

**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 академічної бібліотеки)»

**Виконала:** студентка 4 курсу, групи КН-41  
першого (бакалаврського) рівня вищої освіти  
спеціальності 122 Комп'ютерні науки  
освітньо-професійної програми «Комп'ютерні науки»  
*Черевко Вікторія Ігорівна*

**Керівник:** старший викладач  
*Клебан Юрій Вікторович*

**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

Черевко Вікторії Ігорівни

**1. Тема роботи: Створення інтерактивного віртуального середовища НаУОА (інтер'єр академічної бібліотеки).**

*керівник роботи: Клебан Ю.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 3Ds Max (для моделювання інтер'єру та всіх елементів його наповнення) Unreal Engine 5 ( для віртуалізації та створення інтерактивних елементів ) Adobe Photoshop (для ретуші рендерів).

**4. Перелік завдань, які належить виконати:** створення інтер'єру та всіх елементів наповнення наукової бібліотеки в 3D редакторі, перенесення 3D сцени на ігровий рушій та її налаштування, розробка інтерактивної складової, а саме телефонної системи.

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

*6. Консультанти розділів роботи:*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Клебан Ю.В.	01.12.2023	01.12.2023
2	Клебан Ю.В.	01.12.2023	01.12.2023

7. Дата видачі завдання: 01.12.2023 р.

### КАЛЕНДАРНИЙ ПЛАН

	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
	Затвердження теми роботи/проекту	до 31.10.2023 р.	
	Постановка технічного завдання	до 01.12.2023 р.	
	Ознайомлення з документацією	до 10.12.2023 р.	
	Написання розділу 1	до 01.02.2024 р.	
	Написання розділу 2	до 01.04.2024 р.	
	Тестування системи	до 20.04.2024 р.	
	Виправлення помилок	до 01.05.2024 р.	
	Попередній захист та перевірка на рівень унікальності кваліфікаційної роботи/проекту	до 31.05.2024 р.	
	Здача кваліфікаційної роботи/проекту на кафедрі	31.05.2024 р.	

Студент: \_\_\_\_\_ Вікторія ЧЕРЕВКО  
( підпис )

Керівник кваліфікаційної роботи: \_\_\_\_\_ Юрій КЛЕБАН

# АНОТАЦІЯ

## кваліфікаційної роботи

### на здобуття освітнього ступеня бакалавра

**Тема:** Створення інтерактивного віртуального середовища НаУОА (інтер'єр академічної бібліотеки).

**Автор:** Черевко Вікторія Ігорівна

**Науковий керівник:** Клебан Ю.В., викладач кафедри ЕММІТ.

Захищена «.....»..... 20\_\_ року.

**Пояснювальна записка до кваліфікаційної роботи:** 64 с., 34 рис., 0 табл., 3 додатки, 30 джерел.

**Ключові слова:** 3D моделювання, текстури, пресети, рендер, 3Ds Max, технології, автоматизація, Unreal Engine 5, інтерактивний, віртуальний.

#### **Короткий зміст праці:**

В цій кваліфікаційній роботі представлено процес розробки інтерактивного віртуального середовища Національного університету Острозька академія (інтер'єр академічної бібліотеки). Було зібрано необхідну інформацію, таку як креслення будівлі та реальні фото інтер'єру. В середовищі 3Ds Max було реалізовано 3D сцену яка повністю відтворює інтер'єр бібліотеки. Перенісши сцену на ігровий рушій Unreal Engine 5 було додано інтерактивну частину проекту, а саме телефонну систему а також персонажа. В результаті було отримано функціональне інтерактивне віртуальне середовище з яким може взаємодіяти гравець.

**Keywords:** 3D modeling, textures, presets, rendering, 3Ds Max, technologies, automation, Unreal Engine 5, interactive, virtual.

#### **Summary of the work:**

This qualification work presents the process of developing the interactive virtual environment of the National University of Ostroh Academy (interior of the academic library). The necessary information was collected, such as building drawings and real photos of the interior. A 3D scene was created in 3Ds Max to fully recreate the library interior. Transferring the scene to the Unreal Engine 5 game engine, we added the interactive part of the project, namely the telephone system and the character. The result was a functional interactive virtual environment with which the player can interact.

(підпис автора)



## ЗМІСТ

<b>ВСТУП.....</b>	<b>5</b>
<b>РОЗДІЛ 1</b>	
<b>Теоретичні основи створення віртуального середовища НаУ ОА (інтер'єр академічної бібліотеки).....</b>	<b>7</b>
1.1. Постановка проблеми, опис базових понять процесу.....	7
1.2. Аналіз аналогічних розробок.....	7
1.3. Управління проєктом.....	9
<b>РОЗДІЛ 2</b>	
<b>Створення віртуального середовища НаУ ОА (інтер'єр академічної бібліотеки).....</b>	<b>12</b>
2.1. Опис технологічного стеку.....	12
2.2. Збір інформації та її підготовка.....	13
2.3. Створення 3Д моделей в 3Ds Max.....	16
2.4. Інтеграція 3Д моделей в Unreal Engine.....	40
2.5. Інтерактивна складова віртуального середовища в Unreal Engine.....	48
2.6. Можливості для розвитку проєкту та його перспективи.....	51
<b>ВИСНОВКИ.....</b>	<b>53</b>
<b>Список використаних джерел.....</b>	<b>54</b>
<b>Додатки.....</b>	<b>57</b>

## ВСТУП

**Тема:** Створення інтерактивного віртуального середовища НаУОА (інтер'єр академічної бібліотеки).

**Завдання:** створити інтерактивне віртуальне середовище, відтворивши інтер'єр академічної бібліотеки.

Метою дослідження є розробка та реалізація віртуального середовища інтер'єру академічної бібліотеки НаУОА. Зокрема, ми прагнемо створити віртуальну модель, яка буде максимально функціональною та реалістичною, забезпечуючи студентам, працівникам та гостям академії унікальний досвід взаємодії з простором бібліотеки.

Об'єктом дослідження є створення віртуального середовища інтер'єру академічної бібліотеки НаУОА.

Предметом дослідження є визначення технічних та дизайнерських аспектів створення віртуального інтер'єру, робота включає в себе моделювання об'єктів, розробку текстур, програмування поведінки об'єктів та реалізацію ефективного інтерфейсу для користувачів.

Актуальність даної проблеми. В сучасному освітньому середовищі, де всі намагаються поєднувати традиційні методи навчання та інноваційні технології, створення віртуальних середовищ для навчальних закладів є просто необхідністю. Академічна бібліотека виступає як центр знань та навчання, тому віртуальне відтворення її інтер'єру може стати ефективним інструментом для зручного та цікавого вивчення простору університету. Особливо цей інструмент буде корисним для першокурсників на початку навчання та може використовуватися на Днях відкритих дверей для того аби максимально інтерактивно показати всю красу академії абітурієнтам або ж представлена як ще одна крута особливість навчального закладу під час різних екскурсій.

## РОЗДІЛ 1

### Теоретичні основи створення віртуального середовища НаУ ОА (інтер'єр академічної бібліотеки)

#### 1.1. Постановка проблеми, опис базових понять процесу

Технології віртуальної реальності та 3D застосовуються у дуже різних сферах, зокрема це дизайн інтер'єрів. Перехід до сучасних технологій спонукає до створення віртуальних середовищ, якими користувач може вільно пересуватись та взаємодіяти з ними.

Однією з ключових проблем при створенні віртуальних середовищ, зокрема інтер'єрів, є забезпечення користувачам максимально комфортної та зручної взаємодії з об'єктами. Сюди можна віднести можливість вільно рухатись приміщенням, а також максимально реалістичне сприйняття об'єктів наповнення інтер'єру.

У цьому контексті розглядаються базові поняття процесу віртуального 3D середовища, зокрема аспекти моделювання об'єктів та програмування поведінки об'єктів. Можна розглядати також проблему оптимізації для забезпечення плавності та швидкості віртуальної взаємодії.

#### 1.2. Аналіз аналогічних розробок

Існує ряд віртуальних платформ та середовищ, що дозволяють користувачам взаємодіяти з 3D-моделями інтер'єрів. Один із прикладів цього - платформа **Sketchfab**[1]. Розглянемо основні аспекти та характеристики.

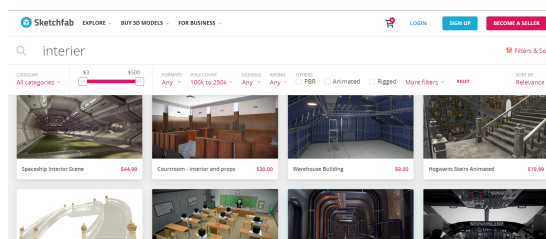


Рис. 1.1 - Зовнішній вигляд платформи Sketchfab

*Джерело: розроблено автором*

### **Функціональні можливості платформи Sketchfab:**

1. **Віртуальна навігація:** схоже на те, що наш проект передбачає. Користувачі на Sketchfab можуть вільно пересуватись всередині 3D-сцени, повертаючи та нахиляючись для отримання кращого огляду.

2. **Масштабування та панорамування:** здійснення зуму, а також можливість обертати та панорамувати камеру, дозволяє користувачам детально розглядати окремі об'єкти в інтер'єрі.

### **Реалізація взаємодії:**

- **Моделі зі Sketchfab:** велика кількість 3D-моделей доступна на платформі, і користувачі можуть взаємодіяти з ними, вивчаючи їхні деталі та розташування у віртуальному просторі (Рис. 1.2).



Рис.1.2 - Приклад вигляду моделі інтер'єру костелу на платформі Sketchfab

*Джерело: розроблено автором*

### **Технічні аспекти:**

- **Підтримка форматів:** Sketchfab підтримує різноманітні формати 3D-моделей, що робить його універсальним інструментом для співпраці з різними джерелами контенту.

- **Сумісність із VR:** деякі моделі можуть бути переглянуті в режимі віртуальної реальності, забезпечуючи більш глибокий та імерсивний досвід.

#### **Сильні та слабкі сторони:**

- **Сильна сторона:** широкий вибір моделей, легкий доступ та можливість спільної роботи.
- **Слабка сторона:** обмежена можливість створення власних інтер'єрів, адже платформа здебільшого спрямована на обмін готовими 3D-моделями.

Аналогічні розробки, зокрема моделі на Sketchfab, підтверджують успішну реалізацію віртуальної навігації та взаємодії з 3D-сценами. Однак, для нашого проекту буде важливо покращити можливості взаємодії користувача з ними, забезпечуючи максимальну гнучкість та реалізм, а також в подальшому забезпечити можливість додати елементи інтерактивної взаємодії між користувачем та об'єктами наповнення інтер'єру.

Зазначимо, що технології віртуальної реальності та 3D є актуальними для дизайну інтер'єрів, і вони спонукають до створення віртуальних середовищ, які користувач може вільно взаємодіяти. Однією з ключових проблем є забезпечення максимально комфортної та зручної взаємодії з об'єктами інтер'єру.

Аналізуючи аналогічні розробки, ми звернули увагу на платформу Sketchfab, яка дозволяє взаємодіяти з 3D-моделями інтер'єрів. Описали функціональні можливості, технічні аспекти, а також сильні та слабкі сторони цієї платформи. Підкреслили, що для нашого проекту важливо покращити можливості взаємодії користувача з віртуальним середовищем, забезпечуючи максимальну гнучкість та реалізм.

### **1.3. Управління проектом**

Під час роботи над різними типами проектів дуже важливою частиною є правильна організація роботи, для цього можуть застосовувати різні інструменти управління проектами.

Інструменти управління проектами - це програми та системи, які допомагають керівникам проектів та виконавцям ефективно планувати та контролювати роботу на всіх етапах. Вони надають централізовану платформу для відстеження часу, завдань та ресурсів. В процесі роботи над нашим проектом ми використовували Jira [2].

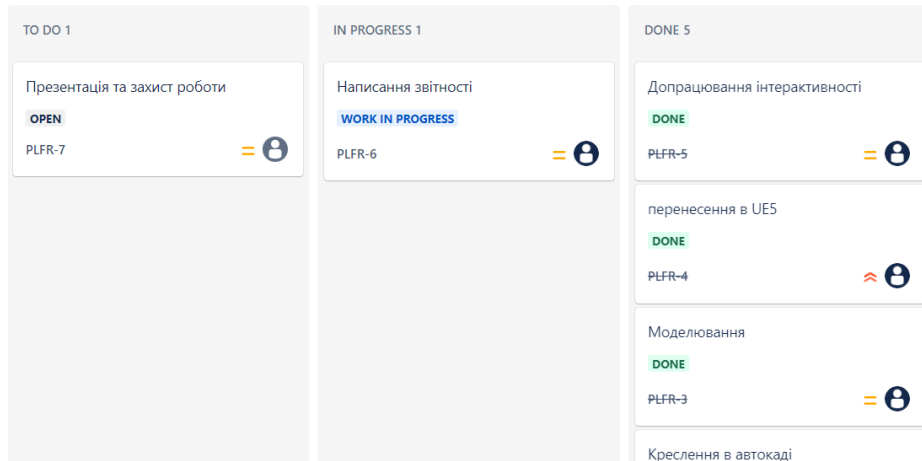


Рис.1.3 - Проект в Jira

*Джерело: розроблено автором*

Jira - це інтуїтивно зрозуміла та гнучка платформа для управління проектами, вона дає можливість ефективно спланувати, відстежувати та керувати робочими процесами та проектами. Вона була розроблена компанією Atlassian ще у 2002 році[3]. Jira постійно оновлюється та надає дуже багато можливостей для управління робочим процесом.

Переваги використання:

- Прозорість: ідеальний інструмент для відстежування прогресу, помилок та повної прозорості в робочому процесі.
- Співпраця: можливість безперебійно працювати в команді використовуючи функції спілкування.
- Гнучкість: має багато функцій, такі як спринти для управління Scrum та інші гнучкі дошки, наприклад, Kanban[4].

Як працює Kanban у Jira:

1. **Створення Kanban проекту**
2. **Налаштування робочого процесу:** Jira пропонує стандартний робочий процес з колонками: Backlog, Selected for Development, In Progress, та Done.
3. **Додавання завдань до беклогу:** завдання, помилки або користувацькі історії додаються до беклогу, який є першою колонкою на Kanban дошці.

Загалом Jira як ефективний інструмент для управління проектами має багато переваг та суттєво спрощує процес роботи.

## РОЗДІЛ 2

### Створення віртуального середовища НаУ ОА (інтер'єр академічної бібліотеки)

#### 2.1. Опис технологічного стеку

Під час відтворення інтер'єру академічної бібліотеки НаУ ОА використовувалось наступне програмне забезпечення:

- **Autodesk AutoCAD** [5] — програмний продукт, основною функцією якого є створення 2D та 3D креслень, проектувань. Використовували для відтворення отриманих креслень у dwg форматі для подальшого моделювання.
- **3Ds Max** [6] — програмний продукт, функцією якого є 3D моделювання, анімація, рендеринг та візуалізація об'єктів. Використовували для моделювання приміщення та всього наповнення і інтер'єру академічної бібліотеки та для створення реалістичних рендерів.
- **V-Ray** [7] — програмне забезпечення, яке використовується для рендерингу та комп'ютерної візуалізації, використовувався як плагін в 3Ds Max для створення високоякісних рендерів інтер'єру. Використовували для створення та налаштування матеріалів та рендерингу проекту.
- **Плагін DataSmith** [8] — призначений для експорту проекту з різних 3D середовищ на ігровий рушій Unreal Engine. Використовували для перенесення сцени з 3Ds Max в середовище Unreal Engine.
- **Unreal Engine** [9] — ігровий рушій комп'ютерної 3D графіки, дозволяє створювати віртуальні інтерактивні середовища, відеоігри. Використали як платформу на якій розміщено віртуальне інтерактивне середовище НаУ ОА, та яка дозволяє забезпечити взаємодію між проектом та користувачем.
- **Adobe Photoshop** [10] — графічний редактор, призначений для роботи з векторною та растровою графікою. Використовувався для створення текстур, постобробки рендерів, кольорокорекції зображень.



