

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки


«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«__» грудня 2023 р.

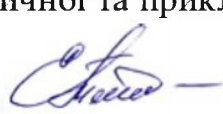
Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «Комп'ютерна модель віртуального музею з елементами віртуального
середовища»

Захищено на засіданні
Атестаційної комісії № 40
протокол № __ від __.12.2023 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ **СКОБ Ю. О.**

Виконав:
студентка 2 курсу, групи КІ– 61
за спеціальністю 123 – Комп'ютерна
інженерія.
Галузь знань: 12 – Інформаційні
технології
НАКОНЕЧНА Інна Михайлівна 

Керівник: д.т.н., с.н.с., професор
кафедри теоретичної та прикладної
системотехніки 
ТОЛСТОЛУЗЬКА Олена Геннадіївна

Рецензент: к.ф.-м.н., доцент, в.о.
завідувач кафедри електроніки і
управляючих систем
ХРУСЛОВ Максим Михайлович _____

Харків – 2023

АНОТАЦІЯ

Пояснювальна записка до магістерської атестаційної роботи складається зі вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. Загальний обсяг роботи складає 87 сторінки, із яких 64 сторінок основної частини з 37 рисунками, 31 найменуваннями списку використаних джерел та трьома додатками.

Метою кваліфікаційної роботи є підвищення якості освіти за рахунок використання комп'ютерної моделі віртуального музею, яка відображає історичні та культурні об'єкти України в інтерактивному віртуальному середовищі.

Актуальність проекту полягає у створенні унікального віртуального простору, який дозволяє детально вивчати експонати у форматі 360 градусів, забезпечуючи високий рівень інтерактивності та залученості користувачів. Особливість проекту полягає в інтеграції аудіогіда та текстової інформації, підтримка української та англійської мов, що робить музей інформаційно насиченим та доступним для широкої аудиторії.

Об'єктом дослідження є процес розробки віртуального музею як комплексного середовища для презентації історичних та культурних об'єктів. Предметом дослідження є методи та інструменти для створення інтерактивних віртуальних музеїв, зокрема, використання фотограмметрії для сканування об'єктів та їх подальше моделювання у програмному середовищі Blender та Unreal Engine 5.

Проблематика кваліфікаційної роботи концентрується на відсутності в Україні інтерактивних віртуальних музеїв, здатних забезпечити детальне вивчення експонатів. З огляду на сучасні виклики, зокрема воєнні дії та природні катастрофи, робота набуває особливої ваги у контексті збереження історичних та культурних пам'яток для майбутніх поколінь.

Ключові слова: Віртуальний музей, VR (віртуальна реальність), Unreal Engine, Фотограмметрія, Історичні та культурні об'єкти, Real-Time Ray Tracing, 3D моделювання.

ABSTRACT

The explanatory note to the master's thesis consists of an introduction, three chapters, conclusions, a list of references, and three appendices. The total volume of the work is 87 pages, of which 64 pages are the main part with 37 figures, 31 references, and three appendices.

The purpose of the qualification work is to improve the quality of education by using a computer model of a virtual museum that displays historical and cultural objects of Ukraine in an interactive virtual environment. This qualification work presents the development of a computer model of a virtual museum with elements of a virtual environment. The relevance of the project lies in creating a unique virtual space that allows for a detailed study of exhibits in a 360-degree format, providing a high level of interactivity and user engagement. The uniqueness of the project lies in the integration of audio guides and textual information, support for Ukrainian and English languages, making the museum information-rich and accessible to a wide audience. Support for Ukrainian and English languages expands the geography of users, allowing people from all over the world to learn about the cultural heritage of Ukraine.

The object of research is the process of developing a virtual museum as a complex environment for presenting historical and cultural objects. The subject of the study is methods and tools for creating interactive virtual museums, in particular, the use of photogrammetry for scanning objects and their subsequent modeling in Blender and Unreal Engine 5 software environments.

The problematic of the qualification work focuses on the lack of interactive virtual museums in Ukraine capable of providing a detailed study of exhibits. Given the current challenges, including military actions and natural disasters, the work acquires special importance in the context of preserving historical and cultural monuments for future generations. The scope of the developed virtual museum includes education, culture, and tourism.

Key words: Virtual museum, VR (virtual reality), Unreal Engine, Photogrammetry, Historical and cultural objects, Real-Time Ray Tracing, 3D modeling.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРОБКИ КОМП'ЮТЕРНОЇ МОДЕЛІ ВІРТУАЛЬНОГО МУЗЕЮ З ЕЛЕМЕНТАМИ ВІРТУАЛЬНОГО СЕРЕДОВИЩА	8
1.1 Сучасні методи розробки віртуальних музеїв	8
1.1.1 Огляд існуючих віртуальних музеїв і їхніх особливостей.	8
1.1.2 Використання VR в музейній сфері.	10
1.1.3 Інтерактивні можливості та їх вплив на досвід користувача.....	11
1.2 Технології віртуальної реальності (VR)	12
1.2.1 Основні види та принципи роботи VR.	12
1.2.2 Обладнання для VR та його характеристики.	14
1.2.3 Програмне забезпечення для створення VR-середовищ.	15
1.3 Методи сканування історичних та культурних об'єктів.	16
1.3.1 Існуючі методи сканування.	16
1.3.2 Фотограмметрія та лазерне сканування: основні принципи та порівняння.	19
1.3.3 Програмне забезпечення для обробки сканів.	21
1.4 Основні інструменти та можливості програми Blender для моделювання.	23
1.5 Unreal Engine як інструмент для розробки віртуальних середовищ.....	26
1.5.1 Основні компоненти та можливості Unreal Engine.	26
1.5.3 Real-Time Ray Tracing.	28
1.6 Мова C++ для написання скриптів.....	29
1.6.1 Вбудований функціонал C++ в Unreal Engine.	29

1.6.2 Основні бібліотеки для розробки віртуальних додатків.....	31
Висновки за розділом 1	34
РОЗДІЛ 2 РОЗРОБКА КОМП'ЮТЕРНОЇ МОДЕЛІ ВІРТУАЛЬНОГО МУЗЕЮ З ЕЛЕМЕНТАМИ ВІРТУАЛЬНОГО СЕРЕДОВИЩА	36
2.1 Розробка структури музею.....	36
2.1.1 Визначення структури музею.....	36
2.1.2 Розробка інтер'єру.....	38
2.1.3 Створення 3d моделей експонатів із сканів	41
2.2.2 Процес оптимізації та деталізації сканованих моделей.....	44
2.2.3 Імпортування та налаштування моделей у Unreal Engine	47
2.2 Розробка функціоналу віртуального музею за допомогою с++	48
2.2.1 Визначення функціональних можливостей користувача	48
2.3.1 Навігація та переміщення в середовищі VR	50
2.3.2 Інтерактивні елементи: вибір, перегляд та маніпуляція об'єктами... ..	52
2.3.3 Реалізація аудіо гіда та текстової інформації	55
Висновки за розділом 2	57
РОЗДІЛ 3 РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ	58
3.1 Рекомендації для Користувача з віртуальним шоломом	58
3.2 Рекомендації для Користувача без Віртуального шолома	62
2.3 Тестування віртуального музею.....	63
Висновки за розділом 3	65
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТКИ.....	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- VR — Virtual Reality
- AR — Augmented Reality
- GPU — Graphics Processing Unit
- CPU — Central Processing Unit
- LiDAR — Light Detection and Ranging
- ПЗ — програмне забезпечення
- DOTS — Data-Oriented Technology Stack
- UMG — Unreal Motion Graphics
- RTX — Ray Tracing Texel eXtreme
- RT Cores — Ray Tracing Cores
- HDR — High Dynamic Range
- UI — User Interface
- STL — Stereolithography
- PLY — Polygon File Format
- FBX — Filmbox

ВСТУП

Віртуальна реальність (VR) є однією з найбільш перспективних та інноваційних технологій сучасності, яка відкриває нові можливості для різних сфер людської діяльності. Однією з таких сфер є культурна спадщина, яка є важливим джерелом історичної пам'яті, національної ідентичності та освіти. За допомогою VR можна створити віртуальні музеї, які зберігають, відтворюють та популяризують культурні та історичні об'єкти, надаючи користувачам можливість відвідувати та вивчати їх без фізичних обмежень. Віртуальні музеї також можуть сприяти розвитку віртуального туризму, який є альтернативою традиційному туризму, особливо в умовах кризи, пов'язаної з пандемією COVID-19 та війною на сході України.

Метою цієї кваліфікаційної роботи є підвищення якості освіти за рахунок використання комп'ютерної моделі віртуального музею, яка відображає історичні та культурні об'єкти України в інтерактивному віртуальному середовищі. Цей проект має велике значення для освіти, оскільки він дозволяє користувачам отримати доступ до унікальних та цінних експонатів, які можуть бути втрачені або пошкоджені внаслідок війни або інших факторів. Також цей проект сприяє збереженню та популяризації культурної спадщини, забезпечуючи можливість відвідування та вивчення об'єктів з усього світу, сприяючи туризму, національній ідентичності та історичній пам'яті.

Особливістю цього проекту є можливість детального вивчення кожного експонату в 360 градусів, наявність аудіогіда та текстової інформації, що виділяє його серед існуючих віртуальних музеїв України, підтримка двох мов: англійської та української робить проект доступним для людей зі всього світу.

Практична значимість роботи полягає в тому, що розроблений віртуальний музей може бути використаний як освітній ресурс, який дозволяє людям з усього світу досліджувати українську історію та культуру без фізичного відвідування музею, а також як засіб збереження та популяризації культурної спадщини, особливо в умовах війни та кризи.

РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРОБКИ КОМП'ЮТЕРНОЇ МОДЕЛІ ВІРТУАЛЬНОГО МУЗЕЮ З ЕЛЕМЕНТАМИ ВІРТУАЛЬНОГО СЕРЕДОВИЩА

1.1 Сучасні методи розробки віртуальних музеїв

1.1.1 Огляд існуючих віртуальних музеїв і їхніх особливостей.

Віртуальні музеї в Україні представляють собою різноманітну колекцію культурних і історичних експозицій, доступних онлайн. Ці музеї пропонують унікальні можливості для ознайомлення з українською спадщиною, особливо важливі в умовах обмеженого доступу через війну або пандемію [1].

Нижче наведен приклад деяких музеїв:

Дніпропетровський художній музей має 3D-тур експозицією, створений спільно з Google Ukraine, та віртуальну виставку, присвячену Миколі Погрібняку. Це дозволяє користувачам ознайомитись з великою колекцією музею, однак музей має обмежену взаємодію з експонатами.

Музей грошей Національного банку України Проводить віртуальні тури з гідом, демонструючи понад 3,5 тис. експонатів у форматі 3D.

Музеї просто неба включають скановані сцени з різних куточків України, де можна віртуально відвідати територію музеїв і дізнатися про історію і культуру регіонів [2].

Карпатські дерев'яні церкви представлені в онлайн-форматі, що дозволяє знайомитися з об'єктами світової спадщини ЮНЕСКО. Проте, цифрові зображення можуть не передати всі деталі архітектури і внутрішнього оздоблення.

Оперні театри України надають можливість віртуально відвідати пишні зали і закуліси, проте такий формат не може передати енергію живих виступів та атмосферу, що створюється присутністю глядачів.

Острозький замок віртуальний тур Острозького краєзнавчого музею, де можна побачити колекції археології, живопису, іконопису, скульптури [3].

Віртуальні музеї України, які ми розглядали, створені на основі сканованих зображень. Користувачі в цих музеях можуть лише переміщатися по просторах музею, але не мають можливості взаємодії з експонатами(приблизити, обертати і тд). Це створює обмеження в досвіді віртуального відвідування.

Додатково, не всі віртуальні музеї надають детальну інформацію про свої експонати. Це обмежує освітню та культурну цінність таких віртуальних турів, оскільки відвідувачі не отримують повного контексту або історичного змісту представлених предметів.

Ще одним важливим аспектом є відсутність підтримки англійської мови у багатьох з цих віртуальних музеїв, що обмежує їх доступність для іноземних відвідувачів. Це є значним недоліком, так як музеї можуть відвідувати люди з усього світу, і мовна доступність є ключовою для міжнародного культурного обміну.

