreal Engine використовує матеріали для відображення текстур та властивостей моделі.

Додали текстури та матеріали до своїх моделей, використовуючи Material Editor[16] в Unreal Engine.

Розміщення моделей в сцені:

Перетягнули свої 3D-об'єкти з Content Browser в сцену.

Використали Transform рухи, щоб розмістити їх у відповідному місці.

Оптимізація:

Переконалися, що наші 3D-моделі оптимізовані для швидкості та продуктивності. Використовували LODs (Level of Detail), щоб управляти рівнями деталей в залежності від відстані до об'єкта.

Тестування:

Протестували нашу сцену для переконання, що всі 3D-об'єкти правильно відображаються та взаємодіють у відповідь на різні події.

Під час перенесення деталей з проекту з Blender в Unreal Engine було зроблено висновки, що краще переносити лише окремі деталі, а невелику кількість об'єктів одразу, адже тоді виникають проблеми з текстурами.

Налаштування програмного середовища Unreal Engine 5

Хід роботи:

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

Опис роботи:

- Встановлення Unreal Engine 5:

Завантажив та встановив останню версію Unreal Engine 5 з офіційного сайту.

Запустив Unreal Engine 5 Launcher та використав його для створення або вибору проекту.

- Конфігурація проекту:

Обрав налаштування для нашого проекту при створенні проекту.

Визначив параметри, такі як тип проекту (ігровий, архітектурний тощо), шаблон та налаштування для графіки.

- Встановлення джерела анімаційного контенту:

Unreal Engine 5 включає покращений додаток Quixel Bridge для імпортування текстур та 3D-моделей. Зареєструвався на Quixel Bridge та додав його в свою робочу область UE5.

- Опціональне налаштування системи світла (Lumen):

Unreal Engine 5 має нову систему глобального освітлення під назвою Lumen.

- Конфігурація проектних параметрів:

Вибрав налаштування проекту, такі як роздільна здатність екрану, мова, управління пам'яттю тощо.

- Робота з Blueprints:

Unreal Engine використовує систему візуального програмування Blueprints. Додав та налаштував Blueprints для визначення поведінки об'єктів та персонажів у нашому проекті.

- Оптимізація та тестування[17]:

Перевірив налаштування, оптимізував графіку та взаємодію об'єктів для досягнення оптимальної продуктивності.

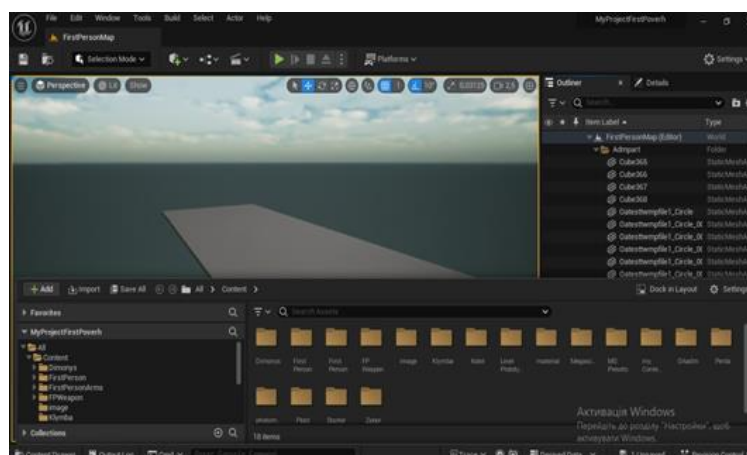


Рис. 2.14. Візуальний вигляд інтерфейсу в програмі Unreal Engine 5

Джерело: розроблено автором

Підготовка моделі для інтеграції.

Хід роботи:

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

Опис роботи:

Перед початком інтеграції 3D-моделі в Unreal Engine, ми провели аналіз та систематично розглянули елементи, які можна обробити безпосередньо в ігровому рушії гри, та ті, які вигідно опрацьовувати в Blender. Цей підхід не лише забезпечив оптимальну інтеграцію, але й дозволив уникнути надмірних проблем на етапі розробки.

Інтеграція 3D-моделей у фреймворк Unreal Engine була неабиякою частиною вибудовування ігор чи віртуальної реальності. З метою досягнення успішної інтеграції ми уважно обирали підтримуваний формат, такий як FBX[18] чи OBJ, і враховували особливості цього формату на етапі моделювання.

Перед імпортом моделі ми переконалися, що всі текстури та матеріали налаштовані належним чином, адже у процесі розробки 3D-моделі в Blender слід враховувати можливості саме цієї програми, створюючи текстури, які ідеально взаємодіють із нею. Оскільки Unreal Engine 5 може не повністю підтримувати окремі матеріали.

Також ми ретельно підготувалися до інтеграції, обираючи формат FBX, щоб уникнути неспівмірності та забезпечити безперебійну візуальну інтеграцію 3D-моделі в Unreal Engine.

Врахувавши всі ці аспекти ми обрали формат FBX для зручної інтеграції моделі і в завершальному результаті все прекрасно імпортувало майже без втрат об'єктів і текстур.

Під час перенесення моделей з Blender в Unreal Engine ми використовували формат FBX щоб уникнути неспівмірності та забезпечити безперебійну візуальну інтеграцію 3D-моделі в Unreal Engine. І в результаті всі роботи з

перенесенням були зроблені правильно, і в завершальному результаті все прекрасно імпортувало, майже без втрат об'єктів і текстур.



Рис. 2.15. Перенесені моделі вежі Національного університету «Острозька академія» в програмі Unreal Engine 5

Джерело: розроблено автором

Імпорт 3D моделей.

Хід роботи:

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

Після підготовки інтеграції центральної вежі та інших дрібних деталей, розпочав імпорт в Unreal Engine 5.

Для цього я виконав такі дії :

- В Unreal Engine вибрав вікно "Content Browser".
- Потім вибрав папку, де я хочу зберегти свою 3D модель.
- Клацнув правою кнопкою миші і обрав "Import".
- Вибрав файл 3D моделі та імпортував його.