- **Blender** [11] — програмне середовище призначене для моделювання тривимірних об'єктів, створення візуалізацій та анімації. Використовували для відкриття проекту з екстер'юром академічної бібліотеки для подальшої звірки в розмірах та плануванні.

## 2.2. Збір інформації та її підготовка

Першим етапом нашої роботи був збір та підготовка інформації. Для цього було зроблено багато фотознімків та декілька відеороликів внутрішньої та зовнішньої частини академічної бібліотеки. Крім цього, також були зроблені скан копії креслень першого поверху..

Було створено проект на ПК (структура папок), в якому відбувалася вся робота та зберігалися всі вихідні дані. Архів з фотографіями був використаний для максимально реалістичного відтворення інтер'єру в 3D середовищі та елементів його наповнення.

Скан-копії креслень були отримані в форматі jpg (Рис.2.1), що не підходило для максимально точного відтворення простору академічної бібліотеки. На скан-копії були вказані точні розміри, то ж це полегшувало процес роботи над відтворенням середовища бібліотеки.

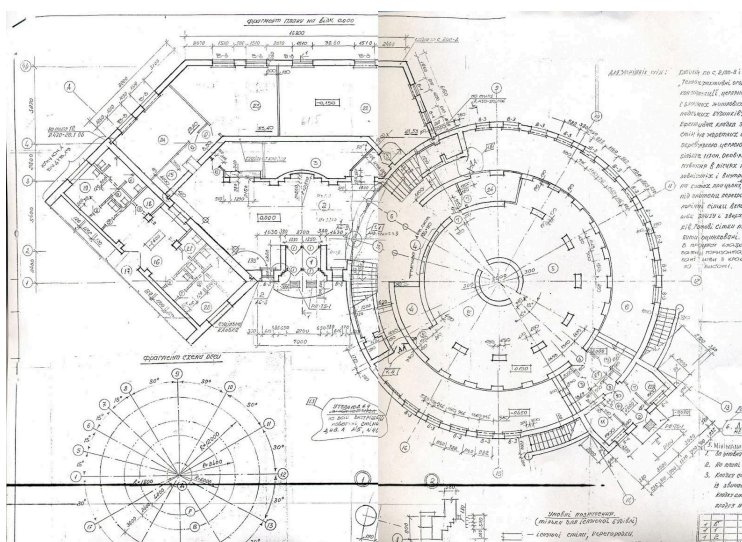


Рис.2.1 Отримана скан копія креслення бібліотеки

*Джерело: розроблено автором*

Для зручного процесу подальшої роботи отримані планування було вирішено відтворити в форматі dwg, який зберігає двовимірні та тривимірні проектні дані, що дозволяє зручно проектувати об'єкти. Відтворення відбувалось у програмному середовищі Autodesk AutoCAD. В AutoCAD був створений новий проект, куди було додано скан копії креслень, для подальшого відтворення.

Після цього на окремому шарі було виведено координаційні осі з позначками.

*“Координаційні осі - це тонкі штрих-пунктирні лінії на плані будівлі чи споруди. На зображенні кожної будівлі або споруди вказують координаційні осі й надають їм самостійну систему позначень. Координаційні осі наносять на зображення будівлі, споруди тонкими штрих-пунктирними лініями з довгими штрихами, позначають арабськими цифрами і великими буквами українського алфавіту (за винятком букв: Є, З, І, Ї, Й, О, Х, Ц, Ч, Щ, Ъ) у колах діаметром 6-12 мм. Пропуски в цифрових і буквених (окрім указаних) позначеннях координаційних осей не допускаються.”* — Державні будівельні норми [12].

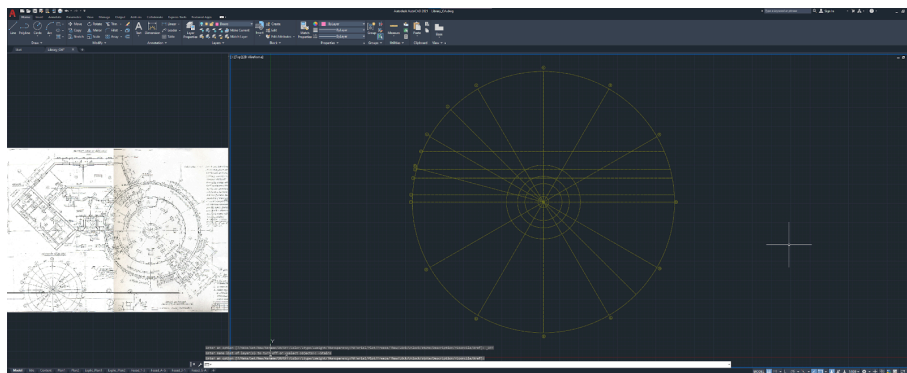


Рис.2.2 Виведення ліній осей в AutoCAD

*Джерело: розроблено автором*

Під час роботи в основному використовувалися такі інструменти як:

- Лінія — інструмент створення прямої лінії
- Коло — інструмент для створення кола, відповідно до вказаного радіуса
- Арка — інструмент для створення примітива, який є частиною кола
- Полілінія — схоже на інструмент Лінія, але дозволяє ставити точку на лінії та в цій точці змінювати її напрямок

- Текст — інструмент для створення тексту та текстових позначок

Також використовувались такі параметри модифікації як:

- Копія — модифікатор, за допомогою якого створюється копія одного або декількох вибраних об'єктів
- Віддзеркалення — модифікатор, з допомогою якого робиться віддзеркалення вибраних об'єктів відносно двох поставлених точок
- Обтинання — модифікатор, який використовується для коректування креслень, коли потрібно видалити частину кола, дуги, лінії і т.п.
- Обертання — модифікатор, який здійснює поворот вибраного об'єкта на вказаний кут
- Масштаб — дозволяє збільшувати або зменшувати вибраний об'єкт у вказану кількість разів
- Пересування — модифікатор, який виконує переміщення об'єктів.

Наступним кроком було відмальовування стін та колон, також на іншому шарі та іншим кольором. Після відмальовування стін були вирізані отвори під вікна та двері. Розміщені на окремих шарах та з різними кольорами. На цьому етапі роботи були використані всі перелічені вище модифікатори та інструменти.

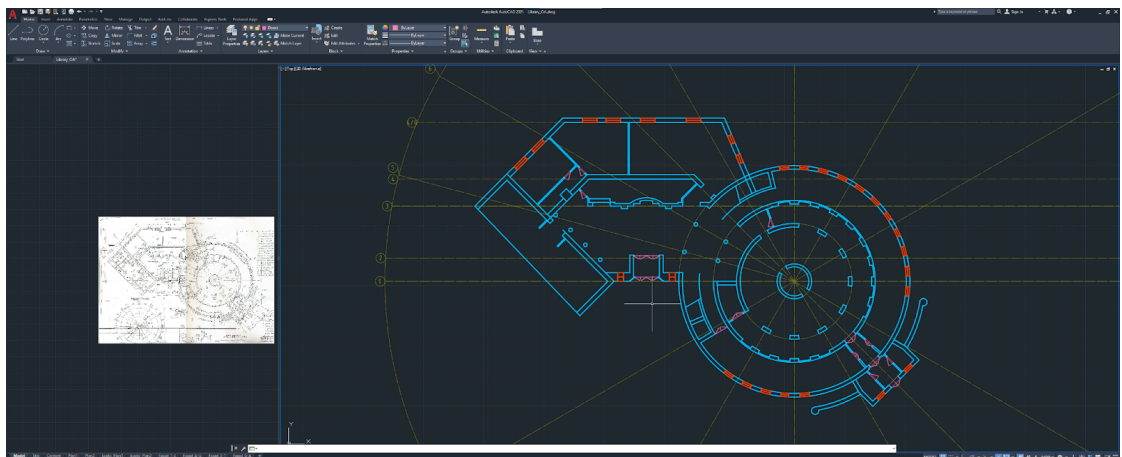


Рис.2.3 - креслення в AutoCAD

*Джерело: розроблено автором*

Закінчивши роботу з відтворенням креслень в Autodesk AutoCAD, ми експортували його у формат dwg для того щоб імпортувати його в 3Ds Max, адже саме цей формат в нього повністю коректно імпортується, що забезпечує зручність в подальшій роботі .

На цьому кроці підготовчі роботи були завершені.

### **2.3. Створення 3Д моделей в 3Ds Max**

Для відтворення реалістичної 3Д моделі використовувалося програмне забезпечення Autodesk 3Ds Max 2020. Структура проєкта складається з наступних папок:

- AutoCAD (містить у собі створені нами креслення у форматі dwg)
- Додаткові файли (допоміжні файли, такі як фотографії інтер'єру)
- Текстури (файли, з допомогою яких створювалися текстури)
- Пресети (заздалегідь підготовлені налаштування для рендера)
- Рендери (готові візуалізації, які були створені після завершення роботи)
- VFB (проміжкові файли рендеру, зберігається історія попередніх рендерів)
- Файли бекапу (резервне копіювання файлів)

Поділ на папки потрібен, адже забезпечує зручність роботи та можливість швидко знаходити потрібні файли, розуміти де і що знаходиться.

Перед початком основної роботи ми також встановили програмне забезпечення Blender та викачали файл з екстер'єром бібліотеки. Файл був отриманий від колеги по проєкту, який займався саме розробкою екстер'єру, в форматі .blend. Оскільки цей формат не підтримувався в 3Ds Max, потрібно було встановити саме Blender. Це все потрібно було для точного звірення пропорцій та розмірів, щоб в подальшому ми могли об'єднати про'єкти в один, та наш інтер'єр вмістився в екстер'єр.

Розпочали роботу над проєктом з імпорту креслень у .dwg форматі, які були відтворені на підготовчому етапі.

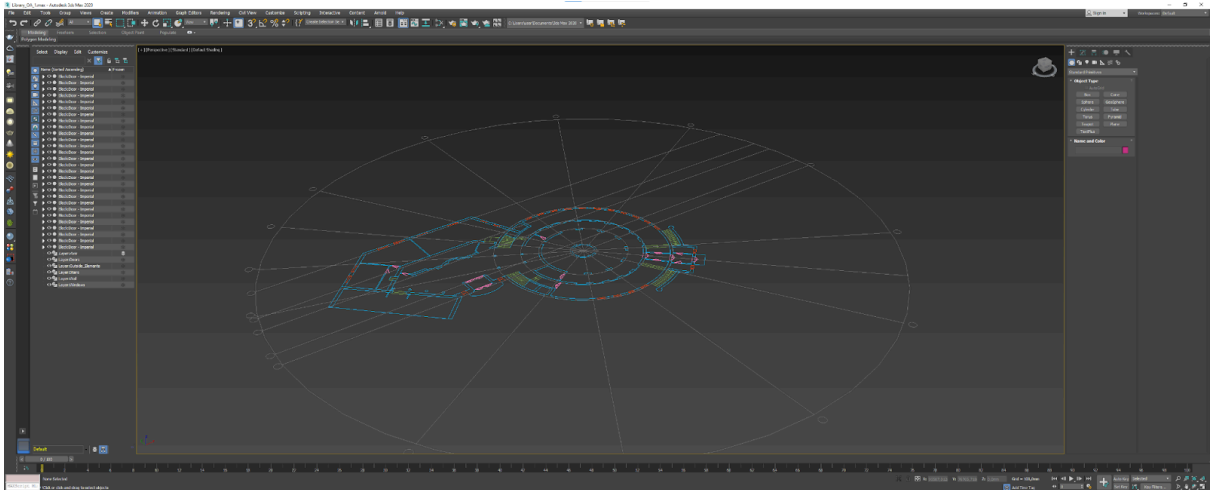


Рис.2.4 - експортовані креслення в 3Ds Max

*Джерело: розроблено автором*

Далі всі об'єкти з креслень було згруповано, заморожено (для того щоб ми не могли виділити ці об'єкти та випадково їх видалити чи змінити), та через Object Properties було виключено параметр “Show frozen in gray” (так як даний параметр фарбує всі заморожені об'єкти в сірий колір і їх тоді важко розрізнити) для візуального розуміння розташування об'єктів креслень. Активована прив'язка до кута та до точок, для того щоб могли ідеально рівно по кресленням відтворити стіни, прив'язуючись до ліній, які ми імпортували з AutoCAD. В налаштуваннях опорної точки, її було розміщено по центру ліній осей, для того щоб групу можна було розмістити по центру координат, для того щоб центр круглої будівлі розташовувався в центрі координат, та всі наступні об'єкти, які будуть створені, розташовувалися відносно центра круглої будівлі та центра координат.

Виведено циліндр з кількістю сторін 96 ребер (було візуально визначено кількість ребер для того, щоб будівля мала плавну округлість та якісно виглядала на рендерах), радіусом 12 000 мм та висотою 3 500 мм, це розміри основної круглої частини першого поверху бібліотеки. Об'єкт було конвертовано в Editable Poly [13] для подальшого полігонального моделювання внутрішніх стін та круглої частини корпусу бібліотеки, без конвертування об'єкта в режим полігонального редагування, ми б не змогли продовжити

роботу над моделюванням стін, так як цей режим дає можливість працювати конкретно з точками, ребрами, краями та полігонами об'єкта..

Таким самим способом були створені внутрішні стіни для подальшого їх з'єднання. Стіни бічної частини бібліотеки були виведені за допомогою об'єкта Plane (він потрібен для того щоб хоч якось почати роботу над відтворенням стін), який також було перетворено нами в Editable Poly, з можливістю додавання нових ребер та редагування точок, це потрібно для того щоб ми могли відтворити всю потрібну нам форму стін бібліотеки НаУОА.

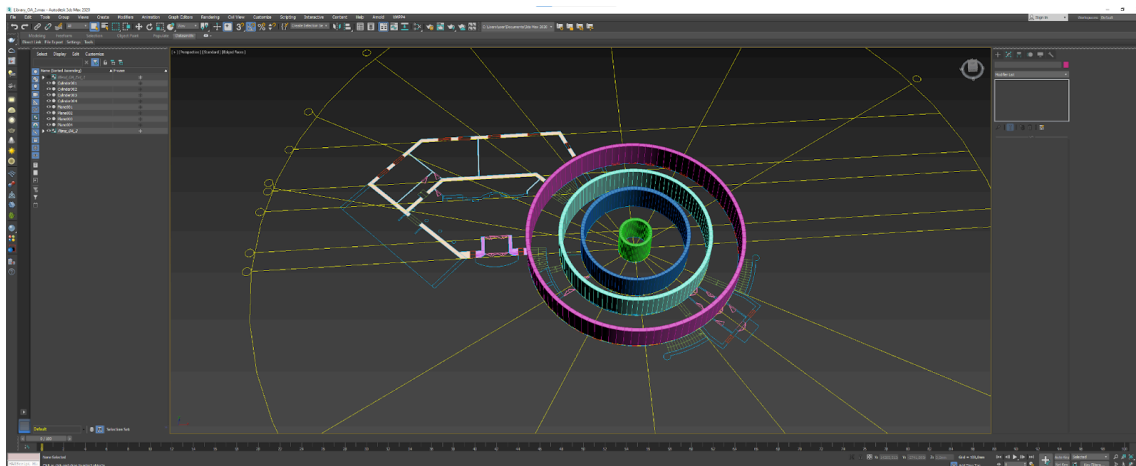


Рис.2.5 - внутрішні стіни будівлі

*Джерело: розроблено автором*

За допомогою операції Extrude [14] було видавлено полігони стін та полігони циліндра для створення зовнішніх стін на другому поверсі. Створення підлоги та стелі відбулося також за допомогою об'єктів Plane, переведених в полігональні об'єкти, яким було додано потрібну кількість ребер для задання форми.