Також однією з проблем є незручна система навігації в деяких з цих музеїв. Для переходу між різними кімнатами чи експозиціями, користувачам потрібно відкривати окремі вікна або переходити на відокремлені веб-сторінки.

Відсутність можливості використання технології віртуальної реальності (VR) є ще однією значним недоліком існуючих віртуальних музеїв. Без VR відвідувачі не можуть повністю зануритися в середовище музею, що знижує відчуття присутності та залученості.

У підсумку, хоча віртуальні музеї України пропонують унікальну можливість ознайомитися з культурною спадщиною країни, вони все ще мають ряд обмежень, таких як відсутність інтерактивності, обмежена інформація про експонати та недостатня мовна підтримка, відсутність можливості використання технології віртуальної реальності, незручна система навігації, які можуть впливати на загальний досвід відвідування.

1.1.2 Використання VR в музейній сфері.

Використання віртуальної реальності (VR) у музейній сфері представляє собою важливий напрямок розвитку технологій, який суттєво змінює спосіб, яким відвідувачі сприймають та взаємодіють з музейними колекціями. Ця інноваційна технологія відкриває нові можливості в освіті, зокрема для школярів, студентів та інших освітніх груп в Україні та по всьому світу.

Віртуальні музеї можуть запропонувати більшу доступність, ніж фізичні музеї, оскільки їх можна відвідувати будь-коли та будь-де, використовуючи будь-який пристрій, наприклад смартфон, планшет, ноутбук або гарнітуру VR/AR [4].

Однією з ключових переваг є інтерактивність віртуальної реальності. Користувачі можуть взаємодіяти з віртуальними об'єктами та оточенням, що сприяє збагаченню освітнього досвіду та розвитку критичного мислення. Віртуальні музеї можуть забезпечувати різні рівні інтерактивності, такі як навігація, масштабування, обертання, анотації та маніпуляції. Віртуальні музеї можуть підвищити освітню цінність цифрового контенту та віртуального середовища, надаючи додаткову інформацію, а також інтерактивні функції взаємодії з об'єктами такі як вікторини, ігри, симуляції та сценарії, щоб полегшити навчання. Це робить навчання та дослідження культурної спадщини більш захопливими та доступними для освітніх закладів та громадськості [5].

Крім того, VR дозволяє зберігати та відновлювати об'єкти культурної спадщини в цифровому форматі, що вирішує проблему збереження та документування цінних артефактів під час війн та природних катастроф.

Використання VR також сприяє популяризації музеїв, допомагаючи залучати нових відвідувачів та створюючи музейні колекції більш доступними для широкої аудиторії. Ця інноваційна технологія відкриває нові горизонти для навчання, досліджень та популяризації культурної спадщини, створюючи їх доступними для освітніх установ, музеїв та суспільства в цілому.

1.1.3 Інтерактивні можливості та їх вплив на досвід користувача

Сучасні технології віртуальної реальності надають користувачам досвід, який радикально відрізняється від традиційного відвідування музеїв [6]. Інтерактивність в віртуальних музеях не тільки поглиблює занурення, але й робить досвід більш особистісним та адаптованим до індивідуальних потреб відвідувача [7].

У віртуальних музеях користувач може вільно переміщатися між експонатами, незалежно від їх розміщення у реальному просторі. Це дає можливість розглядати об'єкти з різних ракурсів і дозволяє відвідувачам вибирати свій власний маршрут.

Віртуальні музеї дозволяють користувачам збільшувати об'єкти, що допомагає вивчати дрібні деталі, які можуть бути недоступними для огляду в реальному музеї. Також можливість обертати експонат на 360 градусів надає повне уявлення про його форму та структуру.

Віртуальний музей може супроводжуватися аудіо гідами, що дозволяє відвідувачам глибше зануритися в історію та значення кожного експоната. Текстова інформація може бути інтегрована безпосередньо в віртуальне середовище, забезпечуючи швидкий доступ до додаткової інформації.

Віртуальний музей надає можливість відвідувачам досліджувати і взаємодіяти з експонатами на більш особистому рівні. Завдяки інтерактивності відвідувачі не тільки спостерігають, але приймають в процесі пізнання. Такий досвід може бути набагато більш занурюючим та пам'ятним, порівняно з традиційним відвідуванням музею.

Відмінність від реального музею:

У порівнянні з реальними музеями, віртуальні пропонують необмежені можливості для взаємодії та дослідження. З одного боку, реальний музей пропонує автентичний досвід, де можна відчувати матеріали та атмосферу простору. З іншого боку, віртуальний музей видаляє фізичні обмеження,

дозволяючи користувачам досліджувати та взаємодіяти з експонатами у способи, які були б неможливими у реальному середовищі [8].

1.2 Технології віртуальної реальності (VR)

1.2.1 Основні види та принципи роботи VR.

- Віртуальна реальність (VR) – це технологія, яка створює імітацію середовища, яку можна відчувати за допомогою гарнітури VR або інших пристроїв. Існують різні типи VR, які відрізняються за рівнем занурення, взаємодії та реалізму, який вони пропонують. Ось деякі з найпоширеніших типів VR:

- VR без занурення (Non-immersive VR) – це тип VR, який використовує екран комп'ютера або проектор для відображення віртуального середовища, але не ізолює користувача від реального світу. Користувач може керувати деякими аспектами симуляції, наприклад персонажем або транспортним засобом, але середовище не реагує на дії або рухи користувача.

- VR з повним зануренням (Fully immersive VR) — це тип віртуальної реальності, який використовує гарнітуру VR та інші пристрої, як-от контролери, рукавички або трекери, для створення реалістичного та інтерактивного віртуального середовища. Користувач повністю ізолюваний від реального світу і може бачити, чути та відчувати симуляцію так, ніби вона справжня. Середовище реагує на дії та рухи користувача в реальному часі.

- Напівзанурений VR (Semi-immersive VR) – це тип VR, який використовує великий екран або купол для відображення віртуального середовища, але не повністю ізолює користувача від реального світу. Користувач може взаємодіяти з деякими елементами симуляції, такими як симулятор польоту або симулятор водіння, але середовище не повністю реагує на дії та рухи користувача.

- Доповнена реальність (AR) – це тип віртуальної реальності, який використовує пристрій, як-от смартфон, планшет або пару окулярів, для накладання цифрової інформації чи зображень на реальний світ. Користувач

може бачити як реальні, так і віртуальні елементи одночасно та взаємодіяти з ними. Віртуальні елементи можуть покращувати або змінювати реальний світ різними способами.

- Спільна VR – це тип віртуальної реальності, який дозволяє кільком користувачам спільно використовувати одне й те саме віртуальне середовище та взаємодіяти з ним локально чи віддалено. Користувачі можуть спілкуватися один з одним за допомогою голосу, тексту або жестів і можуть співпрацювати над різними завданнями чи діями. Віртуальне середовище може бути без занурення, повністю або напівзануреним, залежно від пристроїв і програмного забезпечення, що використовується [9].

Як саме працює VR з повним зануренням. Центральний процесор (CPU) та графічний процесор (GPU) комп'ютера або VR-пристрою розраховують і рендерять два незалежні зображення для кожного ока, які потім відображаються на дисплеї VR-пристрою. Ці зображення забезпечують стереоскопічний вид. Вони злегка відмінні через паралакс, і коли вони комбінуються в мозку користувача, вони створюють ілюзію тривимірного простору.

Системи VR використовують сенсори (зазвичай інерційні мірні одиниці та оптичні сенсори) для відстеження рухів голови користувача в реальному часі. Ця інформація потім використовується для оновлення перспективи користувача в віртуальному середовищі. Сучасні VR-системи часто включають в себе контролери рук або інші засоби вводу, які дозволяють користувачам взаємодіяти з віртуальним середовищем.

Однією з ключових метрик якості VR є затримка між моментом, коли користувач здійснює рух, і моментом, коли відбувається відповідна зміна в зображенні. Висока затримка може призвести до дискомфорту або розладів відчуття.

Для глибшої імерсії VR-системи використовують просторовий звук, який змінюється в залежності від положення та орієнтації користувача.

Інша ключова особливість VR – це можливість динамічно взаємодіяти з оточенням, не тільки спостерігати. Це здійснюється через алгоритми фізики, які моделюють реалістичні взаємодії між об'єктами.

1.2.2 Обладнання для VR та його характеристики.

Віртуальна реальність (VR) вимагає специфічного обладнання для надання користувачеві можливості зануритися у віртуальне середовище та інтерактивно спілкуватися з ним. Розглянемо основні типи обладнання для VR, їх характеристики.

Для створення та використання віртуальної реальності потрібно специфічне обладнання, яке включає в себе комп'ютер, гарнітуру віртуальної реальності (шолом) та контролери.

1. Комп'ютер:

- Процесор (CPU): Рекомендується використовувати мінімум Intel Core i5-4590 або AMD Ryzen 5 1500X.
- Відеокарта (GPU): NVIDIA GTX 1060/AMD Radeon RX 480 або краща. Для високоякісного рендеринга в реальному часі рекомендуються карти з класу NVIDIA RTX 2080 або вище.
- Оперативна пам'ять (RAM): Мінімум 8GB, але рекомендується 16GB або більше.
- Порти: Мінімум 1x USB 3.0 або новіший порт, а також порти HDMI 1.3.

2. Гарнітура VR (Oculus Go):

- Дисплей: Fast-Switch LCD з роздільною здатністю 2560 x 1440.
- Частота оновлення: 60Hz (з деякими додатками, що підтримують 72Hz).
- Процесор: Qualcomm Snapdragon 821.
- Оперативна пам'ять: 3GB RAM.
- Зберігання: Версії на 32GB або 64GB.

- Слідкування: 3-ступінчасте слідкування.
- Батарея: Приблизно 2 години використання.

3. Контролер(Oculus Go):

- Слідкування: 3-ступінчасте слідкування, синхронізоване зі шоломом.
- Кнопки: Сенсорний пульт з тригером, сенсорною поверхнею, кнопкою "Назад" та кнопкою меню.

Сучасне обладнання для VR надає можливість не просто дивитися на 3D-середовище, але й взаємодіяти з ним. Шоломи з високим роздільним здатністю та високою частотою оновлення забезпечують гладкий та якісний візуальний досвід. Контролери дозволяють користувачам взаємодіяти з об'єктами у віртуальному світі, надаючи додатковий рівень занурення.

1.2.3 Програмне забезпечення для створення VR-середовищ.

При створенні віртуальних музеїв одним із ключових етапів є вибір відповідного програмного забезпечення для реалізації VR-середовища. Дане забезпечення дозволяє розробникам створювати інтерактивні тривимірні моделі, симулювати реальний світ і забезпечувати користувача досвідом поглибленого занурення.

Основні програмні рішення:

Unreal Engine - двигун є одним із найпопулярніших виборів для створення VR-середовищ. Він пропонує потужний набір інструментів для рендерингу, фізики та інтерактивності. Додатково, інтеграція з C++ дозволяє створювати налаштовані скрипти та логіку для віртуальних експонатів та багато іншого.

Unity - популярний ігровий рушій, який активно використовується для розробки VR-застосунків. Unity підтримує велику кількість VR-платформ і пропонує широкий набір інструментів для розробки. Завдяки інтеграції з C#, розробники можуть легко додати інтерактивність і динаміку у віртуальних музеях [10].

Blender VR - це розширення для популярного безкоштовного програмного забезпечення для 3D моделювання Blender, яке дозволяє створювати VR-контент.

A-Frame - веб-фреймворк для створення віртуальних реальностей, який дозволяє розробникам створювати VR-досвід, що працює безпосередньо у веб-браузері.

Amazon Sumerian - Облачна платформа від Amazon для створення і розгортання VR, AR та 3D-додатків без необхідності спеціалізованого програмування [11].

Програмне забезпечення для створення VR-середовищ є ключовим компонентом при розробці віртуальних музеїв. Воно дозволяє не тільки відтворювати тривимірні об'єкти або сцени, але й забезпечує користувача можливістю взаємодії. Кожне з цих програмних рішень має свої особливості, переваги та недоліки. Вибір залежить від специфічних потреб проекту, бюджету, та технічних знань розробника.

1.3 Методи сканування історичних та культурних об'єктів.

1.3.1 Існуючі методи сканування.