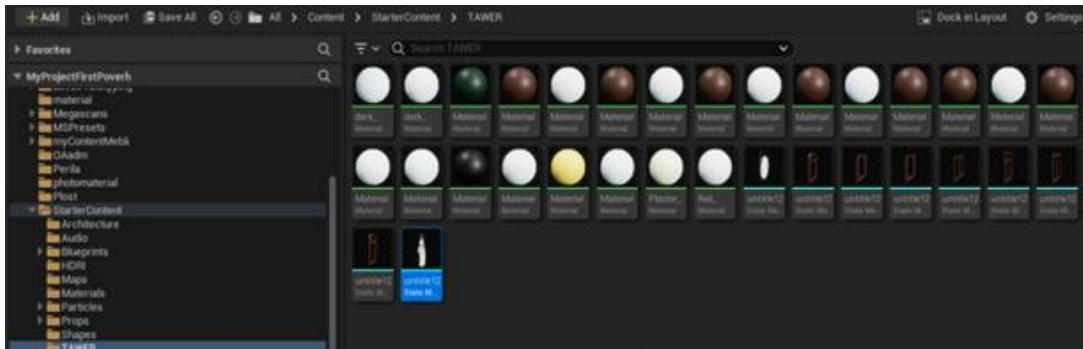


Рис. 2.16. Імпортовані елементи в програмі Unreal Engine 5

Джерело: розроблено автором

Розміщення моделі.

Хід роботи:

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

Опис роботи:

Вибрали відповідне вікно в Unreal Engine та, скориставшись вкладкою "Content Browser", перетягнули наш файл моделі (у форматі FBX) для розміщення його в сцені проекту.

Використовуючи інструменти переміщення, обертання та масштабування для досягнення бажаного розташування взяли нашу модель у сцені, використовуючи вбудовані інструменти переміщення, обертання та масштабування для точного розміщення об'єкта відповідно до нашого задуму розмістили його на його місця розташування.

Оптимізувавши колізії об'єктів для зручності та реалістичності забезпечили адекватну колізію об'єктів, щоб взаємодія з будівлею виглядала більш зручно та реалістично.

У вкладці "Details" визначили тип колізії (наприклад, "Complex Collision"[19] для більш деталізованої колізії чи "Simple Collision"[19] для ефективної, але менш деталізованої). Також провели ці кроки для всіх окремих елементів моделі та налаштували колізію.



Рис. 2.17. Перенесення вежі Національного університету «Острозька академія» в програмі Unreal Engine 5

Джерело: розроблено автором

Деталізація фасаду адмінбудівлі та створення паркових зон з дворовою частиною в Unreal Engine 5

Хід роботи:

1. Деталізація фасаду та внесення змін у конструкцію.
2. Створення Центральної площі перед новим корпусом.
3. Створення задньої дворової частини.
4. Створення різних елементів зовнішньої частини двору.
5. Імпортування різних додаткових елементів до фасаду головного корпусу і веж.
6. Розробка різних дрібних деталей для застосування до фасаду.
7. Тестування та налагодження.

Для початку було розставлено всі дрібні деталі такі як: вікна, двері, виступи, паркани, бордюри, і тд. під час створення площі було вирішено що краще зробити її частинами для кращого накладання матеріалів , та був зроблений спуск і місток до входу в корпус. для створення задньої дворової частини, було розміщено паркан, та накладено матеріали, так як там не багато елементів. Для зовнішньої частини двору, було створено паркани, поручні які розташовані на містку, бордюри та, нанесено матеріали для більш реалістичного вигляду. Також було імпортовано багато додаткових деталей до веж, та самого корпусу (Годинники, вікна, двері), але більша частина роботи була зроблена в

самому ігровому рушії. щодо тестування, то було вирішено додати більше світла, так як в проєкті було стандартне освітлення під час налагодження всі параметри були задовільні і під час тестування все працювало коректно. Після виконання цих пунктів із дотриманням усіх правил розробки ми отримали такий результат:



Рис. 2.18. Готова модель Національного університету «Острозька академія» в програмі Unreal Engine 5

Джерело: розроблено автором

2.6. Створення інтерактивної частини НаУ ОА (екстер'єр нового корпусу та площі перед ним)

Механіка створення звуків

У Unreal Engine 5 (UE5) звуки утворюються за допомогою функціональності, що називається "Physical Material". Physical Material[20] - це об'єкт, який об'єднує в собі фізичні властивості матеріалу, такі як твердість, звукопоглинання, коефіцієнт тертя тощо.

Для початку ми провели дослідження і підібрали звуки, які чудово підходили б до нашого проєкту. Так як у нас екстер'єр нового корпусу та площі

перед ним, найбільше нам потрібні звуки, які максимально описують навколишнє середовище та передають атмосферу максимально схожу до реальності. Обираючи звуки, найбільше нам потрібно було звук ходьби по траві і по бруківці.

Після того як звуки були знайдені та відредаговані ми експортували їх в Unreal Engine 5, після експортування ми почали реалізовувати їх у нашому проекті.

Щоб реалізувати ми створюємо фізичний матеріал і додаємо наші звуки в його налаштуваннях, далі ми беремо цей фізичний матеріал і додаємо його до матеріалу персонажа та на матеріали які ми використовуємо в проекті, після додавання всіх звуків налаштовуємо їх вказуючи радіус дії кожного звуку.

Створення годинника в реальному часі [21]

Для створення нашого годинника ми будемо використовувати Blueprints та нодову систему для того щоб ми могли відображати реальний час такий як і на вашому ПК в реальному часі, та додати нашому проекту більше реалістичності.

Для початку ми створили модель годинника в Blender та імпортували її в Unreal Engine 5.