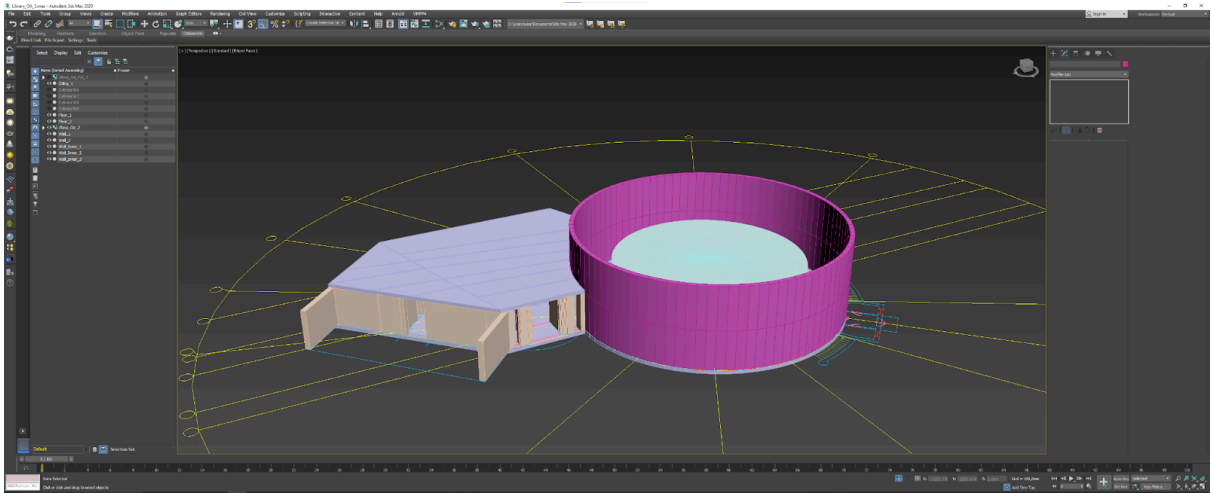


Рис.2.6 - створені підлога та стеля

*Джерело: розроблено автором*

Для вікон та дверей було створено декілька додаткових ліній ребер, після чого, з'єднавши їх, було створено отвори. Додаткові горизонтальні та вертикальні ребра були створені операцією Connect (вона надає можливість додати потрібну кількість додаткових ребер в потрібний нам орієнтації) відповідно до розмірів самих вікон. Вертикальні ребра створювалися відповідно до розташування вікон на кресленнях, а горизонтальні відповідно до висоти підвіконня і висоти самого вікна відносно підлоги. Далі коли додаткові ребра створені в нас з'являється два полігона (два через те що є зовнішня та внутрішня стіни), які дорівнюють розмірам самого вікна, після чого ми ці два полігони видаляємо та в нас з'являються два відкритих контури і простір між ними. На рівні відкритих контурів, виділяємо ці два контури та командою Bridge з'єднуємо їх між собою, таким чином в нас пропадає пустий простір та виходить отвір для вікна.

Під час редагування циліндра круглої частини бібліотеки на рівнях точок, ребер та полігонів було виведено повністю дах. На рівні полігонів було виділено верхню окружність, після чого її видалено і створено два відкритих контура. На рівні точок було задано висоту стін круглої частини бібліотеки, після чого перейшовши на рівень ребер, що нам дає можливість приміняти операцію Extrude, для того щоб видавлювати полігони по формі вже стелі круглої частини бібліотеки. У вигляді Front та Left, для точного позиціонування точок та з



допомогою інструмента Scale (масштаб) проводил такі операції як зменшення радіусу відкритого контуру, для відтворення верхньої частини з вікнами для другого поверху, яка має менший радіус. Так само з допомогою операції Scale та Extrude над відкритим контуром відтворили найвищу частину стелі бібліотеки.

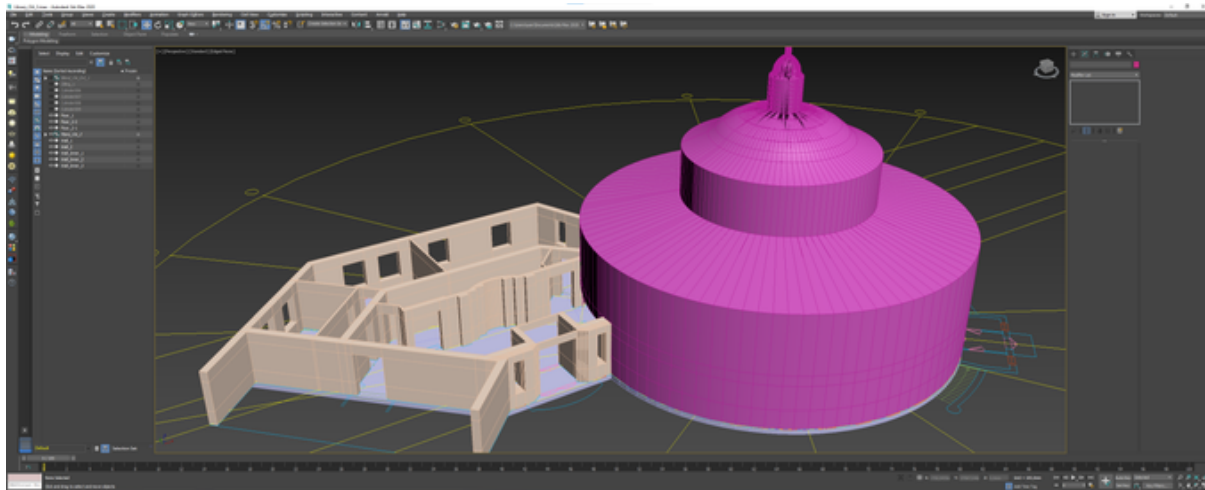


Рис.2.7 - дах круглої частини бібліотеки

*Джерело: розроблено автором*

Всі внутрішні стіни були об'єднані в один об'єкт, для того щоб далі за допомогою операції Bridge об'єднати між собою відповідні елементи стін, оскільки попередньо кожна стіна була виведена окремим блоком і для цільності будівлі бібліотеки вони потребують з'єднання..

За допомогою примітива циліндр було створено колону на першому поверсі, та продубльовано її кілька раз з параметром Instance, він дозволяє одночасно вносити зміни всім дочірнім елементам, коли редагується батьківський елемент, для того щоб в майбутньому деталізувати ці колони.

Два рівні другого поверху виводилися через примітив циліндр з заданими параметрами радіусу та висоти перекриття.

Створення сходів на другий поверх також було за допомогою елемента Cylinder. Спочатку було виведено одну бічну сторону та через деяку відстань іншу частину. Наріzano потрібну кількість ребер, яка рівна кількості сходів та за допомогою операції Bridge [15] було з'єднано обидві частини. За допомогою операції видавлення було видавлено кожен сходинку на невелику відстань, щоб



вони трішки випирали. Після цього елемент продубльовано за параметром Instance, та віддзеркалено.

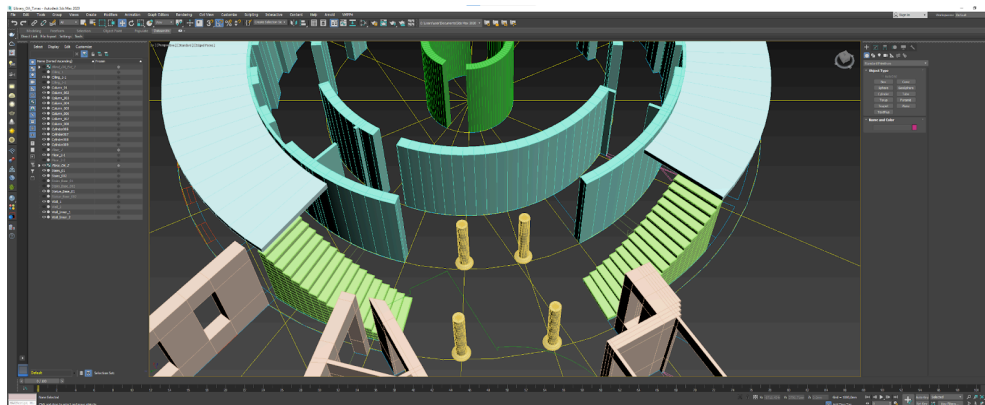


Рис.2.8 - сходи на другий поверх

*Джерело: розроблено автором*

Було змодельовано подіум для статуї, виходячи з елемента шестикутника (тому що половина подіуму це є половина примітива шестикутник, тому простіше було вивести саме його, та видалити зайву частину, ніж створювати цей об'єкт з іншого примітиву), після чого додатково нарізані ребра та видавлений плінтус. Також операцією Extrude видавлено невеличку випираючу частину подіуму.

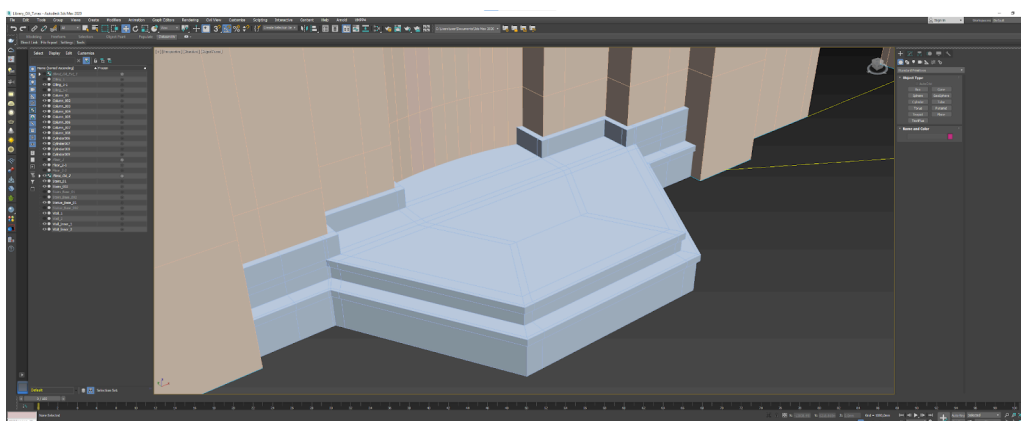


Рис.2.9 - подіум під статую

*Джерело: розроблено автором*

А над подіумом було змодельовано круглу стелю, вививши її з циліндра, оскільки саме цей примітив дозволив максимально відтворити реальну форму стелі, та видавивши додаткові ребра. Щоб створити стелю над подіумом, потрібно було вивести циліндр, задати йому висоту, перевести його в об'єкт

редагування полігонів та видалити непотрібні полігони, які знаходилися в стіні. Після з допомогою операції Extrude для окремо виділених ребер було виведено другий ярус стелі та ще раз провівши цю операцію було видавлено круглу стелю в основну стелю поверху. Далі зроблено два рази операцію Extrude. Одну для дна циліндра, а другу для тих полігонів, які утворилися після першої операції, для того щоб звільнити місце куди потім можна розмістити підсвітку.

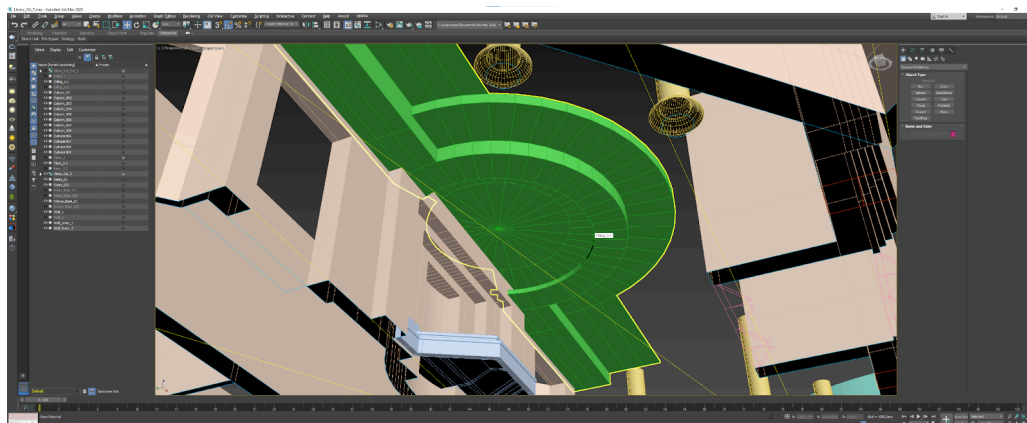


Рис.2.10 - трирівнева стеля над подіумом

*Джерело: розроблено автором*

Було створено деталізацію для колон на першому поверсі та створено одну колону на другому поверсі, яку також було деталізовано та після цього продубльовано потрібну кількість разів. Зробивши залежі копії потрібної кількості колон, переходимо в режим редагування лише однієї колони, так як всі інші є дочірніми тому зміни до них будуть примінятися автоматично. Так як це в нас об'єкт Editable Poly маємо можливість редагування точок, ребер та полігонів. Під час редагування ми дивимося на реальну фотографію колони та рахуємо кількість видавлених частин і візуально дивимося на топологію колони. Після цього додаємо з операцією Connect потрібну нам кількість ребер. Далі по топології за допомогою інструмента Scale, звужуємо або розширюємо полігони, таким чином задаючи форми декорації верхньої та нижньої частин колони. Також для щоб не було гострих кутів між полігонами, ми імітуємо High Poly модель надаючи плавності переходів між видавленнями та вдавленнями за допомогою інструмента Chamfer. Вибираючи кільце з ребер цей інструмент пом'якшує місце з'єднання полігонів, за допомогою автоматичного додавання

додаткових ребер відносно до показника зкруглення в налаштуваннях інструмента. Також над колоною на другому поверсі було змодельовано верхню, фігурну частину. Так само колони на другому поверсі між собою всі є залежними, тому нам вистачить змодельовати фігурну частину лише для однієї колони. Дивлячись на топологію об'єкта з реальної фотографії, та з використанням інструмента Extrude видавили ребра до верху, далі з допомогою інструмента Scael розширили верхню частину та з допомогою інструмента Connect додали додаткових полігонів по всім сторонам, для більшої плавності.