Сканування історичних та культурних об'єктів є важливою частиною збереження та дослідження спадщини. Для цього застосовуються різноманітні методи, кожен з яких має свої особливості та призначення:

Фотограмметрія — Цей метод використовує фотографії для створення 3D-моделей об'єктів. Він базується на принципі стереоскопії, який дозволяє визначати просторові координати об'єкта з двох або більше зображень, зроблених з різних точок зору. Фотограмметрія може бути наземною або аерофотозйомкою в залежності від висоти, з якої робляться фотографії. Цей метод застосовується для документування ландшафтів, археологічних розкопок, геологічних структур, міських об'єктів тощо. Наприклад, фотограмметрія використовувалася для створення 3D-моделі Великої піраміди Хеопса в Єгипті.

Лазерне сканування (LIDAR) — Лазерне сканування: Цей метод використовує лазерний промінь для вимірювання відстані до поверхні об'єкта. Він дозволяє отримати високоякісні та точні 3D-моделі об'єктів різних розмірів та форм. Лазерне сканування може бути стаціонарним або мобільним залежно від типу сканера та об'єкта. Цей метод застосовується для документування архітектурних пам'яток, скульптур, реліквій тощо. Наприклад, лазерне сканування використовувалося для створення 3D-моделі собору Нотр-Дам після пожежі у 2019 році. На Рис 1.1. зображена технологія лазерного сканування.

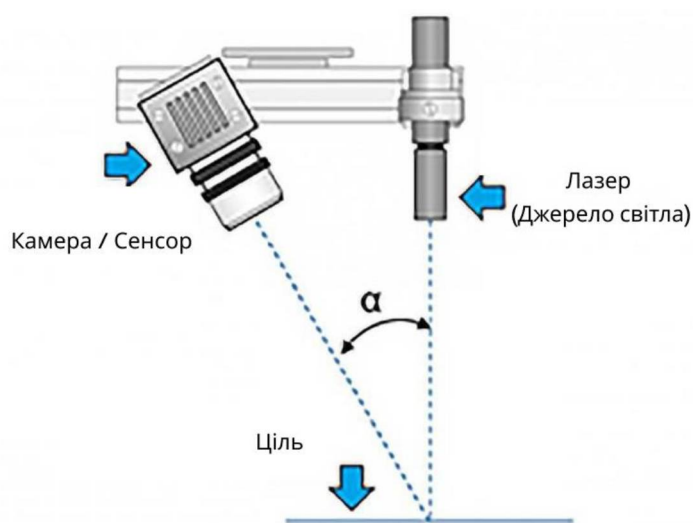


Рис 1.1 — Технологія лазерного сканування

Термографія — Цей метод використовує інфрачервоне випромінювання для відображення різниці температур на поверхні об'єкта. Термографія дозволяє виявляти приховані дефекти, тріщини, порожнини, вологість, втрату тепла тощо. Термографія може бути контактною або безконтактною в залежності від того, чи торкається термографічний пристрій об'єкта. Цей метод використовується для діагностики стану історичних об'єктів, оцінки їх теплової ефективності, виявлення прихованих деталей тощо. Наприклад, термографія використовувалася для виявлення невідомих фресок у церкві святого Миколая у Львові.

Структуроване світло — це метод тривимірної мікроскопії, який використовує проекцію світлових візерунків на об'єкт і аналізує викривлення

цих візерунків, щоб визначити форму та рельєф об'єкта. На Рис 1.2 зображена технологія сканування структурованим світлом. Цей метод можна застосовувати для дослідження різних матеріалів, у тому числі біологічних, хімічних, металевих та ін. Структуроване світло дозволяє отримати високу роздільну здатність і точність вимірювань, а також зменшити вплив зовнішніх факторів, таких як освітлення, тіні, відблиски тощо.

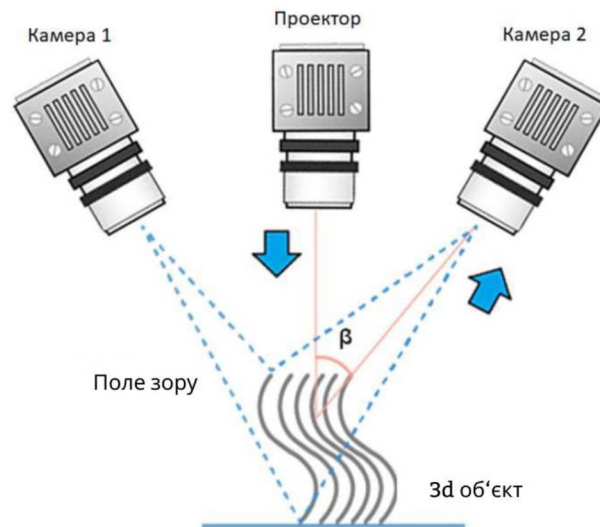


Рис 1.2 — Технологія сканування структурованим світлом

3D-мікроскопія — це метод 3D-сканування, який використовує мікроскоп і джерело світла для збільшення та освітлення поверхні об'єкта. Мікроскоп може бути оптичним або електронним, а джерело світла може бути когерентним або некогерентним. Мікроскоп і об'єкт або фіксуються, або переміщуються відносно один одного, щоб отримати кілька зображень об'єкта в різних фокальних площинах або під різними кутами. Програма об'єднує зображення та створює 3D-модель об'єкта з високою роздільною здатністю та точністю. Використовується для дуже малих об'єктів, таких як монети або ювелірні вироби.

Мультиспектральне сканування — це метод 3D-сканування, який використовує камеру та джерело світла для фіксації відбиття та флуоресценції об'єкта під різними довжинами хвиль світла. Джерело світла може бути як природним, так і штучним, а камера може бути монохромною або кольоровою. Камера та об'єкт фіксуються або переміщуються відносно один одного, щоб

отримати кілька зображень об'єкта в різних діапазонах спектру, наприклад у видимому, інфрачервоному або ультрафіолетовому діапазонах. Програма аналізує зображення та створює 3D-модель об'єкта з покращеним кольором і контрастністю. Мультиспектральне сканування можна застосовувати для вивчення різноманітних об'єктів культурної спадщини, таких як картини, рукописи, текстиль, монети, кераміка тощо. Цей метод допомагає перевірити автентичність, походження, історію, техніку, стиль, реставрацію та консервацію об'єктів. Одним із прикладів сканування історичних та культурних об'єктів за допомогою мультиспектрального сканування є проект *Multispectral Imaging of Cultural Heritage: A Review*, який надає комплексний огляд стану -мистецькі техніки та застосування мультиспектрального зображення у сфері культурної спадщини. Цей проект охоплює принципи, методи, пристрої та проблеми мультиспектрального зображення, а також аналіз, обробку та візуалізацію мультиспектральних даних [12].

1.3.2 Фотограмметрія та лазерне сканування: основні принципи та порівняння.

Нижче описано принцип дії двох основних методів(фотограмметрія, лазерне 3D сканування).

Фотограмметрія - це метод вимірювання геометричних характеристик об'єктів за допомогою фотографій.

Принцип роботи сканування об'єктів методом фотограмметрії базується на використанні двовимірних зображень для отримання тривимірної інформації про об'єкт або територію[13]. Кожний знімок представляє собою центральну проєкцію об'єктів на площину. Коли один і той же об'єкт зображено на двох різних фотографіях з різних ракурсів, утворюється стереоскопічний ефект, дозволяючи виділити просторові характеристики об'єкта. Використовуючи математичні методика, здійснюється відновлення просторової геометрії об'єкта на основі координат точок на фотографіях. Перед цим процесом важливо провести орієнтацію знімків, визначивши взаємне розташування зображень та їх

положення відносно об'єкта. Сучасна фотограмметрія активно інтегрує цифрові технології, що забезпечують автоматизацію процесів вимірювання, обробки та відтворення тривимірних моделей об'єктів.

Основні характеристики:

- Дозволяє отримувати текстуровані 3D моделі.
- Потребує кількох фотографій об'єкта з різних точок зору.
- Залежить від якості камери та освітлення.
- Вимагає спеціалізованого програмного забезпечення для обробки.

Лазерне сканування або LiDAR (Light Detection and Ranging) є методом вимірювання відстані від сканера до об'єкта за допомогою випромінювання лазерних імпульсів і аналізу відбитого випромінювання.

Принцип роботи сканування об'єктів методом лазерного 3D-сканування полягає в використанні лазерного випромінювача, який відправляє лазерний промінь до поверхні об'єкта. Коли промінь вдаряється об об'єкт, він відбивається назад до детектора на 3D-сканері. За допомогою вимірювання часу, що минув від випуску променя до отримання відбитого променя, сканер розраховує відстань до конкретної точки на поверхні об'єкта. При цьому сканер здійснює множинні вимірювання, досліджуючи різні частини об'єкта, тим самим формуючи тривимірне об'ємне зображення. Ці дані потім об'єднуються в одну інтегровану 3D-модель, яка детально відтворює форму та характеристики поверхні об'єкта. Завдяки високій точності лазерних вимірювань, метод 3D-сканування забезпечує дуже детальне і надійне відображення об'єктів у тривимірному просторі[14].

Основні характеристики:

- Дозволяє отримувати високу точність геометрії об'єкта.
- Швидкість вимірювань вища, ніж у фотограмметрії.
- Не залежить від освітлення.
- Зазвичай не передає текстур або кольори об'єкта.

Порівняння фотогеометрії та лазерне 3D сканування:

Точність: Лазерне сканування зазвичай надає вищу геометричну точність, порівняно з фотограмметрією.

Текстури: Фотограмметрія автоматично надає текстури для 3D моделей, тоді як LiDAR зазвичай цього не робить.

Освітлення: LiDAR не залежить від освітлення, тоді як фотограмметрія вимагає адекватного освітлення для отримання якісних зображень.

Обладнання: Лазерні сканери зазвичай дорожчі та більш складні, ніж камери для фотограмметрії.

Обробка: І фотограмметрія, і лазерне сканування вимагають спеціалізованого ПЗ для обробки і створення 3D моделей, але процеси можуть суттєво відрізнитися.

При розгляді методів сканування для зйомки експонатів музею, фотограмметрія виявляється більш підходящою альтернативою порівняно з лазерним 3D скануванням (LiDAR).

Фотограмметрія, як правило, є менш вартісною з точки зору обладнання та обробки. Камери, потрібні для зйомки, часто є менш вартісними в порівнянні з високоякісними лазерними сканерами. Фотограмметрія автоматично захоплює текстури об'єкта, що забезпечує високий рівень деталізації та колірної відповідності експонатам музею. Метод фотограмметрії дозволяє легко адаптуватися до різних розмірів та форм експонатів, не вимагаючи спеціалізованого обладнання для кожного типу об'єкта. Фотограмметричне обладнання та програмне забезпечення зазвичай є більш інтуїтивними для користувачів без спеціальної технічної підготовки.

1.3.3 Програмне забезпечення для обробки сканів.

Програмне забезпечення для 3D-сканування – це тип програмного забезпечення, який дозволяє створювати 3D-моделі об'єктів реального світу, фіксуючи їх форму та зовнішній вигляд за допомогою камери, сканера або датчика.

Нижче наведено декілька прикладів програмного забезпечення для 3D-сканування, яке можна використовувати[15].

Програмне забезпечення для фотограмметрії:

Scanect - це програмне забезпечення 3D-сканера для ПК і Mac, яке сумісне з багатьма 3D-сканерами, включаючи Xbox Kinect. Він простий у використанні та має інструменти редагування, які дозволяють маніпулювати відсканованими моделями. Він безкоштовний для некомерційного використання та коштує 129 доларів США за професійну версію.

Meshroom - це 3D-сканер із відкритим кодом і програмне забезпечення для реконструкції, яке використовує фотограмметрію для створення 3D-моделей із набору зображень. Він заснований на AliceVision, системі комп'ютерного зору. Він сумісний з Windows і Linux і пропонує графічний інтерфейс користувача та інтерфейс командного рядка.

Elcovision - це 3D-сканер і програмне забезпечення для реконструкції, яке використовує фотограмметрію та лазерне сканування для створення 3D-моделей із зображень і хмар точок. Він сумісний з Windows і пропонує графічний інтерфейс користувача та інтерфейс командного рядка. Він має такі функції, як автоматичне калібрування, орієнтація зображення та створення поверхні.