Рис. 2.19. Готова модель годинника Національного університету «Острозька академія» в програмі Unreal Engine 5

Джерело: розроблено автором

Після того як модель була готова до використання було зроблено стрілки окремими частинами щоб можна було відображати час.

Як тільки ми створили вже готову модель з якою можна працювати ми перейшли до інтерактивності, а саме перейшли до створення нодів для реалізації відображення часу. Але для початку ми привязали наші стрілки до незалежних Arrow для визначення центру годинника навколо якого будуть рухатись стрілки.

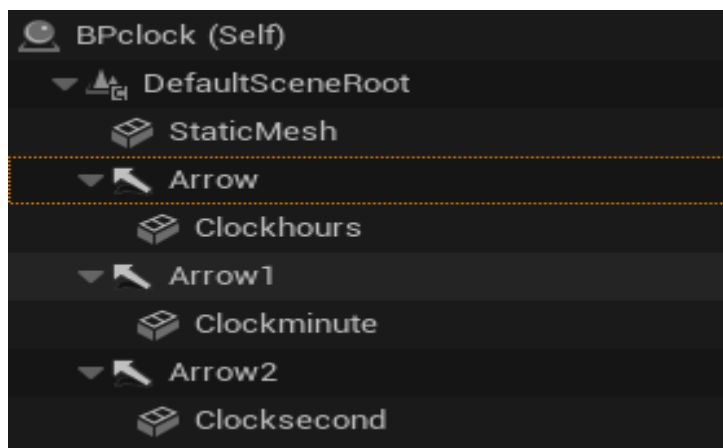


Рис. 2.20. Прив'язування стрілок до об'єкта в Unreal Engine 5

Джерело: розроблено автором

Після того як стрілки були прив'язані до Arrow створюємо ноди для того щоб наші стрілки могли рухатись і вказувати час.

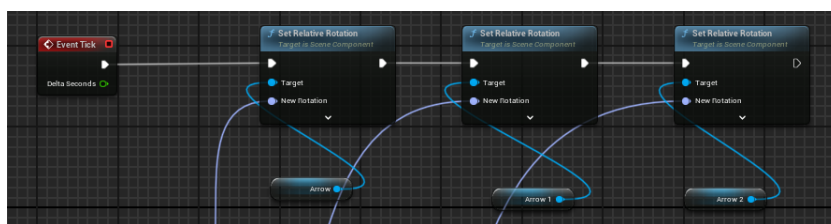


Рис. 2.21. Створення нодів в Unreal Engine 5

Джерело: розроблено автором

Прив'язавши всі стрілки в нодах до компонента "Set Relative Rotation" у вузлі Blueprints в Unreal Engine який нам дозволяє встановити відносний поворот (орієнтацію) об'єкта відносно його батьківського об'єкта або локальну орієнтацію об'єкта.

Коли ми використовуємо цей компонент, ми вказуємо нові значення кутів руху по кожній з трьох осей - X, Y та Z. Ці значення вказують, наскільки об'єкт повернеться відносно його поточної орієнтації на кожній з цих осей.

Загалом, цей компонент дозволяє динамічно змінювати орієнтацію об'єкта в реальному часі відносно його поточної орієнтації, що є корисною функцією при роботі з анімацією, фізичними ефектами або будь-якою ситуацією, де потрібно змінювати орієнтацію об'єкта у відношенні до його батьківського об'єкта або власної локальної орієнтації.

Після прив'язування ми перейшли до встановлення градусів за якими мають повертатись стрілки.

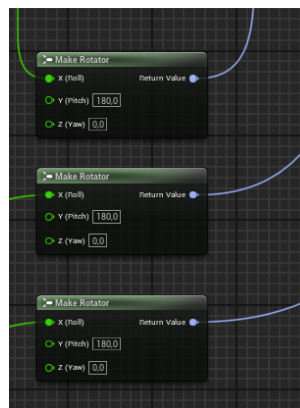


Рис. 2.22. Створення нодів в Unreal Engine 5

Джерело: розроблено автором

Ми вказали по якій осі мають бути закріплені стрілки годинника за допомогою компонента "Make Rotation" у вузлі Blueprints в Unreal Engine який дозволяє створювати об'єкт, який представляє поворот (орієнтацію) у тривимірному просторі.

Коли ми використовуємо цей компонент, ми вказуємо значення кутів повороту по кожній з трьох осей - X, Y та Z. Ці значення вказують, наскільки об'єкт буде повернутий по кожній з цих осей. Цей компонент генерує об'єкт повороту з вказаними значеннями кутів повороту.

Загалом, цей компонент дозволяє створювати об'єкт, що представляє конкретний поворот у тривимірному просторі, що може бути корисним при

роботі з анімацією, фізичними ефектами, розміщенням об'єктів у грі та в інших ситуаціях, де потрібно контролювати орієнтацію об'єктів.

Далі ми перейшли до створення нодів які відповідають за правильне відображення часу і за якими відбувається рух стрілок відносно того часу який відображає цей нод

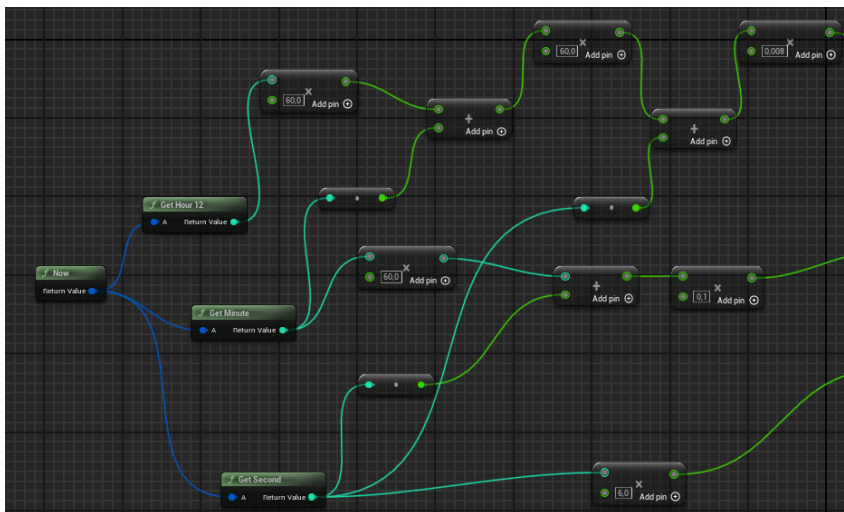


Рис. 2.23. Створення нодів в Unreal Engine 5

Джерело: розроблено автором

Тут ми обраховуємо час, за який має певна стрілка, зрушити навколо своєї осі на певну кількість градусів.