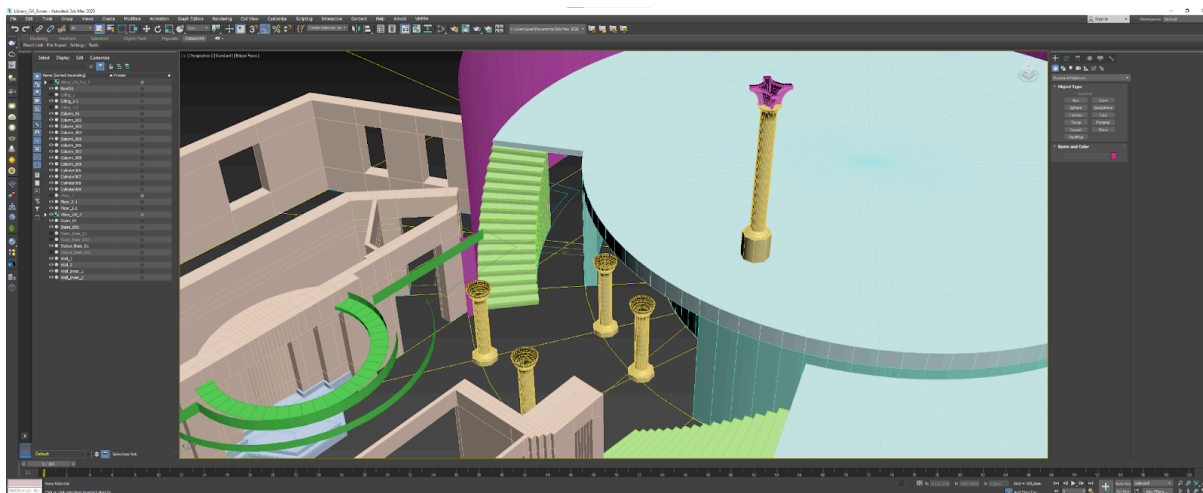


Рис.2.11 - колони

*Джерело: розроблено автором*

Під час моделювання для деяких об'єктів були приховані полігони (для того, щоб ми могли бачити що відбувається всередині бібліотеки, а не тільки зовнішні стіни) та застосований параметр Backface Cull. Цей параметр приховує задні сторони полігонів для таких об'єктів як стіни, стеля та підлога, при цьому залишаючи основну сторону полігонів, цей параметр дозволяє полегшити та пришвидшити процес моделювання, так як візуально полігони, в яких нормаль (основна сторона полігона) відвернута від нас, будуть динамічно приховуватися.

Відповідно до креслень, було також вирізано вікна на першому поверсі круглої частини бібліотеки. Вони вирізані з допомогою проведення додаткових ребер та видалення проміжкових полігонів. Далі, додатковим елементом було зроблено розмір отвору вікна на другому поверсі, продублювавши потрібну

кількість разів за допомогою операції Array з контрольною точкою по центру круглої будівлі. З допомогою операції Boolean було вирізано отвори на стінах другого поверху, та за допомогою операції Bridge були з'єднані отвори в стінах.

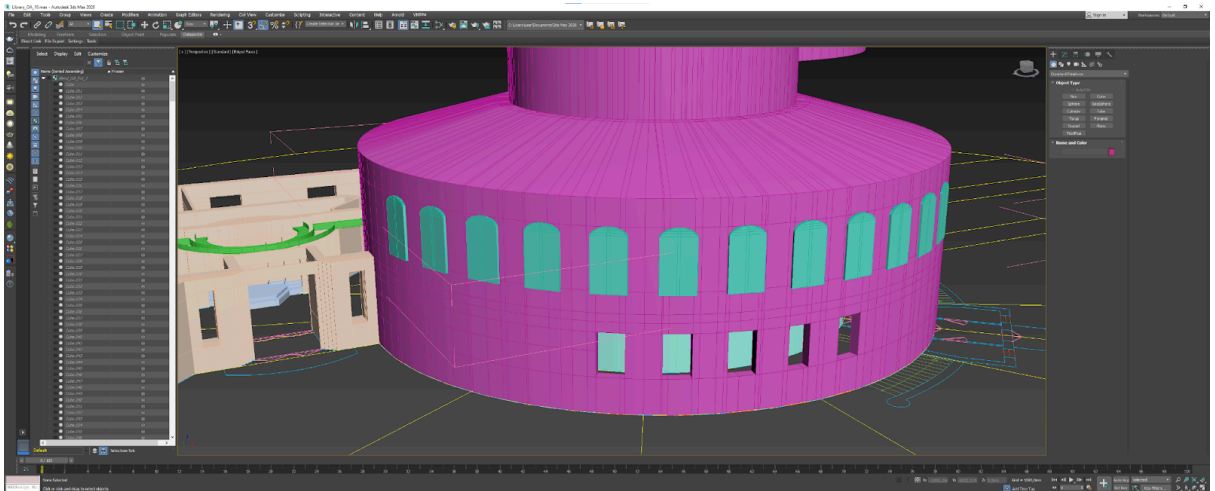


Рис.2.12 - вікна другого поверху круглої частини

*Джерело: розроблено автором*

Для верхніх вікон, які розташовані в куполі другого поверху, було видовжено колони та видалено полігони в стіні, для розташування саме віконних рам. Так як на другому поверсі по всьому периметру стіни розташовані вікна, тому відповідно мають бути колони, які підтримують частину стелі над вікнами. Відповідно до фотографій реальної бібліотеки візуально зрозуміло що колони, які підтримують стелю з фрескою є подовженням колон, які розташовані на другому поверсі, тому колони з другого поверху ми продублювали, на рівні полігонів видалили лишні декоранції, та змінили висоту колон, після цього розмістили їх між стінами, а зайві полігони стін було видалено, та два відкриті краї, які утворилися після цього було з'єднано між собою операцією Bridge.

Після цього розпочалася робота над моделюванням перил для сходів. Потрібно було змодельовати одну стійку перила, пробудлювати її для кожної сходинки та з'єднати всі між собою. Моделювання відбувалося за зробленою зблизька фотографією перила. Під час моделювання використовувалися геометричні примітиви такі як Tube, Box, Cylinder. Та використовувалися такі операції як Connect (для створення додаткових ребер), Bridge (для створення

містків між відкритими краями полігонів) та Extrude (для додаткового видавлення полігонів). Відповідно до топології об'єкта вертикальну металеву частину було відтворено з примітива Box. Далі переведено в режим Editable Poly, так як об'єкт має бути цільно злитим, тому ми маємо проводити операції на рівні точок, ребер та полігонів. Продублювавши елемент вертикальної частини, визначивши місце їх з'єднання ми додали ребра за допомогою операції Connect, видалили полігони, де буде з'єднання та після цього вибрали два відкритих краї та операцією Bridge з'єднали між собою. Далі всі з'єднання металевих конструкції між собою були з допомогою цих дій. Заокруглені металеві частини були створені з допомогою примітива Tube, це схоже на об'єкт трубу, що має обідок з отвором всередині. Далі в кінці круглих металевих частин потрібно було зробити звуження конструкції. Це виконувалося за допомогою операції Soft Selection, яка так і називається — м'яке виділення. Воно працює наступним чином: ми виділяємо коло з ребер, вмикаємо м'яке виділення та задаємо приблизну силу дії інструмента, до того моменту, поки кольором не виділяться ті ребра, з яких починається звуження, а далі інструментом Scale ми звужуємо кінець конструкції і вона плавно звужується.

Після цього були створені додаткові сходи на другий рівень другого поверху та також продубльовані туди перила.

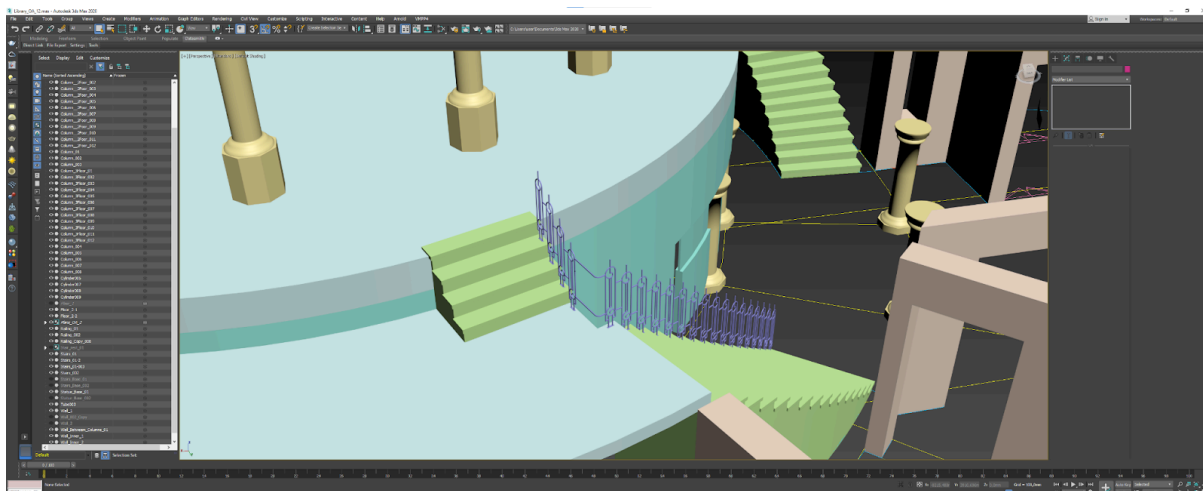


Рис.2.13 - перила

*Джерело: розроблено автором*



На першому поверсі були зроблені отвори для зони видачі книг, операцією Boolean була створена арка у стіні між круглою частиною та бічною частиною академічної бібліотеки.

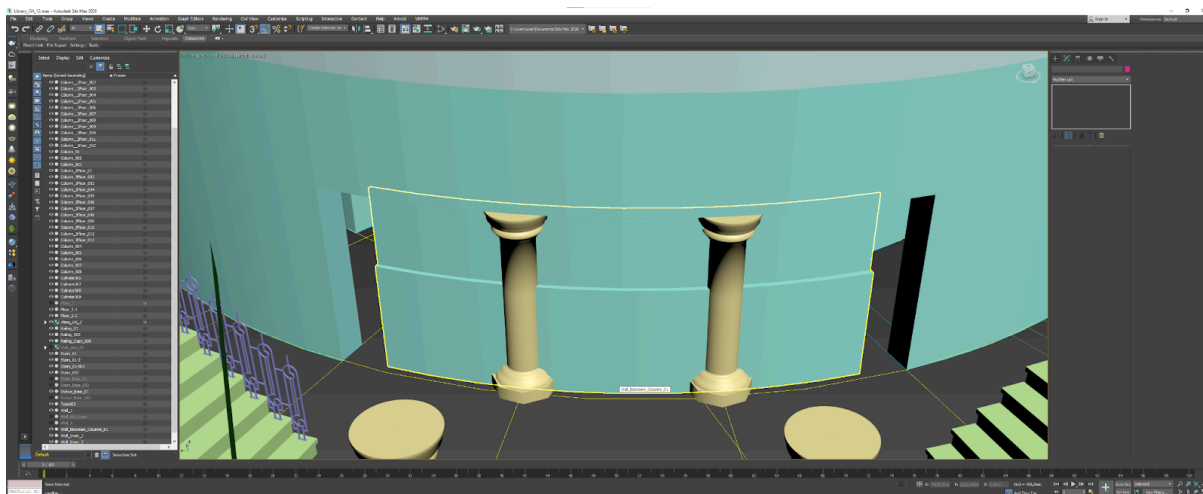


Рис.2.14 - зона видачі книг

*Джерело: розроблено автором*

Далі почалася робота над моделюванням перил огорожі для другого поверху. Вона створювалася за алгоритмом створення перил для сходів, з допомогою деяких примітивів та з використанням топології. Було створено одну частину, після чого за допомогою операції Mirror її було віддзеркалено та з'єднано між собою. Після чого кожену частину огорожі було продубльовано по колу та з'єднано між собою та з перилами для сходів.

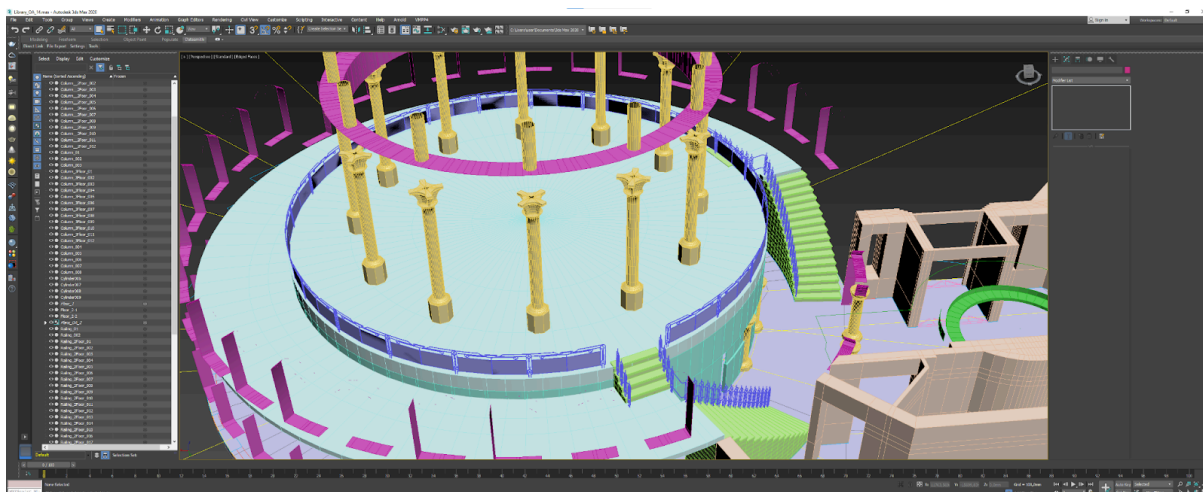


Рис.2.15 - другий поверх з огорожею

*Джерело: розроблено автором*

Наступним етапом було створення верхньої частини перил для огорожі на другому поверсі. Вони були створені за допомогою об'єкта Spline (крива лінія). Інструмент Spline містить в собі набір мінімум з двох точок, які з'єднані між собою в лінію, та за допомогою функції Bezier задається згин лінії під час редагування цих точок. Так як ми працюємо в 3D редакторі, то ці точки мають 3 координати, та цю криву лінію можна редагувати у просторі також. Після чого був створений контур з формою перил, також об'єктом Spline. Після чого модифікатором Sweep (модифікатор, який накладає форму одного контура на інший контур, після чого з'являється об'ємна фігура) було накладено контур форми перил на їх розташування.

Також в проект було експортовано статую у моделі Low Poly [16], її попередньо було отримано від керівника. За допомогою модифікатора TurboSmooth їй було надано вигляд High Poly, для того щоб візуально статуя мала гарний, реалістичний вигляд, з допомогою того, що модифікатор накладає велику кількість додаткових ребер, таким чином сильно зкруглюючи гострі кути між полігонами.. Також було змодельовано плінтуси по периметру стін.