MicMac - це програмне забезпечення для 3D-сканера та реконструкції з відкритим кодом, яке використовує фотограмметрію для створення 3D-моделей із набору зображень. Він сумісний з Windows, Linux і Mac і пропонує інтерфейс командного рядка та графічний інтерфейс користувача. Він має багато опцій і параметрів, які дозволяють налаштувати процес[16].

Програмне забезпечення для структурованого світла:

Reconfusion Pro - це програмне забезпечення 3D-сканера, яке використовує структуроване світло для створення високоякісних 3D-моделей у реальному часі. Він підтримує кілька датчиків, таких як Intel RealSense, Asus Xtion і Orbbec Astra. Він має розширені функції, такі як автоматичне вирівнювання, заповнення отворів і накладання текстури.

Програмне забезпечення для лазерного сканування:

iWitnessPro - це 3D-сканер і програмне забезпечення для реконструкції, яке використовує фотограмметрію для створення 3D-моделей із зображень. Він сумісний з Windows і пропонує графічний інтерфейс користувача. Він призначений для судової експертизи та реконструкції нещасних випадків. Він має такі функції, як автоматичне вимірювання, калібрування камери та 3D-візуалізація.

Найкраще програмне забезпечення - Meshroom. Він безкоштовний і має відкритий вихідний код, що означає, що можна використовувати його без будь-яких витрат або ліцензійних обмежень. Він сумісний з Windows і Linux, які є найпоширенішими операційними системами для персональних комп'ютерів. Він заснований на AliceVision, яка є найсучаснішою структурою фотограмметрії, яка використовує розширені алгоритми та функції, такі як багаторкурсне стерео, щільне зіставлення та текстуровання сітки. Він має зручний інтерфейс, який дозволяє легко регулювати параметри та переглядати хід процесу реконструкції. Meshroom має плагін який працює з Blender. Він може створювати високоякісні та реалістичні 3D-моделі зі звичайних зображень, які можуть бути корисними для різних застосувань, таких як 3D-друк, анімація або віртуальна реальність.

[17]

1.4 Основні інструменти та можливості програми Blender для моделювання.

Blender — це безкоштовне програмне забезпечення для створення 3D із відкритим вихідним кодом, яке пропонує ряд інструментів і функцій моделювання для проектування та створення цифрових моделей. Деякі з основних інструментів моделювання та функцій Blender:

Полігональне Моделювання

Маніпуляції з Вершинами, Ребрами та Гранями: Основа полігонального моделювання в Blender полягає у маніпуляціях з базовими елементами моделі:

вершинами, ребрами та гранями. Це дозволяє створювати та змінювати форми об'єктів.

Вершини (Vertices) - це точки у тривимірному просторі, які визначають положення кутів полігонів.

Ребра (Edges) – це лінії, що з'єднують вершини. Вони формують каркас полігонів.

Полігон (Faces) - це площини, обмежені ребрами, які разом утворюють поверхню об'єкта.

Blender дозволяє користувачам швидко та ефективно отримувати доступ до різних функцій моделювання за допомогою комбінацій клавіш. Наприклад, користувачі можуть натиснути G, щоб захопити, R, щоб повернути, і S, щоб масштабувати об'єкт або частину сітки[18].

Інструмент Knife tool – цей інструмент дозволяє користувачеві розрізати та розділяти грані та краї сітчастого об'єкта рухом від руки. Користувач також може обмежити розріз певним кутом чи віссю або прив'язати його до існуючих вершин. Інструмент Knife tool може бути корисним для створення отворів, деталей або нестандартних форм на сітці

Екструдювання (Extrusion) – цей інструмент дозволяє користувачеві створювати нову геометрію з наявних граней і країв сітчастого об'єкта, розширюючи їх уздовж напрямку. Користувач також може масштабувати, обертати або переміщувати екстудовану геометрію в інтерактивному або числовому режимі.

Інструмент Bevel tool – цей інструмент дозволяє користувачеві створювати округлені краї та кути на сітчастому об'єкті, додаючи геометрію вздовж вибраних країв або вершин. Користувач може регулювати кількість, сегменти, профіль та інші параметри фаски в інтерактивному або числовому режимі. Інструмент скосу може бути корисним для створення плавних переходів, фасок або скруглень на сітці[19].

Loop Cut and Slide – дозволяє додавати нові ряди полігонів на модель для деталізації та кращого контролю форми.

Модифікатори: Модифікатори — це автоматичні операції, які впливають на об'єкт неруйнівним способом, тобто оригінальна геометрія об'єкта зберігається та може бути відредагована в будь-який час. Модифікатори можна застосовувати до об'єктів сітки, кривої, поверхні, тексту, решітки та арматури, їх можна складати та комбінувати для досягнення складних ефектів. Деякі з поширених модифікаторів:

- Модифікатор Subdivision Surface. Цей модифікатор поділяє грані об'єкта на менші грані, створюючи більш гладкий і органічний вигляд.
- Модифікатор Boolean. Цей модифікатор виконує логічні операції (об'єднання, різниця або перетин) між двома об'єктами сітки, у результаті чого створюється нова сітка, яка поєднує або віднімає об'єми вихідних об'єктів.
- Модифікатор Mirror. Цей модифікатор створює дзеркальну копію об'єкта вздовж однієї або кількох осей, дозволяючи користувачеві моделювати симетричні фігури з удвічі меншими зусиллями. Користувач може вибрати вісь віддзеркалення, об'єкт, який використовуватиметься як центр віддзеркалення, а також параметри об'єднання або відсікання вершин у дзеркальній площині.

UV Unwrapping – Процес відображення тривимірного сітчастого об'єкта на двовимірну площину, що дозволяє користувачеві застосовувати текстури зображення або малювати безпосередньо на сітку.

Blender підтримує N-Gon, які є багатокутниками з більш ніж чотирма вершинами або ребрами. N-Gons можуть бути корисними для створення складних форм і топології без додавання непотрібної геометрії або підрозділів. Blender також надає інструменти для перетворення N-Gons на чотирикутники чи трикутники, або навпаки, залежно від потреб моделі.

Інструменти UV Sculpt – це інструменти, які дозволяють користувачеві захоплювати, стискати та розслабляти/згладжувати UVs, так само як у режимі Object Sculpt Mode, що дає більше контролю. і гнучкість відносно UV map [21].

1.5 Unreal Engine як інструмент для розробки віртуальних середовищ

1.5.1 Основні компоненти та можливості Unreal Engine.

Unreal Engine це потужний двигун для розробки віртуальних середовищ, який надає розробникам широкий спектр інструментів і можливостей для створення інтерактивних дослідницьких віртуальних музеїв з використанням віртуальної реальності.

Unreal Engine заснована на компонентній архітектурі, де кожен об'єкт в сцені представлений як сукупність компонентів. Це можуть бути моделі, світла, камери, звуки та інші елементи. Однією з ключових особливостей Unreal Engine є Blueprint — візуальна система програмування, яка дозволяє розробникам створювати функціонал без прямого написання коду. Blueprints забезпечують можливість швидкої прототипізації та ітерації проектів[22].

PBR (Фізично-базований Рендеринг): Unreal Engine використовує систему PBR для створення реалістичних матеріалів, яка враховує взаємодію світла з поверхнею на основі його фізичних властивостей[23].

Sequencer - це потужний інструмент для анімації та кінематографії, який дозволяє розробникам створювати кінематографічні сцени високої якості безпосередньо в редакторі.

World Building: Unreal Engine має набір інструментів для "будівництва світу", які включають Landscape (для створення теренів), Foliage (для розміщення рослинності) та інші інструменти для деталізації сцен.

VR та AR Підтримка: Unreal Engine пропонує рідну підтримку для більшості VR та AR пристроїв, що робить його оптимальним вибором для розробки віртуальних музеїв.

Мультиплатформність: Unreal Engine підтримує велику кількість платформ, включаючи PC, консолі, мобільні пристрої та інше, дозволяючи розробникам досягти широкої аудиторії.

Інтеграція з Третіми Сервісами: Unreal Engine має багатий маркетплейс та підтримку багатьох плагінів, які можна інтегрувати, щоб розширити функціональність двигуна.

Niagara Particle System: Ця система частинок є наступником класичної системи Cascade та дозволяє розробникам створювати набагато більш динамічні, інтерактивні та візуально привабливі ефекти. Завдяки Niagara, можливо симулювати інтерактивні реакції на змінене довкілля або взаємодію користувача.

Ray Tracing: Однією з новітніх особливостей Unreal Engine є підтримка рейтрейсингу (ray tracing), технології, яка дозволяє досягти неймовірного реалізму у відображенні світла, тіней та відблисків.

Chaos Physics and Destruction: Chaos — це система фізики та руйнувань, яка дозволяє додавати до сцен реалістичні фізичні взаємодії та динамічні руйнування.

Data-Oriented Technology Stack (DOTS): Unreal Engine працює над оптимізацією своїх систем для більш швидкої та ефективної роботи на сучасному обладнанні, і DOTS є частиною цього підходу.

Unreal Motion Graphics (UMG): Система для створення інтерфейсів користувача від простих меню до комплексних інтерактивних дашбордів.

Мережева Архітектура: Unreal Engine має потужну мережеву архітектуру, яка дозволяє створювати многокористувацькі віртуальні середовища, включаючи онлайн музеї або виставки.

Технічна Підтримка та Спільнота: Завдяки великій та активній спільноті, розробники можуть знаходити відповіді на свої питання, отримувати ресурси або навчальні матеріали. Крім того, Epic Games активно підтримує Unreal Engine, регулярно випускаючи оновлення та нові можливості.

Ці компоненти та можливості роблять Unreal Engine потужним інструментом для створення віртуальних музеїв з елементами віртуального середовища, дозволяючи розробникам створювати реалістичні та інтерактивні дослідницькі додатки для віртуальної реальності, які забезпечують незабутній досвід користувачам.

1.5.3 Real-Time Ray Tracing.

Real-Time Ray Tracing (RTX) є передовою технологією рендерингу, яка дозволяє симулювати поведінку світлових променів в реальному часі для створення візуально реалістичних зображень та відображень. Ця технологія виграє на актуальному ринку віртуальних музеїв і інших віртуальних середовищах завдяки своїй здатності до точного моделювання відбиття світла, тіней та інших оптичних ефектів. Нижче розглянуті ключові аспекти роботи та технології Real-Time Ray Tracing.

Алгоритм рейського відслідковування:

Основним принципом роботи RTX є алгоритм рейського відслідковування (ray tracing), який базується на симуляції світлових променів, що поширюються від камери до сцени та зворотно. Кожен промінь відбивається та розсіюється на поверхнях об'єктів, враховуючи їхні матеріали та оптичні властивості. Рейське відслідковування забезпечує точну моделювання відбиття світла, тіней, відображення та просвічування, створюючи реалістичний образ сцени[24].

Технологія променевих променів (RT Cores):

RTX використовує прискорювачі променевих променів (RT Cores), які спеціально розроблені для рейського відслідковування. Ці RT Cores виконують складні обчислення для розрахунку взаємодії світлових променів та об'єктів у сцені, що прискорює процес рендерингу.

RTX дозволяє точно розраховувати відбиття світла, тіні та реалістичні тіні, навіть у складних сценах. Це створює реалістичний вигляд об'єктів та допомагає створювати іммерсивний досвід для користувачів віртуальних музеїв.

RTX також дозволяє підтримувати інші графічні ефекти, такі як реалістичні відображення об'єктів у відраженнях, глобальне освітлення, підтримку HDR (High Dynamic Range) і багато інших, що підвищують якість віртуальних музеїв.

Загалом, технологія Real-Time Ray Tracing є важливою для реалістичного рендерингу в реальному часі у віртуальних середовищах, таких як віртуальні

музеї, і дозволяє створювати досвід, який наближається до реального світу завдяки точному моделюванню світлових явищ та оптичних ефектів.

1.6 Мова C++ для написання скриптів

1.6.1 Вбудований функціонал C++ в Unreal Engine.

Unreal Engine, розроблений компанією Epic Games, заслужено здобув своє місце на вершині ігрової індустрії завдяки його вражаючим графічним можливостям, гнучкості та досконалої підтримці C++ програмування. Цей двигун, протягом років, став більше, ніж просто інструментом для розробки ігор. Він є платформою, яка поєднує в собі артистику з технічною потужністю[25].