І ось така загальна картина у нас виходить при створенні готової нодової системи для відображення годинника в реальному часі.

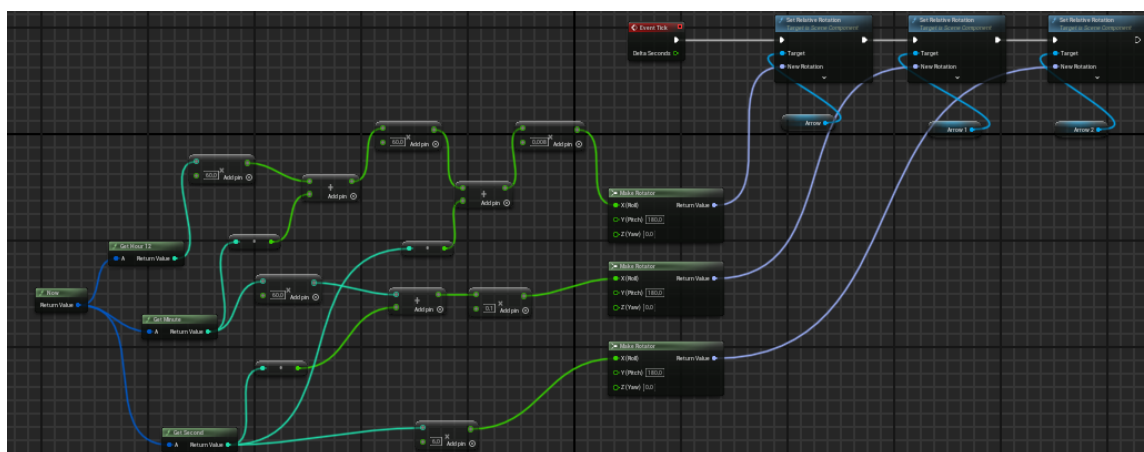


Рис. 2.24. Створення нодів в Unreal Engine 5

Джерело: розроблено автором

Після реалізації усіх цих частин ми отримуємо годинник який відображає час із вашого ПК в реальному часі.

Ось такий результат у нас вийшов:

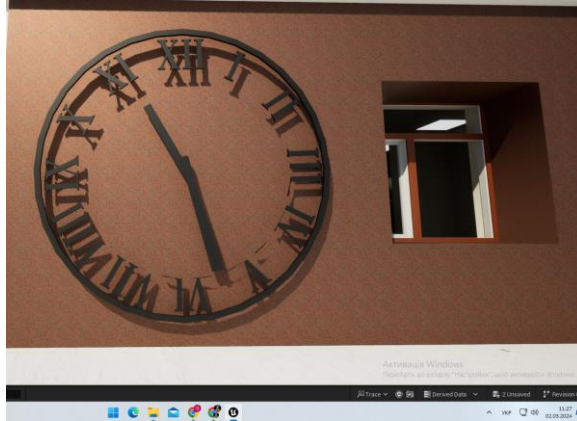


Рис. 2.25. Візуалізація годинника в реальному часі в Unreal Engine 5

Джерело: розроблено автором

Як видно на скріншоті годинник в грі показує 11 годин 27 хвилин і у мене на ПК час в реальному часі 11:27 , отже годинник справний і відмінно працює та виконує свою роль у нашому проекті.

Механіка створення шлагбаума

Для початку створення шлагбаума ми почали з моделювання двох рухомих деталей, після того як деталі були готові ми приступили до інтерактивності.

Спочатку ми зробили з готової моделі Blueprints та перейшли до створення нодів

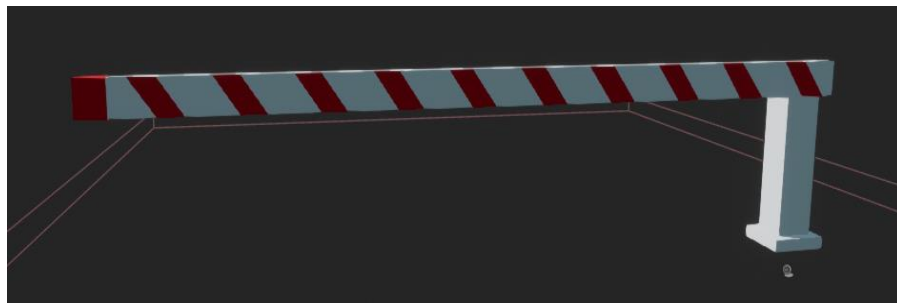


Рис. 2.26. Blueprints шлагбаума в Unreal Engine 5

Джерело: розроблено автором

Ноди нам потрібні щоб створити анімацію рухів відкривання і закривання шлагбаума та розробити логіку управління ним.