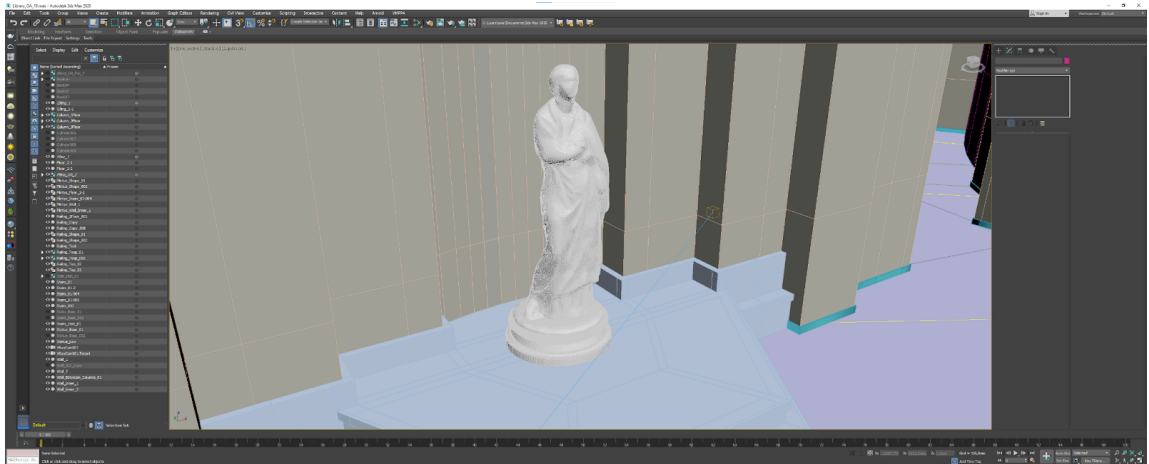


Рис.2.16 - статуя

*Джерело: розроблено автором*

Наступним етапом роботи було наповнення інтер'єру та текстурювання. Були додані дві пари входних дверей. Вони були змодельовані по топології дверей, максимально схожих на двері в бібліотеку, спираючись на реальні фото. Ми відштовхувались з примітива Box, перевівши об'єкт в Editable Poly, для

подальшого моделювання деталей дверей на рівні точок, ребер та полігонів. Були видавлені полігони, де буде розташовуватися скло, з допомогою примітивів циліндр змодельовані ручки для дверей, з заокругленими краями. Додані елементи освітлення такі як лед світильники (з використанням IES карт. IES карти — формат файлів з фотометричними даними освітлювальних приладів. Вони допомагають візуально максимально правильно змодельувати потік світла), прихована підсвітка, настінні світильники, лампи, датчики диму, вікна.

Були створені камери для перегляду проміжних рендерів. В 3Ds Max ми використали фізичні V-Ray камери. Можливість їх встановити з'являється після встановлення рендеру V-Ray. Їх використовуємо для створення коректного рендерингу, відображення V-Ray матеріалів та реалістичних фізичних налаштувань камери. Які далі були детальніше розписані. Проміжні рендери використовували для того щоб одразу перевіряти коректність налаштування та відображення матеріалів та світла, перед тим як створювати фінальні рендери. Так як процес рендерингу довгий, створення одразу фінальних рендерів та переналаштування матеріалів та світла могло б зайняти дуже багато часу.

Процес текстурування (накладання файлу з зображенням текстури на готовий об'єкт) складався безпосередньо з створення матеріалів (набір різних текстурних карт, з підібраними налаштуваннями, який приміняється на готовий об'єкт та під час рендеру створює імітацію реального фізичного матеріалу) під кожен об'єкт в інтер'єрі. В основному матеріал складається з декількох текстурних карт (файл з зображенням текстури під конкретну карту) та правильних налаштувань. При налаштуваннях матеріалів всі властивості було взято з реальних фізичних властивостей цих же матеріалів. Це все виконувалося у редакторі матеріалів. Здебільшого, всі матеріали були створені на основі VRay Mtl (це універсальний матеріал, який забезпечує фізично правильне освітлення у сцені, швидший рендеринг і зручні параметри відбиття та заломлення світла) з використанням додаткових карт.

Здебільшого використовувалися карти для таких параметрів як:



- Diffuse — текстура, якою задається основний колір поверхні. Для того, щоб добре працювати з картою нормалей, дзеркальною та іншими картами дифузна карта не повинна мати ніякого спрямованого світла на своїй поверхні, вона повинна мати тільки загальні навколишні оклюзії, тобто поверхня стає темніша в глибоких тріщинах і навколо тиснення деталі
- Reflect — відображення або сила відображення. Зазвичай значення обирається кольором відтінків білого та чорного або показником яскравості від 0 до 255(тому що колірний простір у комп'ютерних та графічних редакторах поміщається у діапазоні від 0 до 255), де 0 означає що об'єкт взагалі нічого не віддзеркалює, а 255 значить що матеріал має максимальне віддзеркалення. Також, якщо задати не відтінки сірого, а колір, то це буде колір віддзеркалення.
- Bump — карта для імітації рельєфу поверхні. Створюється в градаціях сірого кольору. При рендерингу сірі області карти (мається на увазі сірий колір —  $R = 150, G = 150, B = 150$ ) відобразатиметься звичайним чином, чорні будуть замальовуватись, а білі — освітлюватись (чим область світліше, тим більше рівень яскравості) — ця ілюзія світла і тіні створює ефект рельєфу поверхні
- Glossiness — сила розмиття відображення або глянцевість. У блискучого матеріалу відображення майже не розмивається, тому на карті вони будуть майже білі, на потертих матеріалах відображення розмивається дуже сильно, тому карта буде більш темною в цьому випадку.
- Refraction — заломлення світла через матеріал, або прозорість. Цей параметр впливає на те як матеріал буде пропускати через себе світло. Відповідно якщо колір карти чорний то переломлення немає та матеріал повністю непрозорий, чим світліший колір тим прозоріший матеріал. Існує показник переломлення (IOR), цей показник вже є сталим для багатьох матеріалів. Наприклад, вода має показник 1,33157, золото має показник 0,47.
- Displace — має вплив схожий на параметр Bump, але ця карта насправді деформує поверхню під час рендерингу, в той час коли Bump робить лише

симуляцію деформації. Наприклад матеріал камінь в проекті деформований через параметр Displace.

- Normal map — карта для імітації рельєфу поверхні. Дозволяє змінювати нормаль відображуваного пікселя ґрунтуючись на кольоровій карті нормалей, в якій ці відхилення зберігаються у вигляді текселя, колірні складові якого [r, g, b] інтерпретуються в осі вектора [x, y, z], на основі якого обчислюється нормаль, використовувана для розрахунку освітленості пікселя. Завдяки тому, що в карті нормалей використовують 3 каналу текстури, цей метод дає більшу точність, ніж Bump mapping, в якому використовується тільки один канал і нормалі, по суті, всього лише інтерпретуються в залежності від «висоти»
- AmbientOcclusion (AO) map — карта затінення, використовується для надання об'єктам більш реалістичного вигляду шляхом імітації м'яких тіней
- Opacity map (opaque map), (карта непрозорості) — формується на основі чорно-білої текстури, де повністю чорний колір буде відображатися прозорим (вирізатися з дифузної карти), білий повністю залишатиметься відображатися, а сірі відтінки відображатимуться напівпрозоро з альфа інтенсивністю.

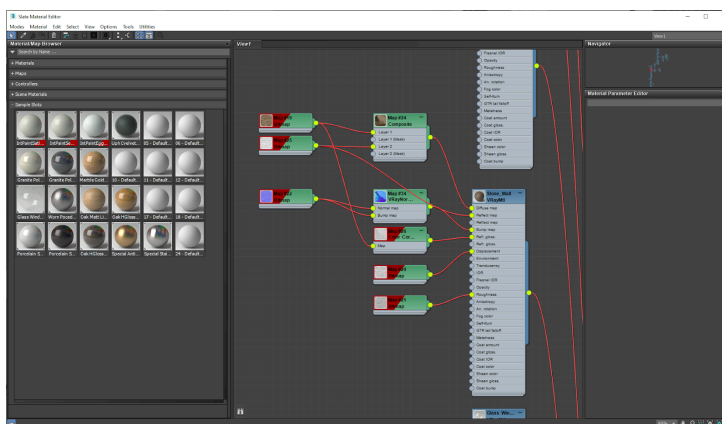


Рис.2.17 - Material Editor (редактор матеріалів), текстура каміння

*Джерело: розроблено автором*

Для складних об'єктів використовувався загальний матеріал Multi Sub Object, який включає в себе налаштування всіх матеріалів об'єкта, які потім застосовуються по полігонам або по елементам. MultiSub Object є стандартним матеріалом в 3Ds Max, який одноразово застосовується до об'єкта. Під час створення цього матеріалу, в нього є додаткові слоти для підключення інших матеріалів, таким чином створивши декілька матеріалів типу V-Ray Mtl, їх всі можна підключити до MultiSub Object. Далі під час редагування об'єкта на рівні полігонів або елементів, окремим полігонам та елементам можна задати параметр ID, який якраз і відповідає за номер відображуваного матеріалу на об'єкті, так як по порядку додані матеріали до MultiSub Object. Він зазвичай приміняється для складних об'єктів, щоб не розділяти один великий об'єкт на декілька маленьких об'єктів, тому він таким чином оптимізує сцену.

Деякі матеріали потребували більш детального підходу для їх створення, з використанням більшої кількості додаткових процедурних карт. Процедурне текстурування — метод створення текстур у комп'ютерній графіці, при якому зображення створюється за допомогою програмного алгоритму (процедурного алгоритму), а не задається наперед. Найкраще процес процедурного текстурування представити у вигляді блоків (операторів). Існує три типи блоків:

- генератори;
- фільтри;
- допоміжні.

Кожен генератор і фільтр реалізує встановлений процедурний алгоритм з певною сукупністю змінних параметрів. Для створення «природних» текстур, таких як дерево, граніт, метал, каміння, лава як фільтри використовуються фрактальний шум (англ. fractal noise) і ніздрюваті текстури (англ. cellular textures). Властивості процедурних текстур:

- оборотність. У процедурній текстурі зберігається вся історія її створення;
- малий розмір (якщо як вихідні дані для процедурних алгоритмів виступають тільки числові значення);

- необмежена кількість варіацій при використанні стохастичних (що використовують генератор псевдовипадкових чисел) алгоритмів;
- масштабованість до будь-якого розміру (залежить від процедурного двигуна/бібліотеки);
- одночасно з підсумкової текстурою дуже легко виходять alpha-, bump-, reflect-карти.

Процедурні карти допомагають створити більш складний матеріал, наприклад поєднавши просту текстуру з текстурою потертості, подряпини чи пилюки. Або, з допомогою процедурних карт, можна створити додаткові карти виходячи лише з однієї текстури. Зазвичай в нашому проекті це і використовували. За допомогою процедурної карти Color Correction ми створювали чорно-білі карти для відображення та для рель'єфності наступним способом. Створювали процедурну карту Color Correction, підключали текстурну карту, яку також під'єднували в канал Diffuse, після чого вибирали режим відображення Black and White, щоб були виключно відтінки сірого та за допомогою повзунків Brightness та Contrast відтворювали потрібні нам відтінки карти, які в подальшому будуть впливати на відображення та рель'єфність. Наприклад, процедурною картою Checker (шахматна текстура), можна одразу перевіряти як матеріал примінився на об'єкт. Ми її використовували перед тим як застосувати основний матеріал на об'єкт, щоб подивитися чи вистачить застосувати для об'єкта модифікатор UVW Map (використовується для швидкого накладання матеріалу на простіші об'єкти з визначенням типу накладання на об'єкта та розмірами текстури). Наприклад, якщо в нас є підлога без складних форм, то можна примінити UVW Map, з розташуванням текстури Plane, тоді текстура буде розміщена на плоскому об'єкті, без її деформації. Так само на стіни, сходи з розташуванням Box чи колони з розташуванням Cylinder вистачить застосувати цей модифікатор. Наприклад якщо ми обираємо Box, тоді поверх об'єкта створюється для нас невидима коробка, на сторонах якої розташовується матеріал і накладається на об'єкт. Інший спосіб накладання текстури це створення розгортки за допомогою модифікатора Unwrap UVW.

Розгортка це відображення 3D об'єкта в площині з двома координатами. В загальному ми розгортаємо 3D об'єкт на 2D текстуру. Розгортку створюємо за допомогою розрізів ребер, які потім розтягуємо на плоский квадрат в редакторі. Для більшості об'єктів ми застосовували модифікатор UVW Map.

UV mapping [17] — процес в 3D моделюванні, який полягає в накладанні двовимірного зображення на тривимірну модель. Літерами U і V позначають осі координат площини розгортки, оскільки літери X, Y і Z використовуються для позначення просторових координат.

UV mapping (UV-розгортка) — відповідність між координатами на поверхні тривимірного об'єкту (X, Y, Z) і координатами на текстурі (U, V). Значення U і V зазвичай змінюються від 0 до 1. Розгортка може будуватися як вручну, так і автоматично.

Сучасне тривимірне апаратне забезпечення вважає, що UV-перетворення в межах одного трикутника є афінним — тому достатньо задати U і V для кожної вершини кожного з трикутників. Втім, як саме стикувати трикутники один з одним, вибирає 3D-моделер, і вміння будувати вдалу розгортку — один з показників його класу.

Окремо ми створили HDRI карту (High Dynamic Range Imaging, це панорамний знімок HDR якості з високими діапазонами, приміняється для створення реалістичного освітлення сцен), яка надає вуличне освітлення та додає вигляд з вікон під час рендерингу. Для нашого проєкту використано готову HDRI карту. Приміняли HDRI карту під час налаштування оточення. В налаштуваннях виставили сферичну форму та вказали як карту оточення, щоб вона відображалась під час рендерингу, також трішки підняли освітлення карти, щоб рендер був світліший.

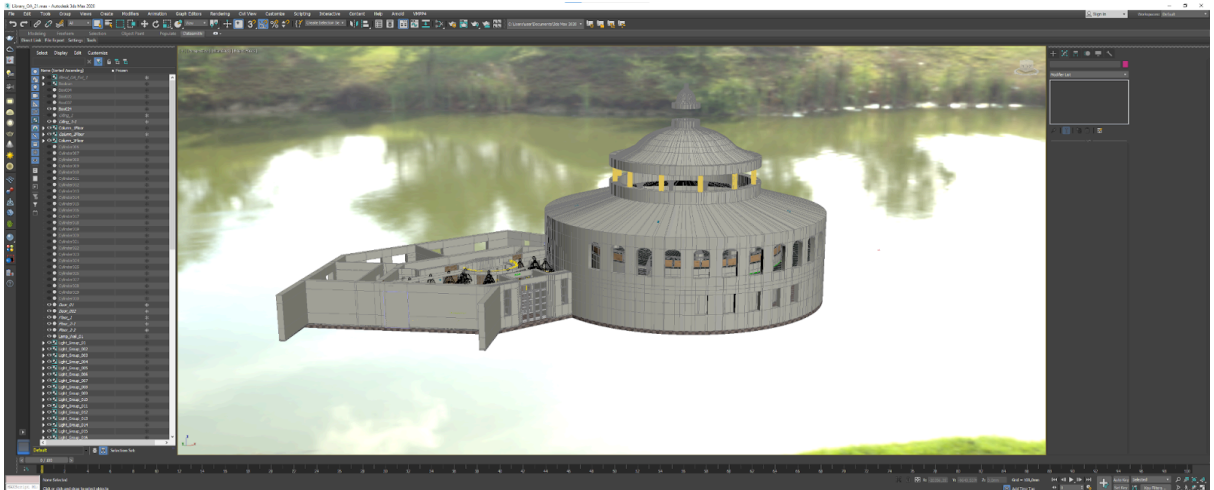


Рис.2.18 - додана HDRI карта

*Джерело: розроблено автором*

Далі потрібно було провести тестування налаштувань матеріалів, освітлення та камер.