Центральною частиною архітектури Unreal Engine є її фреймворк Object. Він закладає фундамент для об'єктно-орієнтованого програмування, дозволяючи розробникам створювати власні класи, базуючись на передбачуваних і добре структурованих базових класах Unreal, як-от Actor, GameMode, PlayerController, і так далі. Ця система успадкування і поліморфізму стає ключовою для створення багаторівневих і гнучких ігрових механік. Приклад на Рис. 1.3.

```
UCLASS()
class MYGAME_API AMyActor : public AActor
{
    GENERATED_BODY()
public:
    AMyActor();
    virtual void BeginPlay() override;
    virtual void Tick(float DeltaTime) override;
    UPROPERTY(VisibleAnywhere)
    UStaticMeshComponent* MyMesh;
};
```

Рис 1.3 — Клас Unreal Engine з перевизначеними методами

- UCLASS(): Це макрос Unreal Engine, який вказує, що наступний клас є частиною Unreal Object System. Це необхідно для інтеграції класу з системами Unreal, такими як Blueprint.
- class MYGAME_API AMyActor : public AActor: Тут ми визначаємо новий клас під назвою AMyActor, який успадковується від AActor. AActor є базовим класом для всіх об'єктів, які можна розміщувати в світі Unreal.

- `GENERATED_BODY()`: Цей макрос автоматично генерує тіло класу, яке необхідне для інтеграції з Unreal Object System.
- `AActor()`;: Це декларація конструктора для нашого класу.
- `virtual void BeginPlay() override`;: Декларація методу `BeginPlay()`, який автоматично викликається, коли об'єкт `AActor` стає активним у грі. Ключове слово `override` вказує, що ми плануємо перевизначити цей метод у нашому класі.
- `virtual void Tick(float DeltaTime) override`;: Декларація методу `Tick()`, який викликається кожного кадру для цього `AActor`. `DeltaTime` представляє час (в секундах) з моменту останнього виклику `Tick`.
- `UPROPERTY(VisibleAnywhere)`: Це інший макрос Unreal, який вказує, як змінна повинна бути представлена в редакторі Unreal. В даному випадку, `VisibleAnywhere` говорить, що змінна буде видимою в редакторі, але не може бути змінена.
- `UStaticMeshComponent* MyMesh`;: Це декларація вказівника на компонент статичного меша. Це може бути використано для того, щоб додати 3D модель до нашого `AActor`.

Особливістю роботи з C++ в Unreal є використання специфічних макросів, таких як `UCLASS()`, `UPROPERTY()` та `UFUNCTION()`. Ці макроси розширюють традиційний C++ код, дозволяючи більш глибокої інтеграції з системою `Blueprint`[26]. Це створює мост між програмістами та дизайнерами, спрощуючи процес створення та ітерації ігрового контенту.

Життєвий цикл `AActor`'а в Unreal Engine – це ключова концепція, яка дозволяє розробникам контролювати поведінку об'єктів протягом їх існування в ігровому світі. З `BeginPlay()`, що відзначає початок життя `AActor`'а, до `Tick()`, який регулює його поведінку кожного кадру, та до `EndPlay()`, який сигналізує про його завершення – розуміння цих стадій дозволяє створювати динамічні та відгуківі ігрові механіки.

Важливим аспектом роботи в Unreal є взаємодія між C++ та Blueprint. Blueprints – це візуальна система програмування, створена для тих, хто може не мати технічного фону. Однак, завдяки можливостям інтеграції, розробники можуть легко поєднувати високопродуктивний код C++ з гнучкістю Blueprints. Приклад функції, написаної на C++ на Рис. 1.4.

```
UFUNCTION(BlueprintCallable, Category="My Functions")
void MyFunction();
```

Рис 1.4 — Макрос BlueprintCallable в Unreal Engine

- `UFUNCTION(BlueprintCallable, Category="My Functions")`: Це макрос, який показує, що наступна функція може бути викликана з Blueprint. Також вказується категорія для цієї функції в редакторі Blueprint.
- `void MyFunction();`: Це декларація функції, яка може бути реалізована в коді C++

Memory Management в Unreal Engine – це ще одна важлива частина. З автоматичним збором сміття та вбудованими засобами оптимізації, розробники можуть зосередитись на логіці гри, не турбуючись про витрати пам'яті[27].

Однією з ключових переваг Unreal є його модульність. Це не тільки стосується архітектури коду, але і системи плагінів. Розробники можуть легко імпортувати, модифікувати або створювати плагіни, що розширюють можливості двигуна.

1.6.2 Основні бібліотеки для розробки віртуальних додатків

Unreal Engine - це потужний та універсальний ігровий двигун, який дозволяє розробникам створювати захоплюючі віртуальні додатки за допомогою мови програмування C++. Unreal Engine надає базові бібліотеки та фреймворки для розробки різних аспектів віртуальних додатків, таких як графіка, звук, фізика, мережева взаємодія, інтерфейс користувача, штучний інтелект та інше[28]. Деякі з основних бібліотек та фреймворків такі:

Core - це базова бібліотека, яка надає основні функції, такі як управління пам'яттю, контейнери, рядки, математика, журналювання та інше. Вона також визначає Об'єктну Модель Unreal, яка є фундаментом системи відображення Unreal Engine. Бібліотека Core є незалежною від будь-якої платформи чи функціоналу двигуна і може бути використана в будь-якому C++ проєкті.

Для створення нового класу C++, який успадковується від класу Unreal Engine, такого як Actor чи UObject, ви можна використовувати Class Wizard c++ в Unreal Editor 5. Цей інструмент автоматично створює файли заголовка та файлу джерела для вашого класу і реєструє його в системі відображення Unreal Engine. Також можна використовувати макроси UCLASS, UPROPERTY, UFUNCTION та UENUM, щоб викласти свій клас, властивості, функції та переліки для доступу до них в Unreal Editor і Blueprints[29].

Наприклад, для оголошення класу C++, який може бути використаний в UE5, необхідно використовувати макрос UCLASS (UCLASS - це макрос Unreal Engine, який дозволяє визначити клас, який буде використовуватися в двигуні)
Рис. 1.5.

```
// Простий клас C++, який може бути використаний в UE5
UCLASS()
class AMyActor : public AActor
{
    GENERATED_BODY()
public:
    // Конструктор, який встановлює значення за замовчуванням для властивостей цього актора
    AMyActor();
    // Функція, яка виводить повідомлення до журналу виведення
    UFUNCTION()
    void SayHello();
};
```

Рис 1.5 — Клас актора з конструктором та функцією в Unreal Engine

У цьому коді макрос UCLASS() вказує Unreal Engine, що клас AMyActor буде використовуватися в рамках двигуна. Клас успадковується від AActor, що робить його актором в грі. Макрос GENERATED_BODY() генерує необхідний код для Unreal Engine.

CoreUObject - ця бібліотека реалізує Об'єктну Модель Unreal, яка дозволяє серіалізувати, збирати сміття, реплікувати та використовувати об'єкти в

Blueprints. Вона також надає функції, такі як делегати, переліки, структури та макроси відображення. Бібліотека CoreUObject залежить від бібліотеки Core і є базою для всіх інших бібліотек Unreal Engine.

Engine - ця головна бібліотека надає основний функціонал Unreal Engine, такий як рендеринг, звук, фізика, анімація, ігровий фреймворк, підсистема мережевої взаємодії та інше. Вона також визначає різні класи, які використовуються для створення та обробки Акторів, Компонентів, Матеріалів, Текstur, Звуків та інших ресурсів віртуального додатка. Бібліотека Engine залежить від бібліотек Core та CoreUObject і є базою для більшості ігрових бібліотек.

Розглянемо на Рис. 1.6 створення власного класу актора в C++. Для створення нового класу актора в C++, ви повинні успадкувати його від класу AActor та перевизначити віртуальні функції.

```
// Власний клас актора, який успадковується від AActor
class AMyCustomActor : public AActor
{
    GENERATED_BODY()
public:
    // Конструктор, який встановлює значення за замовчуванням для властивостей цього актора
    AMyCustomActor();
    // Викликається при запуску гри або при створенні актора
    virtual void BeginPlay() override;
    // Викликається кожен кадр гри
    virtual void Tick(float DeltaTime) override;
};
```

Рис 1.6 — Кастомний клас актора в Unreal Engine

У цьому коді клас AMyCustomActor успадковується від AActor, а потім перевизначає дві віртуальні функції, BeginPlay() і Tick(), які викликаються при запуску гри та кожен кадр гри відповідно.

Slate - бібліотека реалізує Slate UI Framework, який є високопродуктивною, гнучкою та платформонезалежною системою інтерфейсу користувача. Вона дозволяє розробникам створювати багатфункціональні та реактивні інтерфейси користувача за допомогою C++ або UMG (Unreal Motion Graphics UI Designer). Бібліотека Slate залежить від бібліотек Core, CoreUObject та Engine і використовується для створення різних елементів інтерфейсу, таких як віджети,

меню, діалоги, панелі і багато інших. Для створення нового віджета Slate, який успадковується від існуючого віджета Slate, такого як SButton чи STextBlock, можна використовувати макроси SLATE_BEGIN_ARGS, SLATE_END_ARGS та SLATE_ATTRIBUTE для визначення аргументів і атрибутів вашого віджета. Також використовувати макроси SNew, SAssignNew і SVerticalBox (або SHorizontalBox), щоб створювати і призначати нові віджети як дочірні елементи вашого віджета[30].

UMG: Ця бібліотека реалізує UMG UI Designer, який є візуальним інструментом для створення інтерфейсів користувача за допомогою віджетів Slate. Вона дозволяє розробникам проектувати та створювати інтерфейси користувача за допомогою Blueprints або C++, підтримує функції, такі як анімації, прив'язки, стилі та локалізація. Бібліотека UMG залежить від бібліотек Core, CoreUObject, Engine та Slate і використовується для створення різних ресурсів інтерфейсу, таких як шаблони віджетів, шрифти та курсори.

Для створення нового віджета UMG, який успадковується від існуючого віджета UMG, такого як UButton чи UTextBlock, можна використовувати редактор UMG в Unreal Editor для візуального проектування та створення прототипів вашого віджета. Також можна використовувати макроси UCLASS, UPROPERTY, UFUNCTION та UENUM, щоб викласти свій віджет, властивості, функції та переліки.

Висновки за розділом 1

Під час дослідження, для розробки віртуального музею було обрано спеціалізоване програмне забезпечення, яке найкраще відповідає вимогам проекту. Blender та Unreal Engine 5 були обрані як основні інструменти для розробки моделі віртуального музею, завдяки їхнім розширеним можливостям моделювання, візуалізації та створення інтерактивних VR-середовищ .

Було з'ясовано, що фотограмметрія є найефективнішим методом для сканування експонатів, оскільки вона дозволяє точно зафіксувати деталі та текстури об'єктів та для неї не потрібен великий бюджет. Для перетворення

зображень у 3D-об'єкти була обрана програма Meshroom, яка ефективно використовує фотограмметричні методи для створення деталізованих 3D-моделей.

Також було проведено дослідження наявного обладнання для VR, з метою визначення найкращих варіантів для віртуального музею.

Аналіз існуючих віртуальних музеїв дозволив виявити їхні сильні та слабкі сторони, що стало важливим внеском у процес планування та розробки власного проекту віртуального музею.

Наостанок, дослідження показало, що використання сучасних методів розробки, включаючи VR, 3D моделювання, і впровадження інтерактивних елементів, суттєво покращує досвід користувачів та розширює доступність культурних і історичних об'єктів, особливо в умовах в часи кризи. Віртуальні музеї стають ключовими у збереженні та популяризації культурної спадщини, забезпечуючи можливість відвідування та вивчення об'єктів з усього світу, сприяючи освіті, туризму, збереженню національної ідентичності та історичної пам'яті.

РОЗДІЛ 2 РОЗРОБКА КОМП'ЮТЕРНОЇ МОДЕЛІ ВІРТУАЛЬНОГО МУЗЕЮ З ЕЛЕМЕНТАМИ ВІРТУАЛЬНОГО СЕРЕДОВИЩА

2.1 Розробка структури музею

2.1.1 Визначення структури музею

Під час планування і концепції музею було прийнято рішення обрати історичний стиль. Це означає, що музей спрямовувався на представлення історичних подій, культурних артефактів та важливих аспектів минулого.