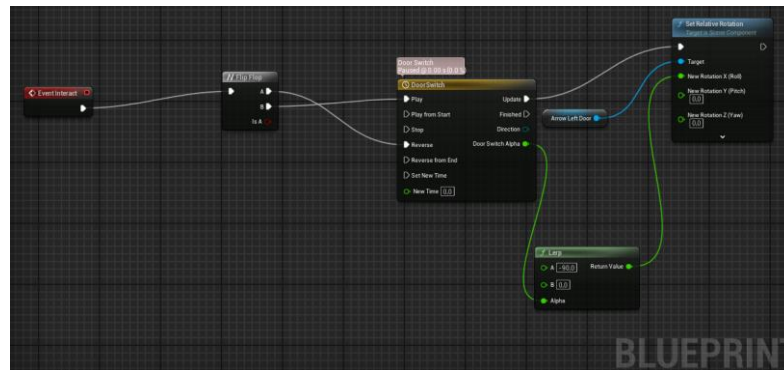


Рис. 2.27. Нод для реалізації відкриття шлагбаума в Unreal Engine 5

Джерело: розроблено автором

За допомогою цієї нодової системи ми можемо керувати відкриванням, а також змінювати його функціонал з часом.

У цьому ноді ми використали такі компоненти як:

Компонент "Flip Flop" у вузлі Blueprints в Unreal Engine дозволяє змінювати вихідне значення між двома станами (наприклад, увімкнено/вимкнено або правда/хиба) кожного разу, коли він активується. Цей компонент дозволяє переключати між двома станами при кожному виклику, зберігаючи останній використаний стан у пам'яті.

Коли сигнал подається на вхід "Flip Flop", компонент змінює свій вихідний стан. Наприклад, якщо спочатку вихідний стан був "False", після першого активації він зміниться на "True". При наступному активації він знову зміниться на протилежний стан, тобто "False". Цей процес може продовжуватися, переключаючи між двома станами кожен раз, коли компонент активується.

Компонент "Lerp" (Linear Interpolation) у вузлі Blueprints використовується для плавного переходу між двома значеннями за допомогою лінійної інтерполяції.

Цей компонент використовується для створення анімаційних ефектів, плавних переходів між позиціями об'єктів, зміни кольорів та інших випадків, де потрібно плавно переходити від одного значення до іншого.

"Set Relative Rotation" у вузлі Blueprints в Unreal Engine який нам дозволяє встановити відносний поворот (орієнтацію) об'єкта відносно його батьківського об'єкта або локальну орієнтацію об'єкта.

Механіка створення погодних умов

В Unreal Engine 5 механіка створення погодних умов зазвичай використовується за допомогою спеціальних систем та інструментів, які дозволяють реалістично моделювати різноманітні погодні явища. Основні компоненти та механіки, які можуть бути використані для цього, включають:

Динамічні погодні ефекти: Unreal Engine 5 надає можливість динамічно змінювати погодні умови у грі в реальному часі. Це означає, що ми можемо створювати різні погодні явища, такі як дощ, хмари тощо, і динамічно контролювати їх розповсюдження та інтенсивність у грі.

Система часток: Unreal Engine 5 має потужну систему часток, яка дозволяє створювати різні візуальні ефекти, включаючи дощ, блискавки, опади тощо. Ця система дозволяє нам контролювати розподіл та рух часток у грі, що дозволяє створювати реалістичні погодні умови.

Звукові ефекти: Додавання звукових ефектів до погодних умов може підвищити реалізм середовища. Наприклад, звуки дощу, грому, шуму вітру можуть допомогти створити атмосферу певних погодних умов.

Динамічне освітлення: Освітлення гри може бути динамічно змінюваним залежно від погодних умов. Наприклад, під час дощу або хмарного дня освітлення може бути більш приглушеним, а під час ясної погоди - яскравішим та насиченішим. Це додає реалізму та атмосферності до погодних умов у грі.

Для реалізації погодних умов в Unreal Engine 5 ми використовуємо Blueprints та нодову систему для створення реалістичних погодних умов.

Для того щоб створити ефект чистого неба ми створили Blueprints та підключили до нього матеріал неба в день і вночі і ось такий результат у нас вийшов



Рис. 2.28. Візуалізація чистого небо в Unreal Engine 5

Джерело: розроблено автором

Для створення хмарного неба ми використовуємо блюпрінт який робили раніше але тепер за допомогою Blender ми створили матеріал хмар та експортували його в Unreal Engine 5 потім додали його в Blueprints та за допомогою нодів підключили хмари до неба.

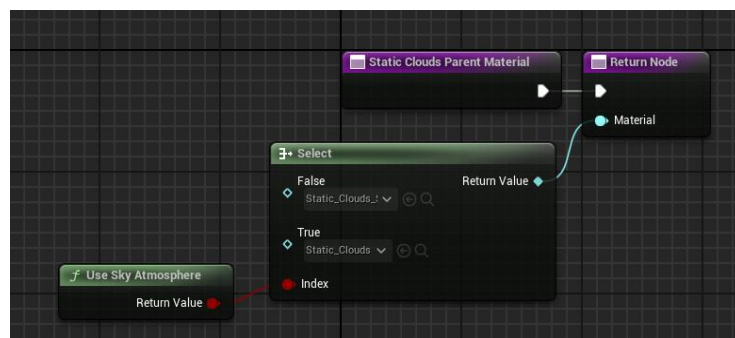


Рис. 2.29. Нод до якого підключені матеріали хмар в Blueprints в Unreal Engine 5

Джерело: розроблено автором

Результат підключення хмар:



Рис. 2.30. Візуалізація хмарно неба в Unreal Engine 5

Джерело: розроблено автором

Для створення дощу [22] ми використовуємо теж саме небо, але окремо ще додаємо краплі, які генеруються рандомно і звук який вмикається під час старту блюпринта після додавання і налаштування