В основному більшість камер мали такі налаштування:

- Film gate: 65 мм
- Focal length: 35 мм
- Zoom factor: 1
- Film speed (ISO): 100
- F-Number: 2,5
- Shutter speed: 200
- Exposure: Physical Exposure
- White balance: Netural

На цьому етапі потрібно було перейти до рендерингу. 3D-візуалізація або 3D-рендеринг – це 3D-процес комп'ютерної графіки, який автоматично перетворює 3D в 2D-зображення з 3D-фотореалістичними ефектами або без фотореалістичної візуалізації на комп'ютері.

Рендеринг — це остаточний процес створення фактичного 2D-зображення або анімації з підготовленої сцени. Це можна порівняти з фотографією або зйомкою сцени після завершення її установки у реальному житті. Розроблено кілька різних, зазвичай спеціалізованих, методів рендеринга. Вони варіюються від чітко нереалістичного рендеринга каркасів за допомогою полігонального

рендеринга, до більш просунутих методів, таких як: рендеринг Scanline, трасування променів та інші. Рендеринг може займати від частки секунди, до доби для одного зображення/кадра. Загалом, різні методи краще підходять для фотореалістичного рендеринга або рендеринга у реальному часі.

Зображення — результат рендерингу може бути описане як набір певних візуальних особливостей, що відповідають справжнім фізичним явищам, властивостям об'єкту. Дослідження та розробки у області рендерингу продовжують шукати найкращі шляхи для більш кращої та ефективної їх симуляції. Деякі з цих особливостей можуть бути прив'язані до конкретного алгоритму або методу, інші ж являти їх сукупність.

Рендери є двох типів GPU (під час рендерингу використовується графічний процесор) та CPU (під час рендерингу використовується центральний процесор). Їх можна використовувати як окремо так і разом. V-Ray дозволяє одночасно рендерити на графічному та центральному процесорі.

Для нашого проекту було налаштовано рендер V-Ray GPU + CPU з розширенням 1920x1080 пікселів.

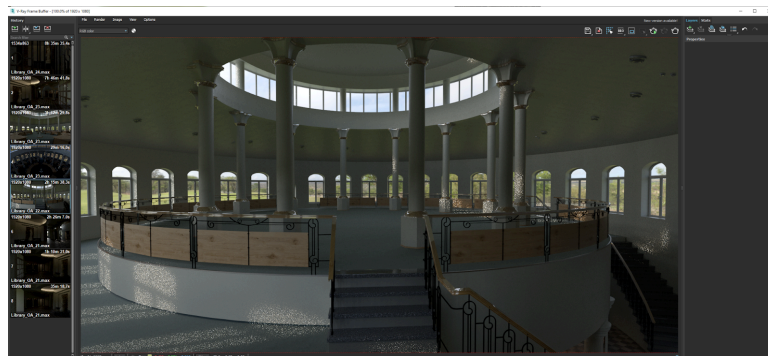


Рис.2.19 - проміжний рендер другого поверху

*Джерело: розроблено автором*

Після перевірки проміжних рендерів, було виконано певні доробки. В освітленні було змінено рівень загального освітлення HDRI карти, змінено яскравість освітлення світильників на першому та другому поверс. Також відбулося налаштування матеріалів, змінено налаштування Displacement для матеріалу камінь. Вставлені міжкімнатні двері, віконні рами, столи та стільці, книжкові шафи, додано підставку для Біблії та захисне скло. І тоді було

виконано фінальні рендери в 3Ds Max. Отримані зображення розміщені в Додатку 1.



Рис.2.20 - фінальний рендер другого поверху в 3Ds Max

*Джерело: розроблено автором*

Порівнявши реальні фото круглої частини бібліотеки та отримані в результаті роботи рендери можна помітити що на стелі немає фрески, яка безперечно є такою собі особливістю та чудовою окрасою приміщення, відтворити її для нас було дуже важливо.

Для якісного відтворення фрески в 3D середовищі ми скористалися UVW mapping, спираючись на реальне зображення.

UVW mapping - це процес створення розгортки текстурної карти з третьою координатою W для того щоб в подальшому її можна було накладати на 3D об'єкт.

Вміння 3D моделера будувати справді якісну розгортку є одним з показників професійності.

На сьогодні існує кілька показників якості розгортки, але вони можуть дещо суперечити одне одному:

- Відсутність областей на яких є зайві геометричні спотворення
- Вдале розташування “швів” - ліній що відповідні до одного ребра але розташовані в різних місцях текстури. Шви доцільні і потрібні якщо в природі поверхні має “розриви”, прикладом може бути одяг, і відповідно недоцільні якщо таких немає.



- Вдале поєднання симетричних та асиметричних ліній. Симетрія підвищує деталізацію текстури та спрощує роботу художників по текстурам вцей же час асиметричні деталі допомагають візуально оживити об'єкт.
- Максимальне використання площі текстури.

У випадку нашого завдання, а саме відтворення фрески через використання розгортки, ми для початку від'єднали полігони котрі власне були на тій частині стелі де є фреска від іншої частини будівлі.

Далі примінили до цього модифікатор UVW mapping.

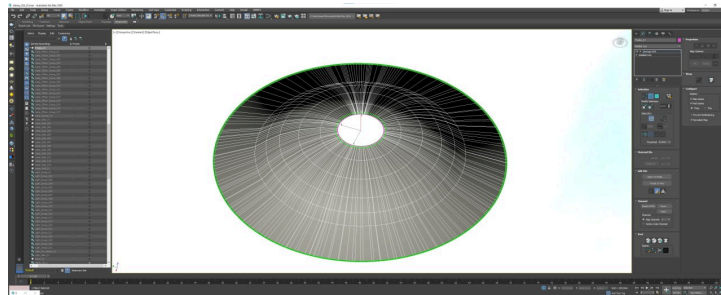


Рис.2.21 - частина на якій буде розміщено фреску

*Джерело: розроблено автором*

Це забезпечило можливість створити розгортку, зробити шви, по яких ми розрізали об'єкт та спроектували його на 2D площину

Після цього ми продублювали матеріал який використовувався раніше для стін та в канал diffuse map (основний колір колір поверхні) підключили текстуру яка містить в собі власне зображення фрески.

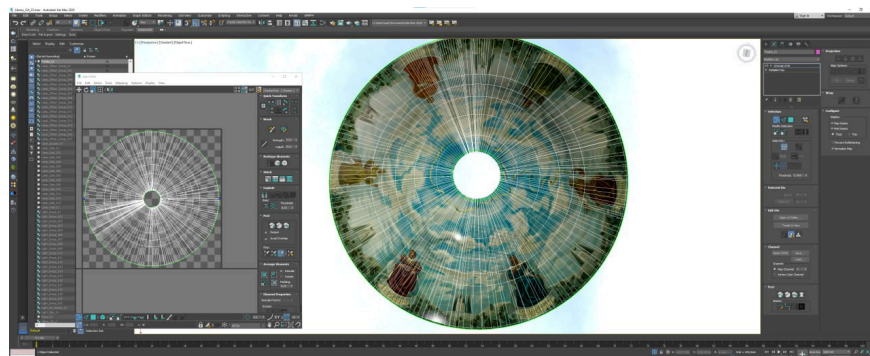


Рис.2.22 - розгортка фрески

*Джерело: розроблено автором*

Застосувавши цей матеріал ми отримали чудовий результат, який має дуже реалістичний вигляд.



Рис.2.23 - відтворена фреска

*Джерело: розроблено автором*

Використана нами техніка UVW mapping має ряд переваг у використанні:

- Контроль над текстурою. Ми можемо дуже чітко контролювати те, як на поверхні моделі буде розміщуватись текстура. Це дуже полегшує роботу зі складними текстурами.
- Редагованість. UVW mapping дозволяє швидко та легко редагувати масштаб текстур на моделі та їх розміщення.
- Незалежність від топології. UVW mapping дозволяє накладати текстуру на поверхню об'єкта незалежно від його топології. Тобто ми можемо без впливу на розташування текстури змінити форму або розмір моделі.

Окрім вже створеної частини, бібліотека також має ще одну читальну залу ім. сім'ї Васишиних, це є важливою та великою частиною бібліотечного приміщення, тож його ми також відтворили.

Для початку також було зібрано необхідні реальні фотографії та креслення приміщення.

Створення 3D моделі цієї частини було таким самим способом як і попередні приміщення.

В 3Ds Max, спираючись на креслення та фото ми вивели стіни, підлогу та стелю, зробили всі потрібні вікна та двері, відтворили всі елементи інтер'єру та

наповнили читальну залу потрібними елементами.

Затекстурували всі об'єкти відповідними текстурами.



Рис.2.24 - модель приміщення читальної зали

*Джерело: розроблено автором*

В результаті нам вдалось максимально реалістично відтворити приміщення. Було налаштовано світло відповідно до реальних фото та створені рендери, які дозволили переглянути результат та впевнитись що все готово до перенесенні в віртуальне середовище.

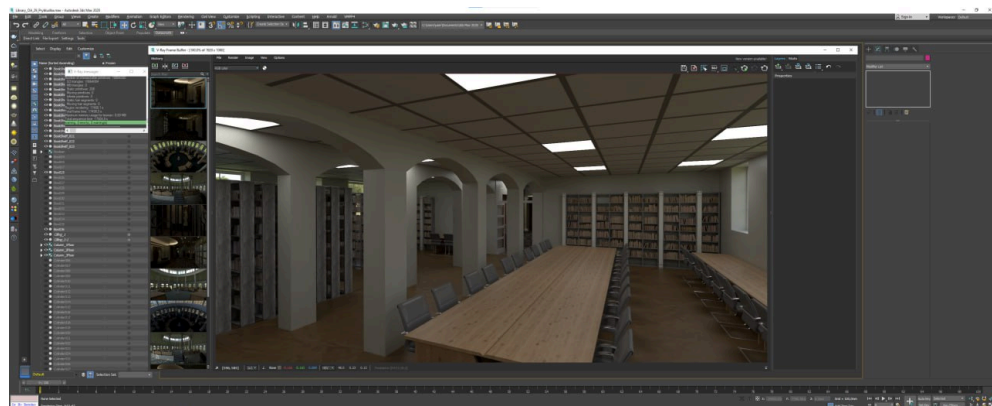


Рис.2.25 - рендер читальної зали в 3Ds Max

*Джерело: розроблено автором*

Щоб коректно все підготувати до перенесення дуже важливо було налаштувати освітлення та перевірити щоб всі лампи були ввімкнені, а також

обов'язково треба слідкувати чи немає ніяких проблем з топологією об'єкта, бо після експорту сцени всі недопрацювання будуть помітними.

## 2.4. Інтеграція 3Д моделей в Unreal Engine

Unreal Engine 5 [18] - це дуже потужний інструмент для розробки ігор, який має купу своїх особливостей та переваг використання:

- Швидкодія. UE5 має оптимізований рушій, чим дозволяє створювати ігри які містять велику кількість деталей та ефектів, працюючи без значної втрати продуктивності.
- Реалістичність. Нам пропонують фото реалістичну графіку, яку забезпечують технології Lumen та Nanite, які відповідно забезпечують освітлення та роботу з великою кількістю деталей.
- Широкий інструментарій. UE5 надає великий набір інструментів для роботи, сюди входять засоби моделювання, текстурювання, анімації та інші.
- Спільнота. UE5 має спільноту розробників, а також отримує регулярні оновлення від EpicGames [19], а це дозволяє стежити за новими тенденціями та швидко отримувати підтримку в ході розробки.

Експорт з 3Ds Max в Unreal Engine виконувався для того щоб запустити проект на ігровому рушії, який не гірший за реалістичні зображення. Перевага ігрового рушія це швидкість провантаження об'єктів та текстур в режимі реального часу та можливість зробити 3D сцену інтерактивною.

При підготовці сцени до експорту були видалені всі зайві об'єкти у файлі (такі об'єкти як Spline, V-Ray камери, допоміжні об'єкти для моделювання такі як різні заготовки та примітиви) проекта, перевірене освітлення та перевірені матеріали.

Після цього встановлено середовище Unreal Engine 5 з Epic Game Store. Створено новий проект типу Game. Цей тип забезпечує можливість в подальшому додати в проект елементи інтерактивної взаємодії між користувачем та елементами інтер'єру. Після цього активовано також плагін

Datasmith, який дозволяє експорт сцен з різних 3D середовищ на ігровий рушій Unreal Engine.

Плагін Datasmith Exporter для Unreal Engine є інструментом, який дозволяє імпортувати дані з різних програм для проектування, таких як Autodesk Revit, Autodesk 3ds Max, SketchUp, SolidWorks та інші, безпосередньо в Unreal Engine. Це дозволяє дуже швидко та ефективно імпортувати готові моделі, сцени та дані для використання у їхніх проектах в Unreal Engine.

Ось ключові характеристики та переваги плагіну Datasmith Exporter:

1. Імпорт геометрії та матеріалів: Плагін дозволяє імпортувати геометрію та матеріали з різних програм для проектування, зберігаючи їхню структуру та налаштування.
2. Збереження ієрархії об'єктів: Під час імпорту плагін зберігає ієрархію об'єктів, це дозволяє зручно керувати моделями у Unreal Engine та редагувати їх.
3. Підтримка матеріалів і текстур: Дозволяє імпортувати матеріали та текстури з програм для проектування, що дозволяє зберегти реалістичний вигляд моделей у Unreal Engine.
4. Автоматизований процес імпорту: Плагін надає автоматизований процес імпорту, що спрощує та прискорює імпорт моделей та сцен у Unreal Engine.
5. Підтримка різних програм для проектування: Плагін підтримує імпорт даних з таких програм, як Autodesk Revit, Autodesk 3ds Max, SketchUp, SolidWorks, Cinema 4D та інші.
6. Підтримка різних типів даних: Плагін дозволяє імпортувати різні типи даних, включаючи геометрію, матеріали, текстури, світло, камери, анімації та багато іншого.

Datasmith Exporter є потужним інструментом для розробників, що працюють з Unreal Engine, оскільки він спрощує процес імпортування та інтеграції готових моделей та даних у їхні проекти.

Після встановлення цього плагіну було виконано імпорт сцени зі всіма об'єктами, текстурами та освітленням. Так як це різні рушії, то все ж потрібно додатково налаштувати відображення матеріалів та освітлення.

Після коригування яскравості світла та налаштувавши деякі матеріали складні матеріали, такі як метали, дерево та каміння, маємо готовий проект в Unreal Engine, який можна запустити в реальному часі. Це дає змогу віртуально відвідати академічну бібліотеку НаУ ОА.

Маємо зазначити що UE5 надає широкі можливості в плані роботи зі світлом.

Переваги та можливості налаштування світла в UE5 [20]:

1. Lumen Global Illumination: Lumen - це нова технологія глобального освітлення, яка дозволяє відображати світло в грі в реальному часі. Автоматично розсіює світло в просторі, забезпечуючи реалістичне освітлення сцени навіть при динамічних змінах.
2. Nanite Virtualized Geometry: Nanite дозволяє використовувати величезну кількість деталізації в реальному часі, що дозволяє створювати неймовірно деталізовані моделі з реалістичною геометрією і деталями, які виглядають природно при будь-якому освітленні.
3. High-Quality Reflections: UE5 використовує новий метод розрахунку віддзеркалень, який дозволяє створювати високоякісні віддзеркалення навіть у складних сценах. Це важливо для створення реалістичних зображень, особливо у воді та на дзеркальних поверхнях.
4. Global Illumination Mixing: UE5 дозволяє розробникам налаштовувати взаємодію різних джерел світла для створення унікальних і атмосферних ефектів.
5. Volumetric Clouds: Система хмар в UE5 дозволяє створювати реалістичні та динамічні хмари, які реагують на освітлення та атмосферні умови.
6. Real-Time Ray Tracing: UE5 підтримує трасування променів у реальному часі для створення реалістичних ефектів світла і тіні.