У музеї було заплановано розмістити історичні експонати, такі як артефакти, речові свідчення, старовинні предмети побуту, архівні документи та інші об'єкти, які ілюструють історію та культурну спадщину минулих епох. Ці експонати мали бути ретельно відібрані та документовані, щоб передати їхню цінність та інформацію віртуальним відвідувачам. Крім того, було заплановано розміщення картин, які зображують важливі історичні події, портрети видатних осіб того часу та твори українських художників, які віддавали дань минулим епохам через своє мистецтво. Ці картини мали створити атмосферу та допомогти відтворити враження від певного історичного періоду.

Пріоритет у створенні віртуального історичного музею віддавався тим експонатам, які перебувають у суспільному надбанні та мають важливе значення для культурної спадщини. Для цього було проведено дослідження архівів культурної та історичної спадщини, пошук інформації про музеї та їхні експонати.

Для додавання детального опису самої моделі віртуального музею з використанням IDEF0, потрібно виокремити ключові функції та процеси, які були реалізовані в проекті. Ось основні аспекти, які можуть бути включені в модель:

1. Визначення та Аналіз Вимог:

- Функція: Визначення основних вимог до віртуального музею.

- Входи: Інформація про історичні та культурні об'єкти, вимоги до віртуального музею.
- Виходи: Список вимог до музею, включаючи інтерфейс, функціональність, доступність.

2. Розробка Дизайну та Архітектури:

- Функція: Створення дизайну та архітектури віртуального музею.
- Входи: Список вимог, зразки дизайну, технологічні інструменти (Blender, Unreal Engine 5).
- Виходи: Проект дизайну та архітектури віртуального музею.

3. 3D-Моделювання та Текстурування:

- Функція: Створення 3D-моделей об'єктів музею та їх текстурування.
- Входи: Фотографії об'єктів, програмне забезпечення для 3D-моделювання (Blender, Meshroom).
- Виходи: Готові 3D-моделі та текстури об'єктів.

4. Реалізація Інтерактивності:

- Функція: Додавання інтерактивних елементів до віртуального музею.
- Входи: Проект дизайну, програмне забезпечення (Unreal Engine 5, C++).
- Виходи: Інтерактивні елементи (навігація, аудіогіди, текстова інформація).

5. Тестування та Оцінка:

- Функція: Перевірка та оцінка роботи віртуального музею.
- Входи: Готовий віртуальний музей, тестові сценарії.
- Виходи: Звіт про тестування, виявлені проблеми та рекомендації.

Кожна з цих функцій детально описана з використанням діаграми IDEF0

Рис.2.1.

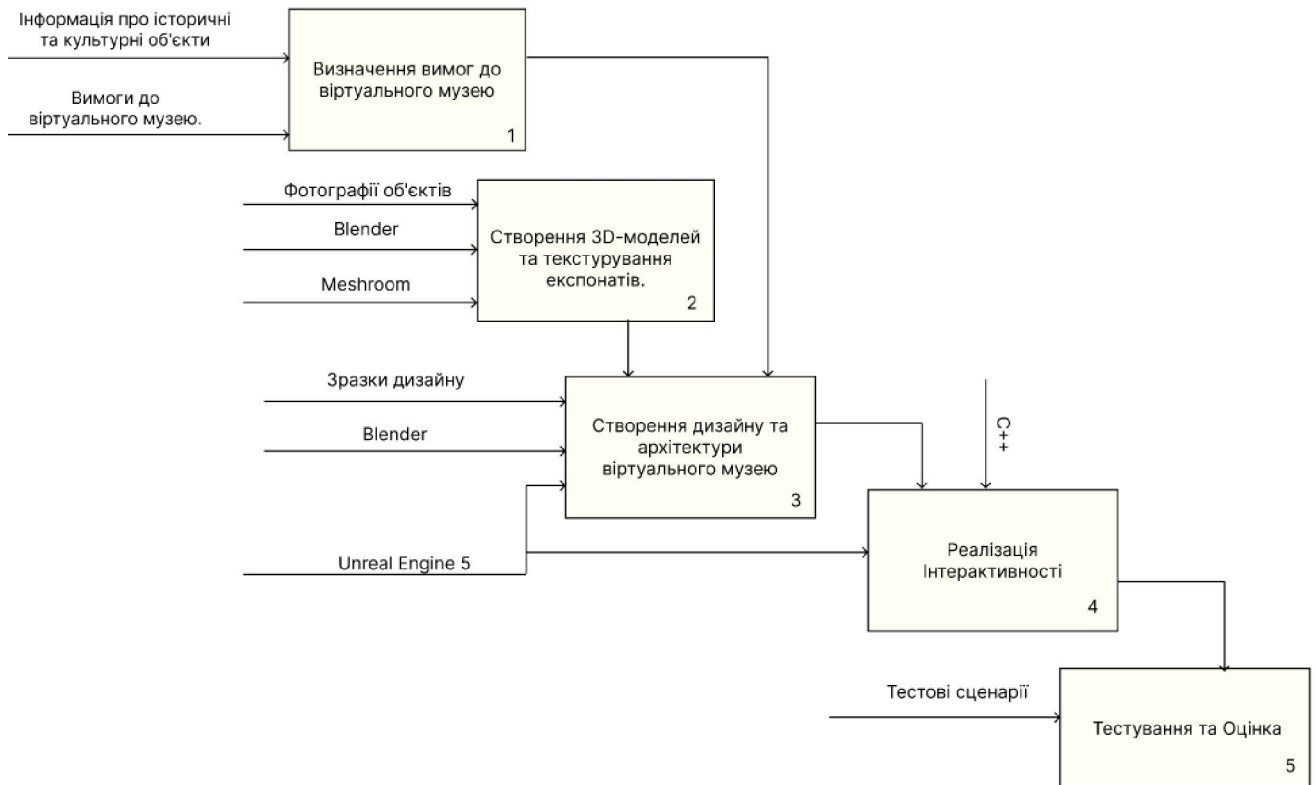


Рис 2.1 — Діаграма IDEF0

2.1.2 Розробка інтер'єру

Перш ніж почати моделювати музей в тривимірному просторі, потрібно намалювати на аркуші як буде виглядати кімнати музею Рис. 2.2, це дасть уявлення як він має виглядати. Надалі ми будемо відштовхуватись від нього.

Для створення основної геометрії кімнат музею використовувалась програма Blender, яка є безкоштовним і потужним інструментом для 3D-моделювання, анімації, рендерингу та інших задач. Для початку в сцену додали меш куба, який буде основою кімнати. Далі необхідно перейти у режим Edit Mode, щоб можна було редагувати геометрію куба. Вибрали верхню грань куба, і зменшили висоту кімнати, щоб зробити грань на рівні з нижньою гранню куба. Потрібно вибрати всі грані куба, і зробити їх внутрішніми, щоб побачити всередину кімнати і додати матеріали та освітлення. Для цього були використані такі інструменти, як Face Select, Extrude, Loop Cut, Bevel, Knife, Bridge Edge

Loops, Flip тощо . Додавання дверей та вікон вимагало створення отворів у відповідних гранях куба.

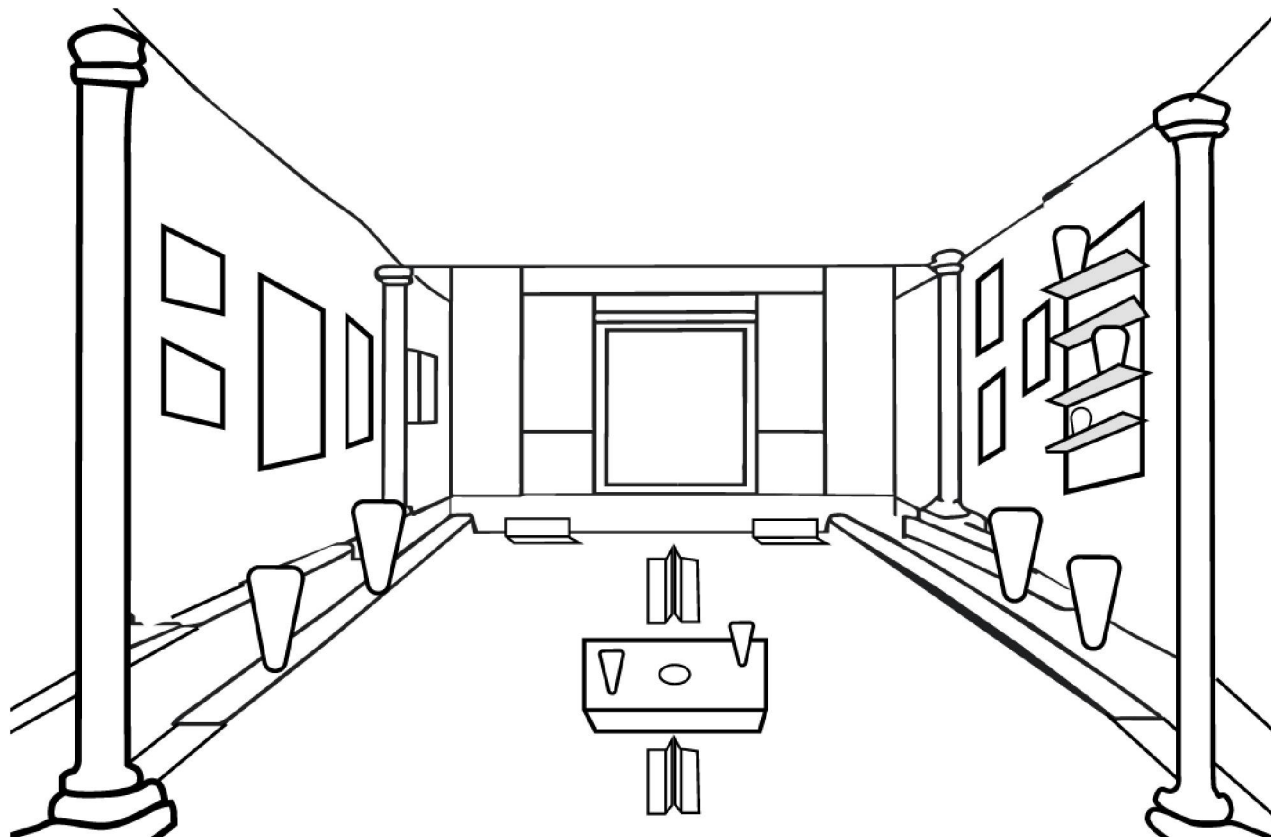


Рис 2.2 — Малюнок кімнати майбутнього музею

Ці отвори були створені за допомогою інструментів, таких як Inset, Delete, Faces тощо. Далі додавали деталі, такі як меблі та декорації Рис 2.3, для цього використовувалися різні геометричні примітиви, які потім були перетворені та налаштовані на відповідний вигляд та розмір. Також використовувалися різні модифікатори для створення складних форм та додавання деталей. Детально про всі використанні інструменти було описано в першому розділі, де розглядались можливості Blender.

Перш ніж додати матеріали до об'єктів потрібно зробити розгортку, це можна зробити в режимі UV Editing. Для розгортки об'єкта використовувались два способи:

Автоматична розгортка. Це найшвидший і найпростіший спосіб розгорнути об'єкт, але він може дати небажані результати, такі як спотворення, перекриття, нерівномірність тощо. Використовувався для простих форм.