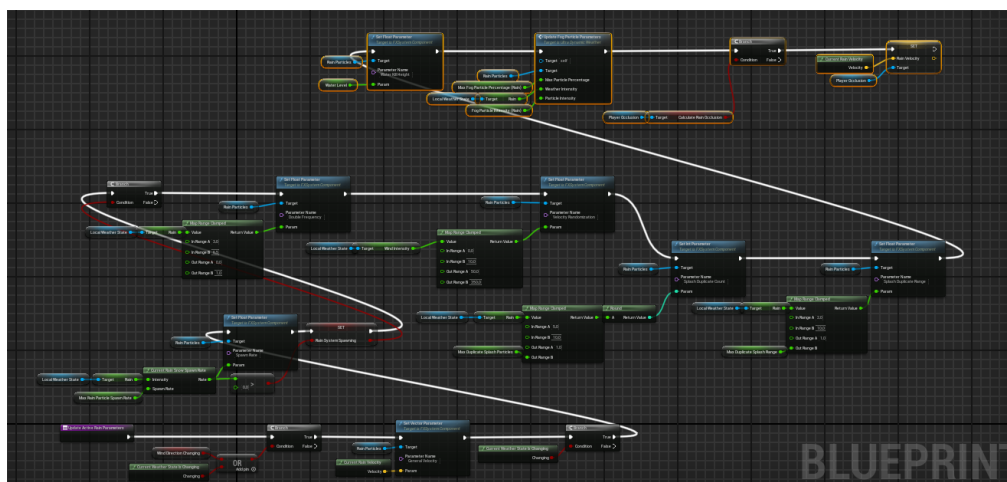


Рис. 2.31. реалізація дощу через нодову систему в Unreal Engine 5

Джерело: розроблено автором

За допомогою цього ноду ми можемо робити плавний перехід з хмарної погоди в дощ, адже цей нод при запуску робить небо похмурим, та запускає спавн крапель дощу в рандомному порядку.

Ось кілька важливих компонентів які ми використовували для створення нода:

Компонент "Current Rain Velocity" у вузлі Blueprints в Unreal Engine дозволяє отримати поточну швидкість опадів дощу в заданій точці у вашій грі або сцені.

Цей компонент корисний для реалізації різних логічних умов або ефектів, які можуть залежати від інтенсивності дощу в певних областях гри. Наприклад, ви можете використовувати поточну швидкість опадів дощу для налаштування реакції персонажів або об'єктів на дощ. Ви також можете використовувати її для зміни візуальних ефектів або звукових ефектів, що пов'язані з дощем, в залежності від того, наскільки сильно дощить у певній області гри.

Отже, "Current Rain Velocity" надає можливість динамічно реагувати на зміни у погодних умовах, зокрема на інтенсивність опадів дощу, що дозволяє створювати більш реалістичні та захоплюючі віртуальні світи.

Компонент "Map Range Clamped" в ноді в Unreal Engine дозволяє змінювати діапазон значень чисел з одного діапазону в інший, з обмеженням значень у новому діапазоні.

Цей компонент особливо корисний, коли нам потрібно перетворити значення з одного діапазону на інший, забезпечуючи при цьому обмеження для вихідного діапазону. Наприклад, ми можете використовувати "Map Range Clamped" для перетворення значень від 0 до 100 в значення від 0 до 1, або навпаки, забезпечуючи тим самим, щоб значення не виходили за межі нового діапазону.

Цей компонент використовується для різних цілей, таких як налаштування параметрів відображення, анімації, фізики, звуку та багатьох інших. Саме нам він допомагає керувати інтенсивністю дощу.

Компонент "Set Int Parameter" у вузлі Blueprints в Unreal Engine призначений для встановлення цілочисельного параметра для об'єкту, зазвичай в контексті матеріалу чи частинок наприклад як краплі дощу.

Цей компонент дозволяє динамічно змінювати цілочисельні параметри в реальному часі в залежності від певних умов або взаємодії з іншими елементами гри. Наприклад, ви можете використовувати "Set Int Parameter" для зміни інтенсивності світла, яскравості кольорів, рівня деталізації чи інших атрибутів матеріалу в реальному часі, що дозволяє створювати різноманітні візуальні ефекти та анімації.

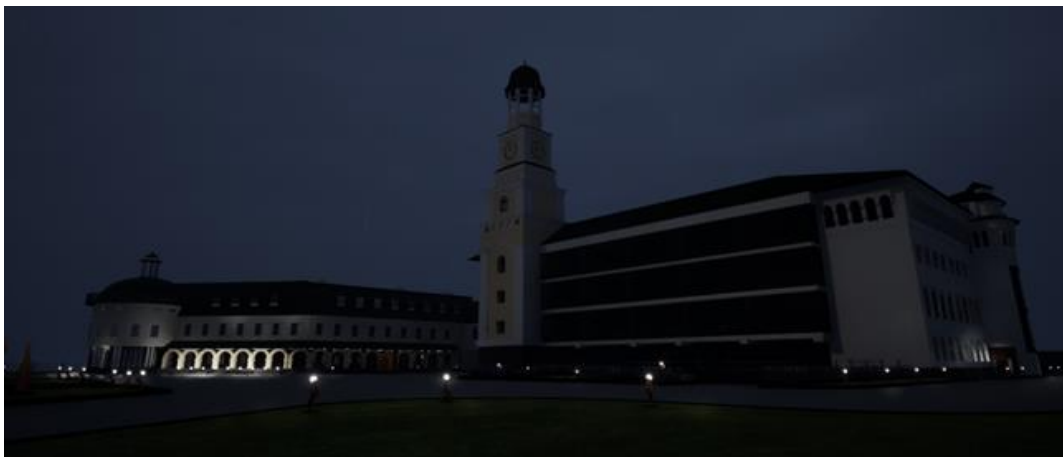


Рис. 2.32. Візуалізація дощу в Unreal Engine 5

Джерело: розроблено автором

Для створення грози в Unreal Engine 5 ми використовували матеріали хмар і блискавок, які через нодову систему підключали до Blueprints, та зациклювали їх і вказували рандомне значення і спавн, а для поєднання блискавки хмар і дощу,

ми використали один загальний нод в Blueprints, щоб реалізувати це все, як один елемент а саме Грозу .

Для імітації блискавки ми реалізовували рандомним відображенням зображення блискавки яке супроводжується звуковими ефектами і зміною сили світіння джерела світла яке окремо було додано для того щоб імітувати блискавку, далі підключаємо до нодової система, яка буде керувати появами блискавок в ігровому часі під час грози.