7. HDR Display Output: UE5 підтримує широкий динамічний діапазон (HDR), що дозволяє створювати більш яскраві та деталізовані зображення.

Ці можливості дають надзвичайний рівень контролю над освітленням та створюють умови для створення графіки, яка наближена до реальності.

Щоб мати можливість комфортно взаємодіяти та переміщуватись по створеному середовищі варто додати персонажа та застосувати колізію до об'єктів.

В Unreal Engine 5 колізія використовується для визначення зіткнення між об'єктами у грі, що забезпечує можливість обробляти взаємодію між ними.

Налаштування колізії [21] містить деякі параметри для об'єктів у грі:

- Collision Objects. Ми маємо можливість встановити колізійні об'єкти для своїх моделей, що будуть визначати область, котра буде якось взаємодіяти з іншими ігровими об'єктами.
- Collision Meshes. UE 5 підтримує використання окремих колізійних мешів для складних об'єктів, які можуть мати власну форму колізії.
- Collision Shape Settings. UE5 надає можливість налаштувати різні параметри колізійних форм, це може бути радіус, нахил, висота та інші.
- Collision Volumes. Ми можемо створити колізійні області для об'єктів які мають складну форму.
- Collision Channels. UE5 містить різні типи колізій, такі як блокування, затримка, огляд, ігнорування, проходження, тощо.

Для нашого проекту нам був потрібен базовий персонаж у вигляді звичайної капсули. В Unreal Engine 5 це робиться за допомогою вбудованого компонента Character Capsule [22]. Цей компонент створює простий персонаж з колізійною капсулою що визначає його область в грі. Для цієї капсули ми налаштували потрібні під ігрову сцену розміри.



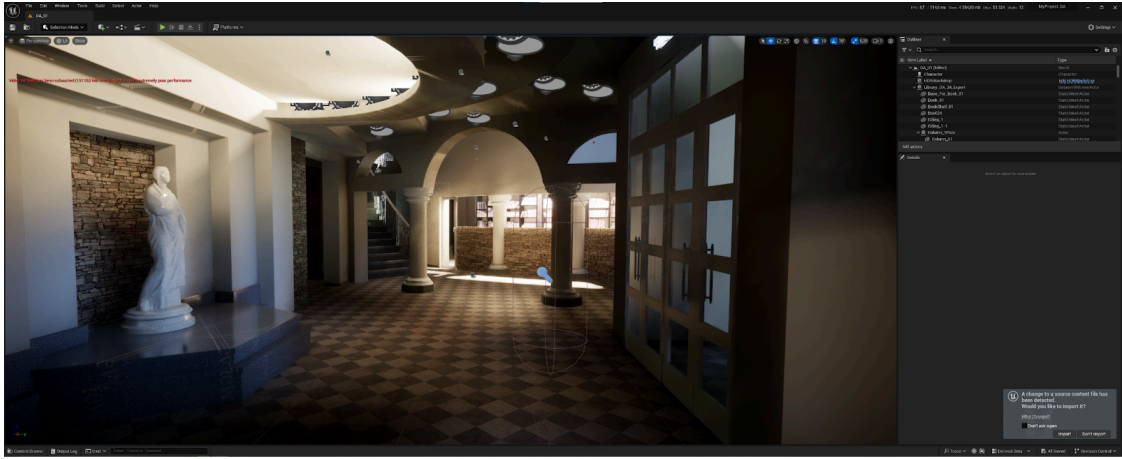


Рис.2.26 - перший поверх в UE5

*Джерело: розроблено автором*

Інтеграція проекту в Unreal Engine виконана, проект працює та виглядає дуже реалістично. Детально розглянути вигляд готового проекту можна в Додатку 2.

Для подальшого вдосконалення нашого проекту може бути використаний інший доступний функціонал Unreal Engine 5, розвинути більшу інтерактивність можна буде попрацювавши з блупрінтами та віджетами.

Блупрінти - це система візуального програмування в Unreal Engine 4 та 5. Вона базована на вузлах з даними, функціями та подіями. Основною перевагою та ідеєю блупрінтів є те що розробник може відтворити складну логіку гри використовуючи графічний інтерфейс, писати код немає потреби.

Є кілька головних деталей, які треба знати перед початком роботи з блупрінтами [23]:

- Події та функції. Розробники мають можливість створювати свої власні функції, блупрінти реагують на події, такі як припустимо зіткнення об'єктів або натискання клавіші.
- Візуальне представлення. Графічний інтерфейс блупрінтів дозволить розробникам дуже просто відстежувати зміни та логіку в грі.



- Вузли та зв'язки. Розробники мають можливість просто створювати логіку для гри, обробляти події, анімацію, фізику та багато іншого, обираючи вузли з палітри. Потік даних і керування визначають зв'язки між вузлами.
- Візуальне програмування. Це зручний варіант для тих хто не має достатнього досвіду в програмуванні. Можна будувати логіку гри, перетягуючи та з'єднуючи вузли.

Віджети [24] - це компоненти інтерфейсу користувача, які надають можливість створення графічних елементів у грі. Дозволяють створити кнопки, текстові поля, прогрес - бари, меню та інші подібні інтерактивні елементи.

Головне про віджети:

- Анімація і гнучкість. Є можливість анімувати віджети, змінювати їх стан під час гри реалізувати динамічний інтерфейс
- Віджет на блупрінтах. Наприклад, можемо за допомогою віджету на блупрінті відображати результат гри.
- Взаємодія з користувачем. Можна налаштувати щоб віджет реагував на дії гравця, наприклад, зменшенн життя персонажа або ж відкриття меню.
- Створення інтерфейсу. З віджетами людина яка розробляє гру може налаштувати інтерфейс як їй потрібно, можна підігнати розміри або наприклад розташування елементів інтерфейсу.

Наступним етапом роботи було створення інтерактивності для користувачів. Основне завдання полягало в створенні початкового варіанту мобільного телефону з яким взаємодіяти гравець.

Найбільш доцільно це робити через використання віджетів.

Ми створили в проєкті нову папку в якій власне будуть міститись всі необхідні матеріали. В UE 5 в проєкті ми створили Widget Blueprint, який відповідатиме за інтерфейс телефону і в ньому, відкривши Widget Blueprint Editor, відтворили необхідний UI, в початковому варіанті телефон містить на екрані кілька кнопок, серед них є кнопка для відкриття клавіатури для набору номера, саме вона на цьому етапі буде робочою.

Widget Blueprint - це такий тип блупринта в UE5, який використовують саме для створення UI частини, тобто дозволяє візуально створити потрібний інтерфейс, без написання коду.



Рис.2.27 - UI телефону

*Джерело: розроблено автором*

Зазначимо, що Unreal Engine надає можливість зручно створювати такі віджети редагувати елементи, наприклад змінювати розмір чи розміщення.

Для того щоб телефон мав початковий функціонал, а саме можливість набору номера, ми створили ще один Widget Blueprint який відповідає за клавіатуру з цифрами.

В цьому блупрінті ми на інтерфейсі телефону відтворюємо відповідно функціонал клавіатури, цифри, кнопка для виклику та дві кнопки для повернення на головний екран, а також рядок де буде відображатись номер який вводить гравець.

Але щоб відповідна кнопка відкривала клавіатуру необхідно додати для неї таку функцію. Для додавання логіки ми відкрили вкладку Graph та підключили Event On Clicked для кнопки. Та налаштували щоб після натискання гравцем на відповідну кнопку перед нами відкривався віджет на якому телефон вже з відкритою клавіатурою.

Event On Clicked є власне подією яка спрацьовує, коли відбувається клік по об'єкту, наприклад по кнопці. При цьому подія може бути налаштована так як потрібно розробнику, можна відкривати якісь вікна, іншу сторінку, або ж змінити стан гри.

Клавіатура на телефоні також має бути робочою, тобто має бути можливість набрати номер або за необхідності повернутись на попередню сторінку.

Реалізувати це можна за допомогою підключення того ж Event On Clicked до кожної кнопки налаштувавши відповідну поведінку. Тобто для кнопок з цифрами реакцією на клік буде відображення відповідної цифри, а реакцією на клік по кнопці Back повертає на головний екран.



Рис.2.28 - телефон в UE5

*Джерело: розроблено автором*

Щоб додати готовий телефон до нашої сцени та він міг викликатись гравцем під час гри, в Level Blueprint [25] ми додали відслідковування події, налаштувавши логіку таким чином, щоб після натискання гравцем комбінації правої клавіші миші та клавіші P, викликався попередньо створений віджет з головним екраном телефону.

Після цього запустивши проект ми переконались в вірному налаштуванні логіки роботи телефону.

Цим же методом роботи з віджетами в подальшій роботі над проектом можна пропрацювати більш широкий функціонал, який буде доцільним та чудово доповнить віртуальне середовище наукової бібліотеки та додасть деякої інтерактивності, що в свою чергу стане неабиякою особливістю нашого проекту.

## **2.5. Інтерактивна складова віртуального середовища в Unreal Engine**

Для того щоб розширити можливості проекту та зробити його цікавішим для користувача, було вирішено додати інтерактивності, а саме реалізувати телефонну систему з якою міг би взаємодіяти гравець. Для цього було розроблено інтерактивний асет, який був успішно інтегрований у віртуальне середовище.

Створення всіх інтерактивних асетів в UE передбачає розуміння багатьох теоретичних концепцій та вміння застосовувати їх на практиці.

Найважливішим є знання UMG UI Designer (Unreal Motion Graphics UI Designer) в Unreal Engine [26] - це інструмент, з допомогою якого ми можемо створювати користувацькі інтерфейси, такі як ігрові HUD [27], меню та інші елементи які треба показати користувачам. В основі UMG є віджети, які являють собою серію попередньо створених функцій, враховуючи слайдери, кнопки, індикатори, прапорці та індикатори.

Основні компоненти UMG:

- Віджети: будівельні блоки з допомогою яких створюють інтерфейси, які можна стилізувати.
- Блупринти: спеціальні блупринти, які використовують для того щоб редагувати віджети, мають дві вкладки для конструкції: Design та Graph.

Основні функції UMG:

- Стилізація теми: можна налаштувати вигляд віджетів для створення стилів або консистентних тем.
- Розширення взаємодії: є можливість створити більш складні взаємодії, наприклад drag-and-drop [28]

- Динамічне зв'язування: можливість пов'язати властивості віджетів з даними в блупринтах.
- Візуальне створення UI: є можливість візуально відтворювати інтерфейси, не пишучи при цьому програмний код.

Використання UMG значно може спростити та пришвидшити процес розробки, забезпечує контроль над кінцевим продуктом та робить його гнучким до змін.

Створення інтерактивного асету [29] телефонної системи для нашого проекту вимагає детального планування та використання різних інструментів, які надає рушій UE5, зокрема ті ж блупринти та віджети.

Перш за все ми визначили які саме функції буде мати телефон та склали їх в список:

- Дзвінок (набір номера та імітація дзвінка на цей номер)
- Повідомлення (отримання та відправлення текстового повідомлення)
- Інтернет браузер (є можливість відкрити Google та вільно шукати інформацію)
- Музика (програвання музичних композицій які були додані до системи в процесі розробки)
- Фото (можливість зробити фото як на фронтальну так і на селфі камеру, зберігання зроблених світлин)
- Галерея (можливість переглянути всі зроблені раніше фото)

За допомогою системи UMG (Unreal Motion Graphics), ми створили віджети, які є відображенням інтерфейсу телефону. До кожної функції яка була потрібна, дзвінки фото і тд. було зроблено окремий віджет. Після цього ми створили ще один, головний віджет який поєднував всі інші, за рахунок чого ми отримали цілісну телефонну систему.



Рис.2.29 - блюпринт в UE5

*Джерело: розроблено автором*

Також, крім віджетів дуже важливими були блюпринти. Для кожної функції необхідно було зробити окремий блюпринт, який би визначав її поведінку.

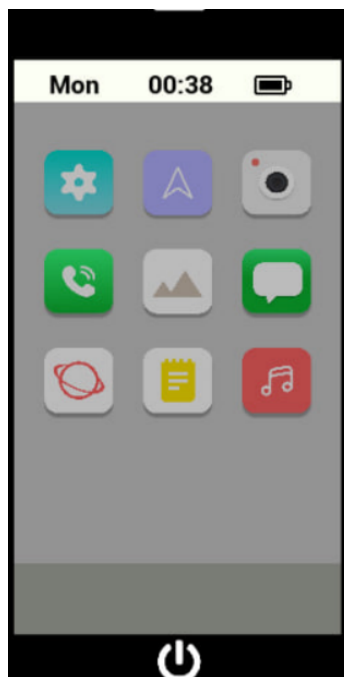


Рис.2.30 - телефон в UE5

*Джерело: розроблено автором*

Після цього було налаштовано взаємодію гравця з телефоном через інтерфейс та додано тригери [30], тобто створено події які активують різні функції телефону.

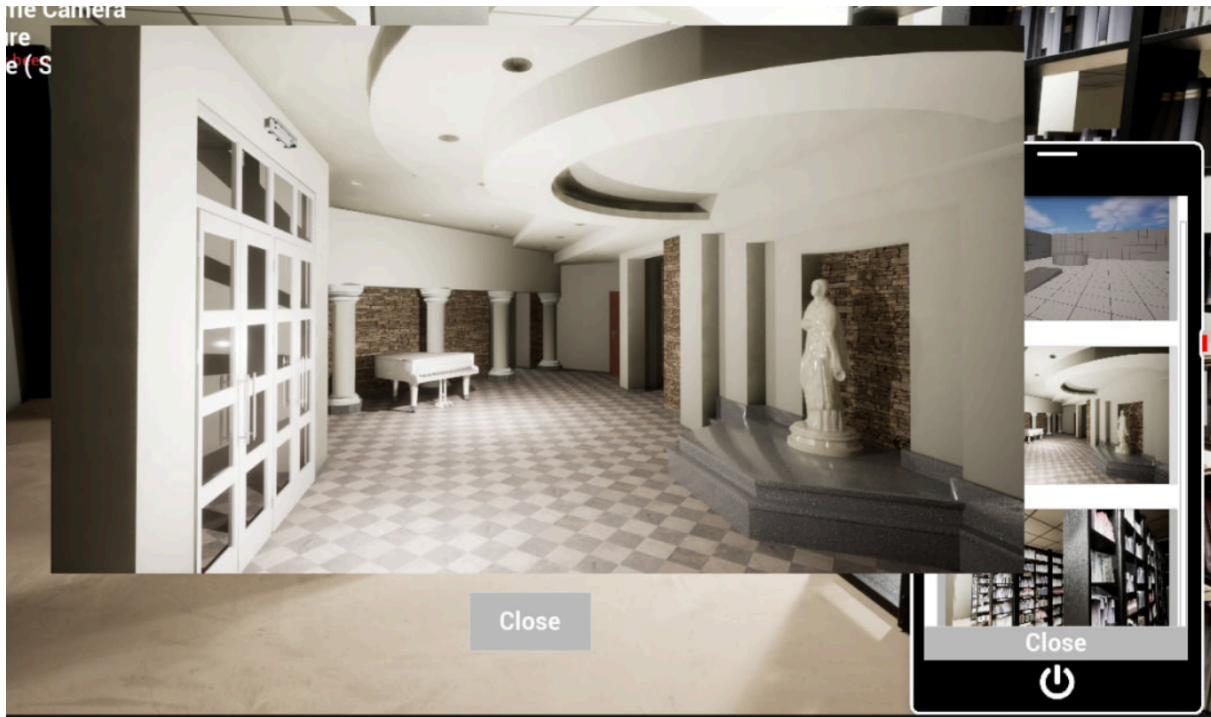


Рис.2.31 - функція перегляду фото

*Джерело: розроблено автором*

Останнім етапом було об'єднання ассету який мав в собі телефонну систему з створеним раніше віртуальним середовищем. Після підключення ми перевірили коректність роботи всіх функцій та виправили помилки.