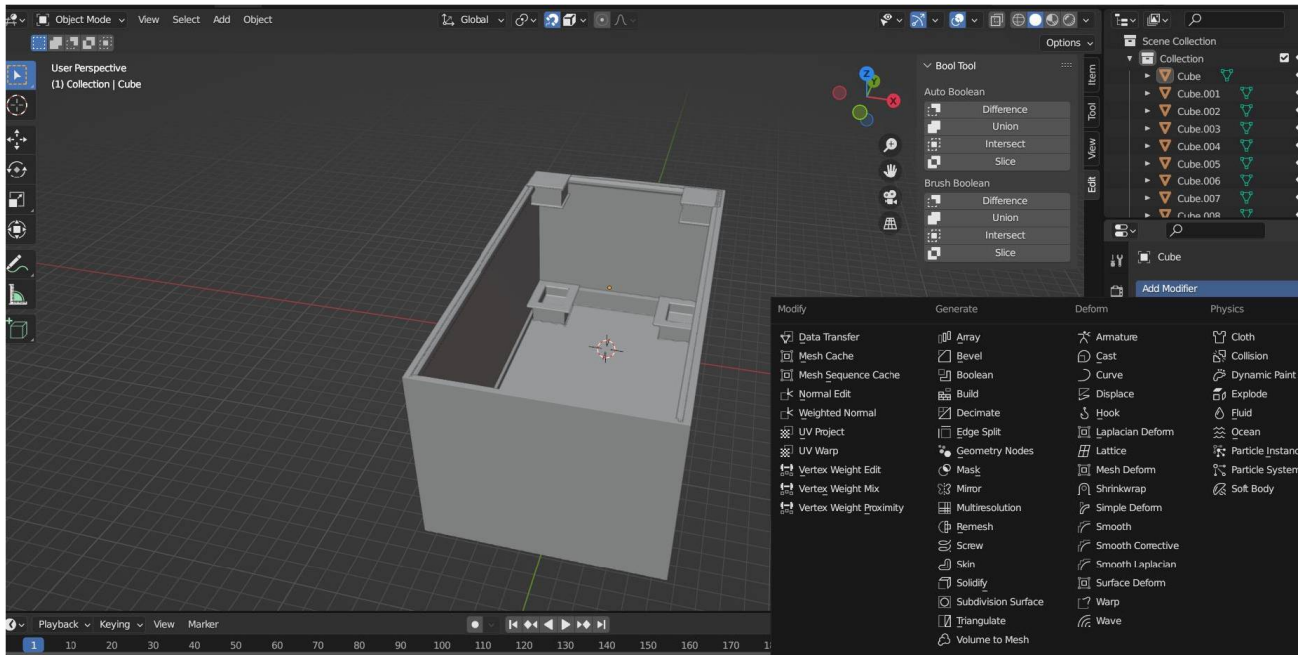


Рис 2.3 — Моделювання кімнати музею

Для автоматичної розгортки були вибрані всі грані об'єкта, натиснувши клавішу A, і натиснути клавішу U, щоб відкрити меню UV Mapping. Там ми виберемо один з варіантів, таких як Smart UV Project, Cube Projection, Cylinder Projection, Sphere Projection тощо, в залежності від форми об'єкта. Ми також налаштували параметри розгортки, такі як Angle Limit, Island Margin, Area Weight тощо.

Ручна розгортка. Це більш трудомісткий і складний спосіб розгорнути об'єкт, але він дає більш контрольовані і якісні результати, такі як мінімізація спотворень, оптимізація простору, збереження пропорцій тощо. Ідально підходить для складних форм. Для ручної розгортки обрали грані об'єкта, які повинні бути розгорнуті, і натиснемо клавішу U, щоб відкрити меню UV Mapping. Там вибрали варіант Unwrap, щоб розгорнути грані відповідно до їхньої форми. Також були використані такі інструменти, як Mark Seam, Clear Seam, Live Unwrap, Pin, Stitch, Minimize Stretch, Average Islands Scale тощо, щоб поліпшити розвертку.

Для створення матеріалів для всіх об'єктів, використовується режим Shading та редактор Node Editor, де були створені складні матеріали з різних вузлів, таких як Principled BSDF, Image Texture, Mix Shader, Bump, Noise, Voronoi

тощо . Ми налаштуємо параметри вузлів, такі як Base Color, Metallic, Roughness, Normal, Displacement тощо, щоб досягти бажаного ефекту.

2.1.3 Створення 3d моделей експонатів із сканів

За допомогою Meshroom, програма яку було описано вище, можна створити 3d моделі із сканів, для віртуального музею.

Перше, що необхідно, зробити серію фотографій експонатів Рис 2.4. Необхідно забезпечити хороше освітлення, фотографувати з різних кутів, охоплюючи кожну частину об'єкта, щоб забезпечити точну реконструкцію.

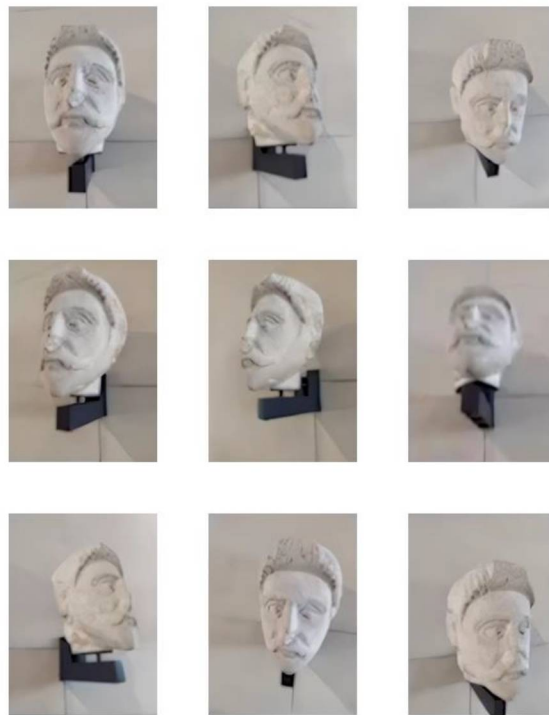


Рис 2.4 — Серія фотографій експоната

Далі необхідно запусити Meshroom і створити новий проект, після чого перетягнути фотографії на панель зображень Рис 2.5. Можна використовувати кнопку «Імпортувати зображення», щоб переглядати файли.

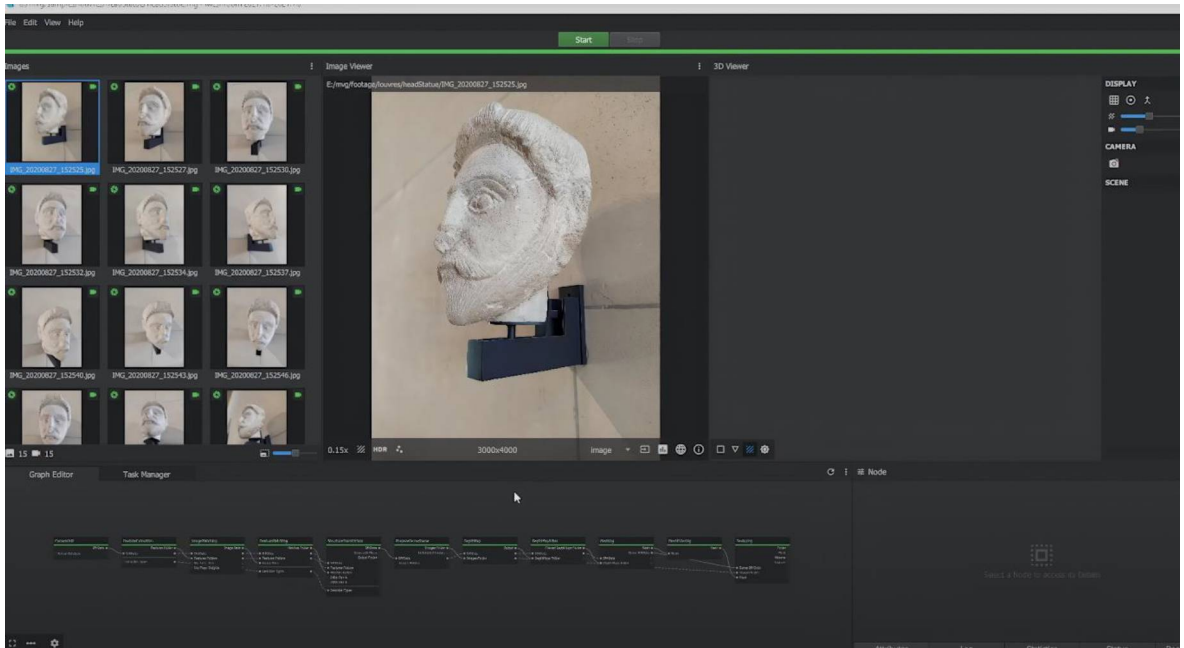


Рис 2.5 — Інтерфейс програми Meshroom

Meshroom автоматично визначає спільні точки між різними фотографіями та створює "хмару точок" Рис 2.6. Кожен вузол відповідає етапу обробки, такому як калібрування камери, виділення ознак або створення сітки.

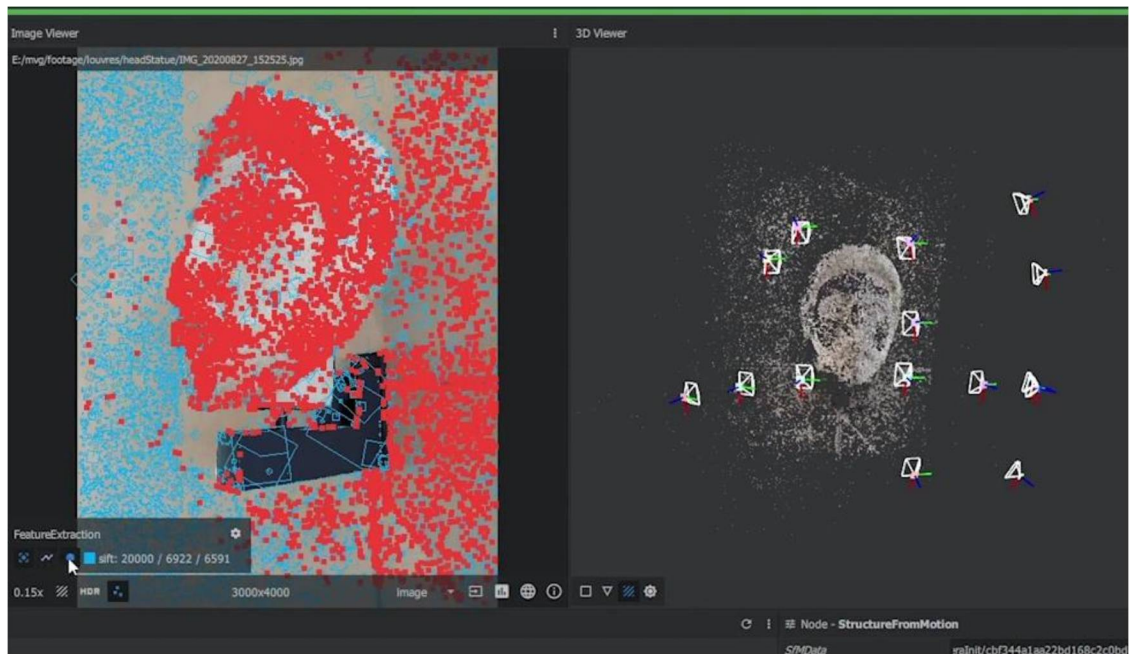


Рис 2.6 — "Хмара точок"

Можна налаштувати параметри кожного вузла, клацнувши його та відредагувавши значення на панелі «Атрибути».

Після цього програма конвертує хмару точок у сітку, формуючи базову 3D-модель об'єкта. Meshroom запускатиме кожен вузол послідовно та відобразатиме хід виконання та результати в засобі 3D-перегляду. Залежно від кількості та якості фотографій цей процес може тривати від кількох хвилин до кількох годин.

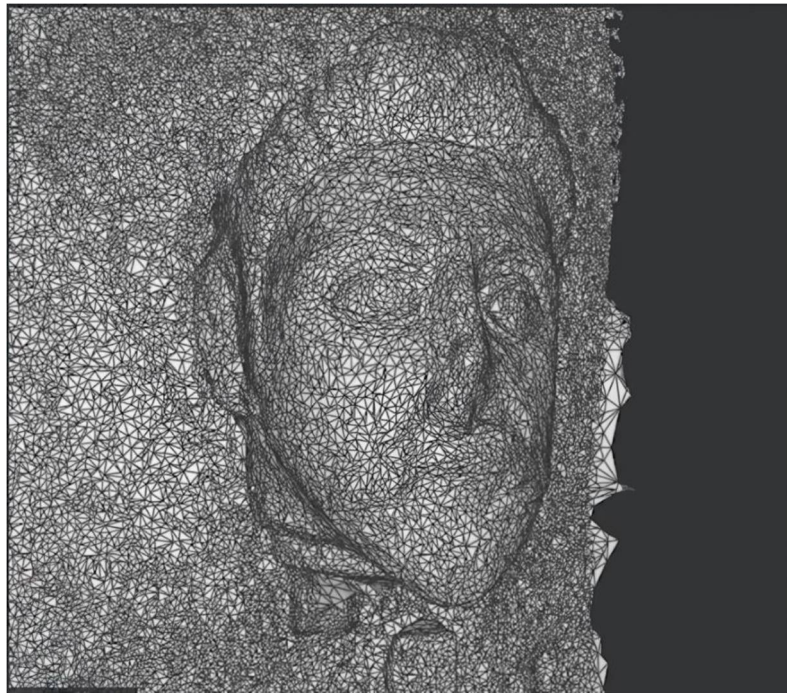


Рис 2.7 — Сітка готової 3d моделі

Як видно на зображенні Рис.2.7 сітка має недоліки, та багато зайвого, для того щоб отримати якісну модель її необхідно відредагувати та оптимізувати сторонніх програмах таких як 3dsmax, Blender і т. ін., про це буде описано в інших підрозділах.