Ось що у нас виходить:

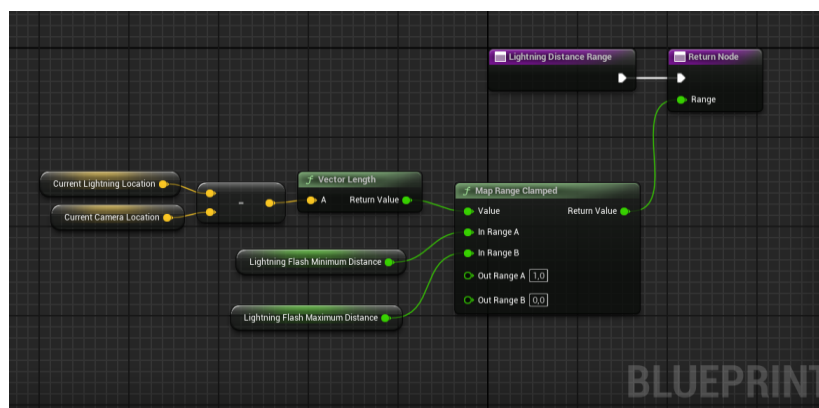


Рис. 2.33. Блискавка в нодовій системі в Unreal Engine 5

Джерело: розроблено автором

Після реалізації блискавки, додавання до вузла з дощем може включати декілька додаткових кроків для створення реалістичної грози.

Збільшення інтенсивності дощу: Після блискавки можна збільшити інтенсивність дощу, щоб створити враження посилення опадів після грому. Це може бути досягнуто збільшенням значення параметра інтенсивності дощу у відповідному вузлі чи системі погоди.

Збільшення кількості генерації крапель: Щоб підсилити ефект дощу після грому, можна збільшити кількість генерації крапель або частоту появи крапель. Це дозволить створити враження посилення опадів після грому та відтворити реалістичну атмосферу грози.

Збільшення захмарення неба: Після блискавки можна також збільшити захмарення неба для створення враження затяжного дощу після грому. Це може

включати зміну текстури неба, додавання хмар або зміну освітлення, щоб створити темнішу та загрозливішу атмосферу.

Додавання звукових ефектів: Для підсилення враження грози важливо додати відповідні звукові ефекти, такі як грім, грохіт дощу та вітер. Ці звукові ефекти допоможуть створити атмосферу напруження та доповнити візуальні ефекти грози.

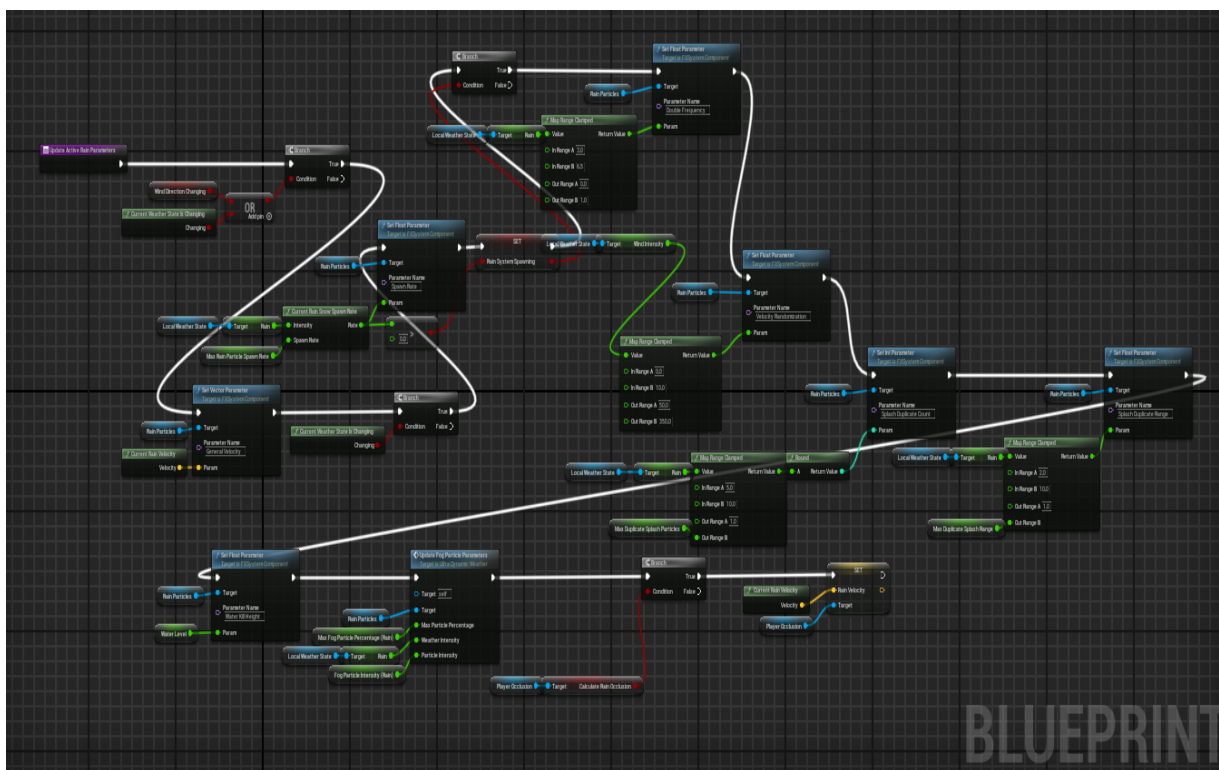


Рис. 2.34. Гроза в нодовій системі в Unreal Engine 5

Джерело: розроблено автором

Отже, реалізація грози включає не лише створення блискавки, але й налаштування різних параметрів дощу та неба, а також додавання відповідних звукових ефектів для створення реалістичної та захоплюючої атмосфери грози у нашому проєкті.