## **2.6. Можливості для розвитку проекту та його перспективи**

Проект який є результатом виконання нашої кваліфікаційної роботи має багато варіантів для подальшого застосування. Може слугувати просто як гра або ж стати крутим маркетинговим іміджевим інструментом.

В процесі роботи ми зберегли кілька варіацій проекту, тому ми маємо повністю готове інтерактивне віртуальне середовище, віртуальне середовище без інтерактиву та 3D сцену в 3Ds Max.

Все це має місце, в той час коли фінальний варіант, за рахунок інтерактивності може бути повноцінною грою та виступати як розважальний інструмент, ми можемо використавши створену на початку роботи 3D сцену перенести її у формат який можна інтегрувати на сайт та додати це на сайт

бібліотеки чи академії. Такий варіант дасть змогу показати роботу величезній кількості аудиторії.

Можу в залежності від потреб студентів та абітурієнтів Острозької академії а також цілей адміністрації можна обрати найбільш підходящий варіант застосування або ж навіть втілити в життя обидві ідеї.

Наш проект може допомогти показати абітурієнтам чого вони можуть досягнути в процесі навчання та які ідеї реалізовувати. Також слугуватиме інструментом підтримки іміджу та просування Острозької академії як передового закладу вищої освіти.

У результаті ми отримали інтерактивне середовище, яке реалістично відтворює академічну бібліотеку Національного університету "Острозька академія", з можливістю взаємодії та дослідження для користувачів. Цей процес дозволив нам вивчити та використати різноманітні техніки та інструменти для створення віртуальних середовищ, а також розвинути наші навички у сфері 3D, комп'ютерної графіки та розробки ігор.



## ВИСНОВКИ

У ході роботи над створенням інтерактивного віртуального середовища інтер'єру академічної бібліотеки НаУОА, наша команда пройшла шлях успішного досягнення запланованих цілей, виявивши та розглянувши ключові аспекти технічних та дизайнерських вимог у створенні вражаючих віртуальних середовищ.

Результатом наших зусиль стала створена функціональна модель інтер'єру бібліотеки, що вражає своєю реалістичністю та високим рівнем деталізації. Ця модель не лише відповідає всім технічним вимогам, але й відкриває широкі можливості для використання в навчальних та інформаційних цілях. Вона стала не тільки інструментом для демонстрації, але й потужним засобом для навчання та дослідження. Користувачі віртуального середовища можуть вільно пересуватись приміщенням академічної бібліотеки, що є дуже незвичним досвідом. Ця робота відзначається високою актуальністю в контексті використання віртуальних середовищ, підкреслюючи, що відтворення інтер'єрів може слугувати інноваційним засобом в освітньому просторі.

Неабияким плюсом розробленого віртуального середовища є його використання як інтерактивного путівника для нових студентів. Це не лише полегшує їхню адаптацію, але й сприяє ефективній орієнтації в університетському просторі.

У цілому, результати нашої роботи не лише відкривають нові перспективи для подальшого вдосконалення віртуальних технологій в освіті, але й акцентують на їхньому важливому внеску у сучасний університетський дослідницький та освітній ландшафт.

Крім цього наша робота є важливим кроком у моєму особистому розвитку як спеціаліста, адже дозволила мені застосувати та вдосконалити навички 3D моделювання та розробки інтерактивних віртуальних середовищ.

## Список використаних джерел

1. Sketchfab - The best 3D viewer on the web. *Sketchfab*. URL: <https://sketchfab.com/> (дата звернення: 12.09.2024).
2. Jira Atlassian. *Collaboration software for software, IT and business teams*. URL: <https://www.atlassian.com/ru/software/jira> (дата звернення: 01.05.2024).
3. Учасники проєктів Вікімедіа. Jira – Вікіпедія. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/Jira> (дата звернення: 11.05.2024).
4. Kanban Jira | Atlassian. *Atlassian*. URL: <https://www.atlassian.com/ru/agile/tutorials/how-to-do-kanban-with-jira> (дата звернення: 12.05.2024).
5. autodesk. URL: <https://www.autodesk.com/products/autocad/overview?term=1-YEAR&tab=subscription> (дата звернення: 15.02.2024).
6. Учасники проєктів Вікімедіа. Autodesk 3ds MAX – Вікіпедія. *Вікіпедія*. URL: [https://uk.wikipedia.org/wiki/Autodesk\\_3ds\\_MAX](https://uk.wikipedia.org/wiki/Autodesk_3ds_MAX) (дата звернення: 23.05.2024).
7. Учасники проєктів Вікімедіа. Autodesk 3ds MAX – Вікіпедія. *Вікіпедія*. URL: [https://uk.wikipedia.org/wiki/Autodesk\\_3ds\\_MAX](https://uk.wikipedia.org/wiki/Autodesk_3ds_MAX) (дата звернення: 23.05.2024).
8. Unreal Engine - Datasmith Plugins. Unreal Engine. URL: <https://www.unrealengine.com/en-US/datasmith/plugins> (дата звернення: 23.05.2024).
9. Unreal Engine. URL: <https://www.unrealengine.com/en-US> (дата звернення: 23.05.2024).
10. Adobe Photoshop. Adobe. URL: [https://www.adobe.com/ua/products/photoshop.html?gclid=Cj0KCQjw0ruyBhDuARIsANSZ3wrCB2SB7qm7G7BHnx1mf0x3ePdtaImPIhxaWZmH\\_mdXhm1zBTFPzLgaAjLYEALw\\_wcB&skwid=AL!3085!3!442252953564!e!!g!!adobe%20photoshop&mv=search&mv2=paidsearch&sdid=LZ32SYVR&ef\\_id=Cj0KCQjw0ruyBhDuARIsANSZ3wrCB2SB7qm7G7BHnx1mf0x3ePdtaImPIhxaWZmH\\_m](https://www.adobe.com/ua/products/photoshop.html?gclid=Cj0KCQjw0ruyBhDuARIsANSZ3wrCB2SB7qm7G7BHnx1mf0x3ePdtaImPIhxaWZmH_mdXhm1zBTFPzLgaAjLYEALw_wcB&skwid=AL!3085!3!442252953564!e!!g!!adobe%20photoshop&mv=search&mv2=paidsearch&sdid=LZ32SYVR&ef_id=Cj0KCQjw0ruyBhDuARIsANSZ3wrCB2SB7qm7G7BHnx1mf0x3ePdtaImPIhxaWZmH_m)

dXhm1zBTFPzLgaAjlYEALw\_wcB:G:s&s\_kwcid=AL!3085!3!442252953564!e!!g!!adobe%20photoshop!1712281352!67017686636&gad\_source=1 (дата звернення: 23.05.2024).

11. Blender. URL: <https://www.blender.org/> (дата звернення: 23.05.2024).

12. Координаційні осі. DBN. URL: [https://dbn.co.ua/blog/koordinacijni\\_osi/2021-05-24-24553](https://dbn.co.ua/blog/koordinacijni_osi/2021-05-24-24553) (дата звернення: 23.05.2024).

13. Autodesk 3ds Max. Autodesk. URL: <https://help.autodesk.com/view/3DSMAX/2024/ENU/?guid=GUID-D05DF785-F905-453E-BF64-DB4D59A9F200> (дата звернення: 23.05.2024).

14. Autodesk 3ds Max Help 2024. Autodesk. URL: <https://help.autodesk.com/view/3DSMAX/2024/ENU/?guid=GUID-36DF6364-2624-4CE1-8AC0-5CEED91CB393> (дата звернення: 23.05.2024).

15. Autodesk 3ds Max Help 2022. Autodesk. URL: <https://help.autodesk.com/view/3DSMAX/2022/ENU/?guid=GUID-B4E6A89C-3481-4F32-AF22-FC3C06B9F899> (дата звернення: 23.05.2024).

16. High-Poly Models vs Low-Poly Ones. CGI Furniture. URL: <https://cgifurniture.com/high-poly-models-vs-low-poly-ones/> (дата звернення: 23.05.2024).

17. What is UV Mapping? 3D Coat. URL: <https://3dcoat.com/ua/articles/article/what-is-uv-mapping/> (дата звернення: 23.05.2024).

18. Unreal Engine 5. Unreal Engine. URL: <https://www.unrealengine.com/en-US/unreal-engine-5> (дата звернення: 23.05.2024).

19. Epic Games Store. Epic Games. URL: <https://store.epicgames.com/en-US/> (дата звернення: 23.05.2024).

20. Lighting for Educators and Students. Unreal Engine. URL: <https://cdn2.unrealengine.com/lighting-for-educators-and-students-20mb-6b793664e8.pdf> (дата звернення: 23.05.2024).

21. Collision Overview. Unreal Engine. URL: <https://docs.unrealengine.com/4.26/en-US/InteractiveExperiences/Physics/Collision/Overview/> (дата звернення: 23.05.2024).
22. Remnant: From the Ashes Discussion. Steam Community. URL: <https://steamcommunity.com/app/617290/discussions/0/1741140686421557734/?l=russian> (дата звернення: 23.05.2024).
23. Blueprints Maps. Unreal Engine. URL: <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/UserGuide/Maps/> (дата звернення: 23.05.2024).
24. Blueprint API Widget. Unreal Engine. URL: <https://docs.unrealengine.com/4.27/en-US/BlueprintAPI/Widget/> (дата звернення: 23.05.2024).
25. Level Blueprint Types. Unreal Engine. URL: <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/UserGuide/Types/LevelBlueprint/> (дата звернення: 23.05.2024).
26. UMG (Unreal Motion Graphics). Unreal Engine. URL: <https://docs.unrealengine.com/4.26/en-US/InteractiveExperiences/UMG/> (дата звернення: 23.05.2024).
27. Жаргоніст: PN8O0BB8BZ. TSEIVO. URL: <https://tseivo.com/b/jargoniist/t/pn8oobb8bz> (дата звернення: 23.05.2024).
28. UX Патерни для перетягування Drag-and-Drop компонентів у дизайн системі. ProDesign. URL: <https://prodesign.in.ua/2018/11/ux-paterni-dlya-peretyaguvannya-drag-and-drop-komponentiv-u-dizayn-sistemi/> (дата звернення: 23.05.2024).
29. What are Assets in Game Design? RetroStyle Games. URL: <https://retrostylegames.com/blog/what-are-assets-in-game-design/> (дата звернення: 23.05.2024).
30. Triggers. Unreal Engine. URL: <https://docs.unrealengine.com/4.26/en-US/Basics/Actors/Triggers/> (дата звернення: 23.05.2024).

## Додатки

### Додаток 1



## Додаток 2



**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, куші та квіти, а також твердого покриття, такого як доріжки, тротуари та дороги.
- **Освітлення:** Додайте освітлення до 3D-моделі, щоб імітувати умови освітлення в різні пори дня. Це включає моделювання природного освітлення, такого як сонячне світло і тіні, а також штучного освітлення, такого як вуличні ліхтарі, освітлення будівель і внутрішнє освітлення. Освітлення повинно бути налаштоване так, щоб створити реалістичну і візуально привабливу 3D-модель.
- **Текстури та матеріали:** Використовуйте відповідні текстури і матеріали для точного відображення поверхонь будівель і ландшафтів. Це включає в себе моделювання матеріалів, таких як цегла, бетон, скло і метал, а також застосування відповідних текстур, таких як фарба, штукатурка і підлога. Текстури та матеріали повинні бути підібрані таким чином, щоб відображати реальний вигляд університету.
- **Функціональність:** Пророблена вами робота повинна бути реалізована в Unreal Engine з метою подальшого використання з метою створення інтерактивного середовища. 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 змодельовати будівлі, використовуючи точні розміри, матеріали та текстури. Будівлі повинні бути змодельовані таким чином, щоб відображати їхній реальний вигляд, включаючи архітектурні деталі, такі як вікна, двері та дахи. 3D модель також повинна включати внутрішні приміщення, такі як класи, лабораторії та офіси. Внутрішні приміщення повинні бути змодельовані з відповідними меблями, обладнанням.
- Етап ландшафтного дизайну та моделювання місцевості: На цьому етапі команда моделює всі відкриті простори, ландшафти і рельєф в межах університету, щоб точно відобразити топографію місцевості. Сюди входить моделювання рослинності, такої як дерева, кущі та квіти, а також благоустрою, наприклад, доріжок, тротуарів та доріг.

- **Етап освітлення:** На цьому етапі команда додає освітлення до 3D-моделі, щоб імітувати умови освітлення в різні пори дня і року. Сюди входить моделювання природного освітлення, такого як сонячне світло і тіні, а також штучного освітлення, такого як вуличні ліхтарі, освітлення будівель і внутрішнє освітлення. Освітлення повинно бути налаштоване так, щоб створити реалістичну і візуально привабливу 3D модель.
- **Етап роботи з текстурами та матеріалами:** На цьому етапі команда використовує відповідні текстури та матеріали для точного відтворення поверхонь будівель і ландшафтів. Це включає моделювання матеріалів, таких як цегла, бетон, скло та метал, а також застосування відповідних текстур, таких як фарба, штукатурка та підлога. Текстури і матеріали повинні бути підібрані так, щоб точно відображати реальний вигляд кампусу.
- **Етап перенесення моделі в Unreal Engine:** На цьому етапі команда імпортує створені моделі в Unreal Engine, за потреба переробляє проблемні моделі використовуючи функціонал моделювання UE, налаштовує матеріали, освітлення, налаштовує колізії, додає персонажа.
- **Етап створення інтерактивних елементів:** На цьому етапі команда реалізовує всі необхідні інтерактивні елементи відповідно до вимог та стандартів.
- **Етап забезпечення якості:** На цьому етапі команда тестує середовище на точність і зручність використання, а також вносить будь-які необхідні зміни або корективи. Сюди входить перевірка на наявність помилок моделювання, таких як перекриття полігонів, інвертовані нормалі та відсутні текстури. Також важливо протестувати модель на зручність використання, наприклад, навігацію по моделі в реальному часі та рендеринг високоякісних зображень і анімації.

Кожен виконавець проекту має свою зону відповідальності:

- **Новий корпус**

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

- звукове оточення
- реалізація інтерактивного комп'ютера з можливістю виходу в інтернет через нього та анімацією відсунення стільця;

- відкриття-закриття дверей і вікон;
- увімкнення/вимкнення світла;
- реалізація механіки ліфта;
- камери безпеки та можливість їх перегляду на окремому моніторі;
- реалізація механіки бігу для основного персонажа;

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

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

3. Лаптев Богдан(фасад адмінбудівлі, частина 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 саява.

## ● Основний корпус

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

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

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

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

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

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

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

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

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

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