Рис. 2.35. Візуалізація грози в Unreal Engine 5

Джерело: розроблено автором

Після створення нодів і Blueprints додаємо його у наш проект і в результаті, у нас виходить гроза з блискавками і сильним дощем, яка супроводжується звуками дощу і грому.

ВИСНОВКИ

В рамках даного дослідження було вивчено та проаналізовано можливості використання віртуальних технологій для створення інтерактивного навчального середовища для Національного університету "Острозька Академія". Практичний етап дослідження включав аналіз віртуальних інструментів та платформ, розробку концепції інтерактивного віртуального середовища та впровадження конкретних інноваційних рішень.

Зокрема, було розглянуто такі аспекти, як звуки та атмосфера, відкриття шлагбаума, реалізація годинника в реальному часі та погодні ефекти в Unreal Engine 5. Ці функції були обрані з урахуванням їх можливості підвищити реалізм навчального середовища та зробити навчання більш захоплюючим та ефективним для студентів.

Під час практичного застосування віртуальних технологій для створення навчального середовища було виявлено, що їх використання дозволяє створити неперевершене навчальне середовище, яке забезпечує студентам можливість активної взаємодії та залучення до навчального процесу.

У результаті практичної реалізації вищезгаданих інноваційних рішень було досягнуто високого реалізму та ефективності навчального середовища, що сприяє покращенню якості освіти.

Застосування платформи Jira для управління проектом виявилось додатковим важливим кроком у забезпеченні успіху проекту. Це дозволило зберігати прозорість та структурованість управління завданнями, що сприяло ефективному вирішенню завдань та своєчасній реакції на зміни. Розгортання проекту в системі Jira додало нові можливості для співпраці, комунікації та координації всіх учасників команди, підсилюючи позитивний вплив використання віртуальних технологій на освітній процес.

Отже, на практичному рівні використання віртуальних технологій для створення інтерактивного навчального середовища для Національного університету "Острозька Академія" є дієвим та перспективним підходом, що відкриває нові можливості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Blender URL: <http://surl.li/uemqv> (дата звернення: 01.12.2023 р.).
2. Unity URL : <http://surl.li/tytvs> (дата звернення: 06.12.2023 р.).
3. Unreal Engine URL: <http://surl.li/uemrf> (дата звернення: 08.12.2023 р.).
4. Використання механізмів LOD URL: <http://surl.li/uemrj> (дата звернення: 15.12.2023 р.).
5. Використання Nanite та Lumen URL: <http://surl.li/uemrk> (дата звернення: 25.12.2023 р.).
6. MetaHuman Creator URL: <http://surl.li/uemrs> (дата звернення: 05.01.2024 р.).
7. Jira URL: <http://surl.li/uemrw> (дата звернення: 11.01.2024 р.).
8. Blueprints в Unreal Engine 5 URL: <http://surl.li/uemrz> (дата звернення: 14.01.2024 р.).
9. Макроси (Macros) URL: <http://surl.li/uemse> (дата звернення: 21.01.2024 р.).
10. Jira Software URL: <http://surl.li/uemsh> (дата звернення: 12.12.2023 р.).
11. Photoshop URL: <http://surl.li/uemsm> (дата звернення: 26.01.2024 р.).
12. Illustrator URL: <http://surl.li/uemso> (дата звернення: 27.01.2024 р.).
13. Booltool URL: <http://surl.li/uemsq> (дата звернення: 29.01.2024 р.).
14. Налаштування UV-розгортки URL: <http://surl.li/uemst> (дата звернення: 02.02.2024 р.).
15. Інтеграція 3D моделей URL: <http://surl.li/uemtd> (дата звернення: 08.02.2024 р.).
16. Використання Material Editor в Unreal Engine URL: <http://surl.li/uemth> (дата звернення: 12.02.2024 р.).
17. Оптимізація та тестування URL: <http://surl.li/uemtl> (дата звернення: 15.02.2024 р.).
18. Робота з файлами FBX URL: <http://surl.li/uemtn> (дата звернення: 20.02.2024 р.).
19. "Complex Collision" "Simple Collision" URL: <http://surl.li/uemto> (дата звернення: 25.02.2024 р.).
20. Physical Material URL: <http://surl.li/uemtw> (дата звернення: 02.03.2024 р.).

21. Створення годинника в реальному часі URL: <http://surl.li/tytwu> (дата звернення: 10.03.2024 р.).
22. Реалізація дощу URL: <http://surl.li/uemtz> (дата звернення: 15.03.2024 р.).
23. 3D Viewport URL: <http://surl.li/uemug> (дата звернення: 19.03.2024 р.).
24. Materials URL: <http://surl.li/uemul> (дата звернення: 23.03.2024 р.).
25. Adobe After Effects URL: <http://surl.li/uemup> (дата звернення: 28.03.2024 р.).
26. Datasmith URL: <http://surl.li/tytys> (дата звернення: 05.04.2024 р.).
27. World Partition URL: <http://surl.li/uemur> (дата звернення: 10.04.2024 р.).
28. Інтеграція Blueprints з C++ URL: <http://surl.li/uemuy> (дата звернення: 18.04.2024 р.).
29. Реалізація штучного інтелекту (AI) URL: <http://surl.li/tytxl> (дата звернення: 19.04.2024 р.).
30. Виклик функцій URL: <http://surl.li/tytyg> (дата звернення: 10.05.2024 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: Змодельуйте всі відкриті простори, ландшафти і рельєф в межах кампусу, щоб відобразити топографію місцевості. Це включає в себе моделювання рослинності, такої як дерева, кущі та квіти, а також твердого покриття, такого як доріжки, тротуари та дороги.

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

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

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

Кожен виконавець проекту має свою зону відповідальності:

● Новий корпус

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

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

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

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

3. Лаптев Богдан(фасад адмінбудівлі, частина 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 клавiші;
- 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 сьйва.

● 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 номера.