Щоб зробити модель реалістичнішою, необхідно застосовувати текстури, використовуючи оригінальні фотографії Рис 2.8. Meshroom проектує ці текстури на сітку, імітуючи кольори та деталі реального об'єкта. На Рис 2.9 продемонстрована готова 3d модель.

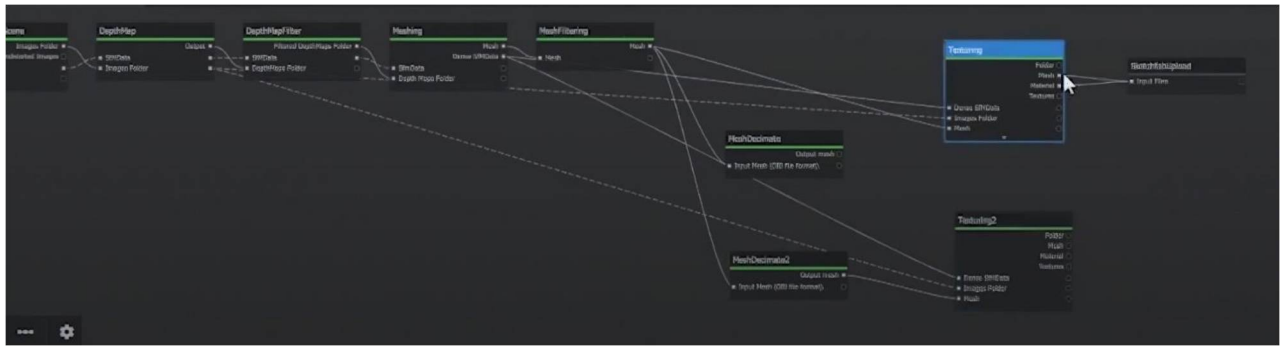


Рис 2.8 — Налаштування текстур

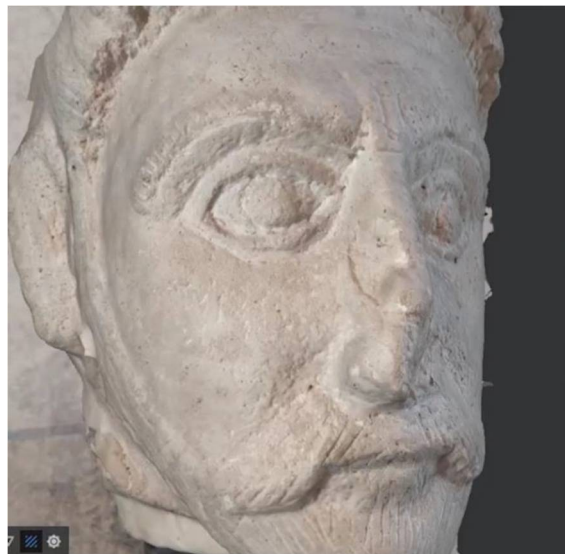


Рис 2.9 — Готова 3d модель

Щоб експортувати 3D-модель, необхідно натиснути правою кнопкою миші на вузлі Meshing і вибрати Export Mesh, або за допомогою спеціального плагіну експортувати модель у Blender.

2.2.2 Процес оптимізації та деталізації сканованих моделей.

Процес оптимізації та деталізації відсканованих моделей у Blender включає кілька ключових кроків. Ці кроки спрямовані на вдосконалення необроблених даних, отриманих із 3D-сканування.

Перше що необхідно зробити, імпорт відсканованої моделі. Blender підтримує різні формати файлів, такі як OBJ, STL, PLY тощо.

Другий крок полягає в очищенні відсканованої моделі, яка може мати шуми, дірки, артефакти або небажані частини Рис 2.10.

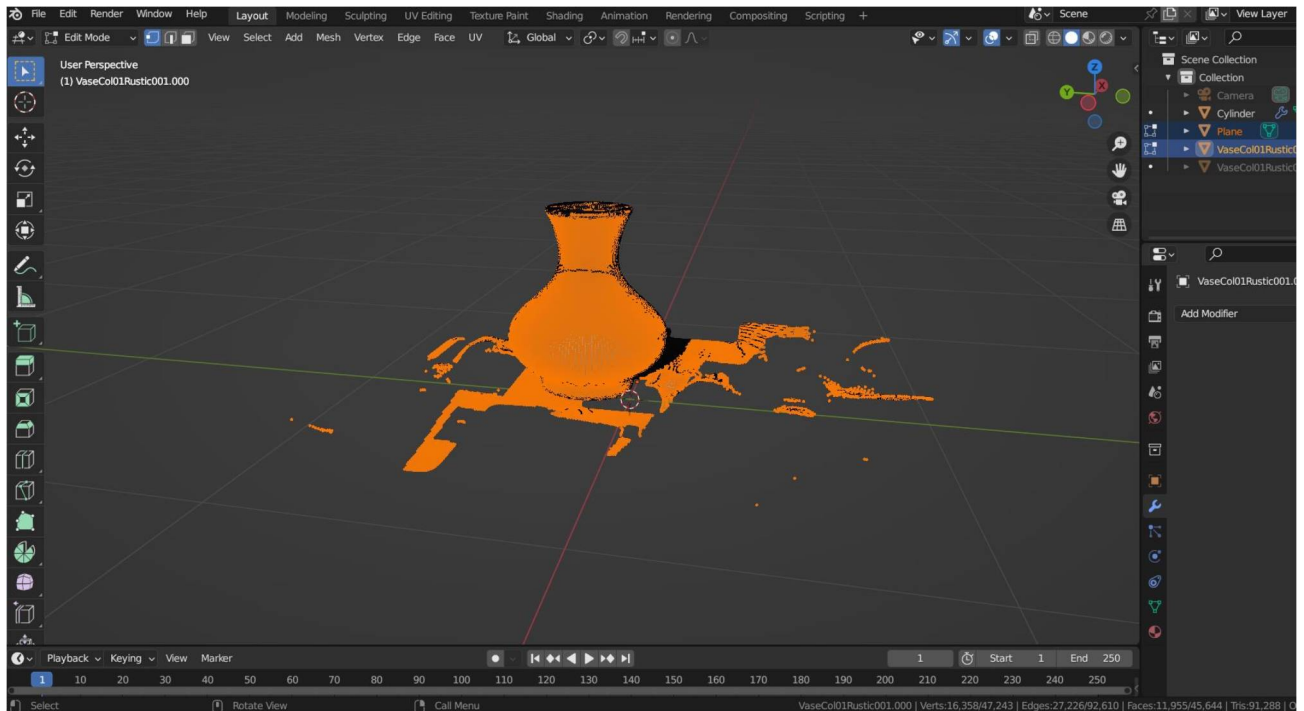


Рис 2.10 — Неоптимізована 3d модель

У Blender є різні інструменти та модифікатори, які можуть допомогти з цим кроком, наприклад, модифікатор Decimate, який зменшує кількість багатокутників у моделі, інструмент Fill Holes, який заповнює прогалини в сітці, і режим Sculpt, який був описан в 1 Розділі.

Четвертий крок полягає в повторній топологізації відсканованої моделі, що означає створення нової сітки з кращою топологією та меншою кількістю багатокутників, зберігаючи при цьому форму та деталі вихідної моделі Рис. 2.11. Цей крок важливий для підвищення продуктивності та якості моделі, особливо якщо вона призначена для анімації або візуалізації. У Blender є кілька інструментів і доповнень, які можуть допомогти з цим кроком, наприклад, модифікатор Shrinkwrap, який проектує нову сітку на відскановану модель, параметр Snap to Faces, який вирівнює вершини нової сітки до поверхні відскановану модель і надбудову RetopoFlow, яка надає спеціальний інтерфейс і інструменти для ретопології.

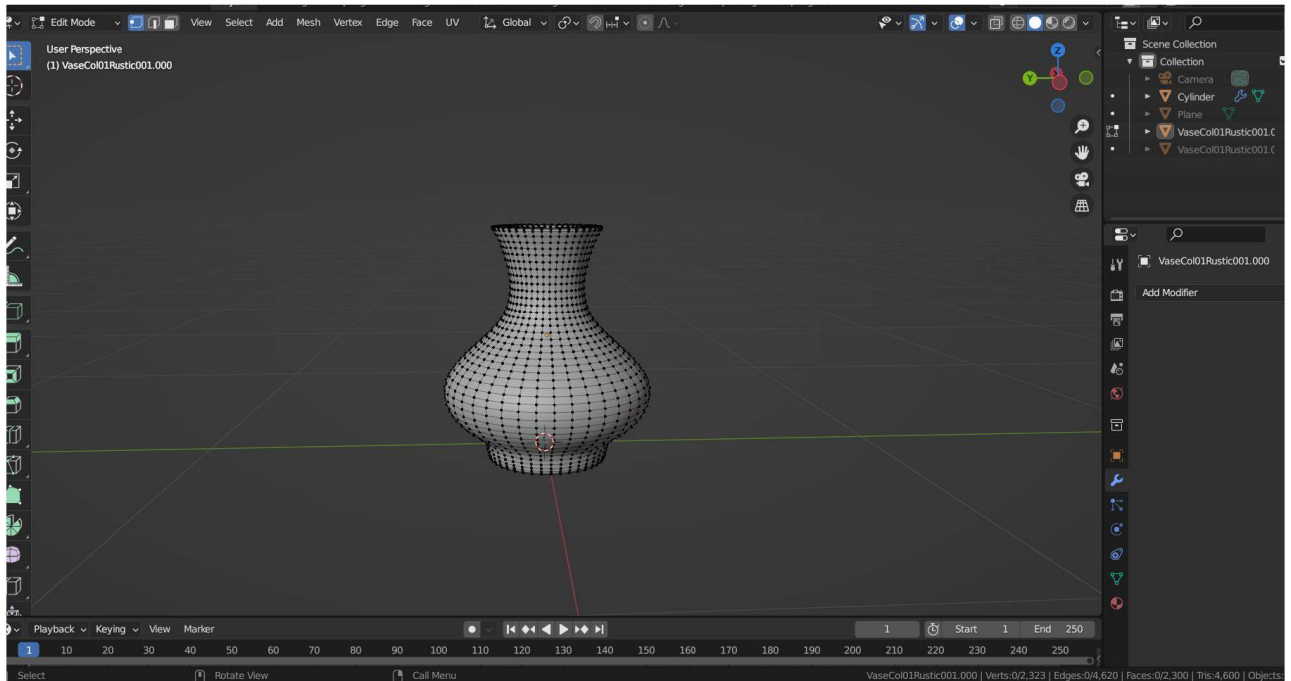


Рис 2.11 — Оптимізована 3d модель

UV-розгортка - цей крок передбачає створення UV карти, яка є способом відображення координат 2D текстури на 3D поверхню. Можна використовувати опцію Smart UV Project у Blender, щоб автоматично розгортати скан, або вибрати режим UV Editing, щоб вручну налаштувати шви.

Останнім кроком є текстурування сканованої моделі Рис 2.12. Перед налаштуванням текстур необхідно відредагувати текстуру, усуваючи дефекти, покращуючи деталі та оптимізуючи розмір текстури. Великі текстури можуть спричинити багато обчислювальних завдань та займати багато пам'яті.

Щоб додати текстури і матеріали до моделі, використовується вкладка "Material Properties" та "Texture Properties" у вікні Properties.

Після всіх етапів моделі готові до використання.

Оптимізація та деталізація відсканованих моделей у Blender — це комплексний процес, який вимагає як технічних навичок використання інструментів Blender, так і художньої оцінки. Мета полягає в тому, щоб зберегти баланс між збереженням точності оригінального сканування та забезпеченням ефективності моделі та її використання за призначенням.

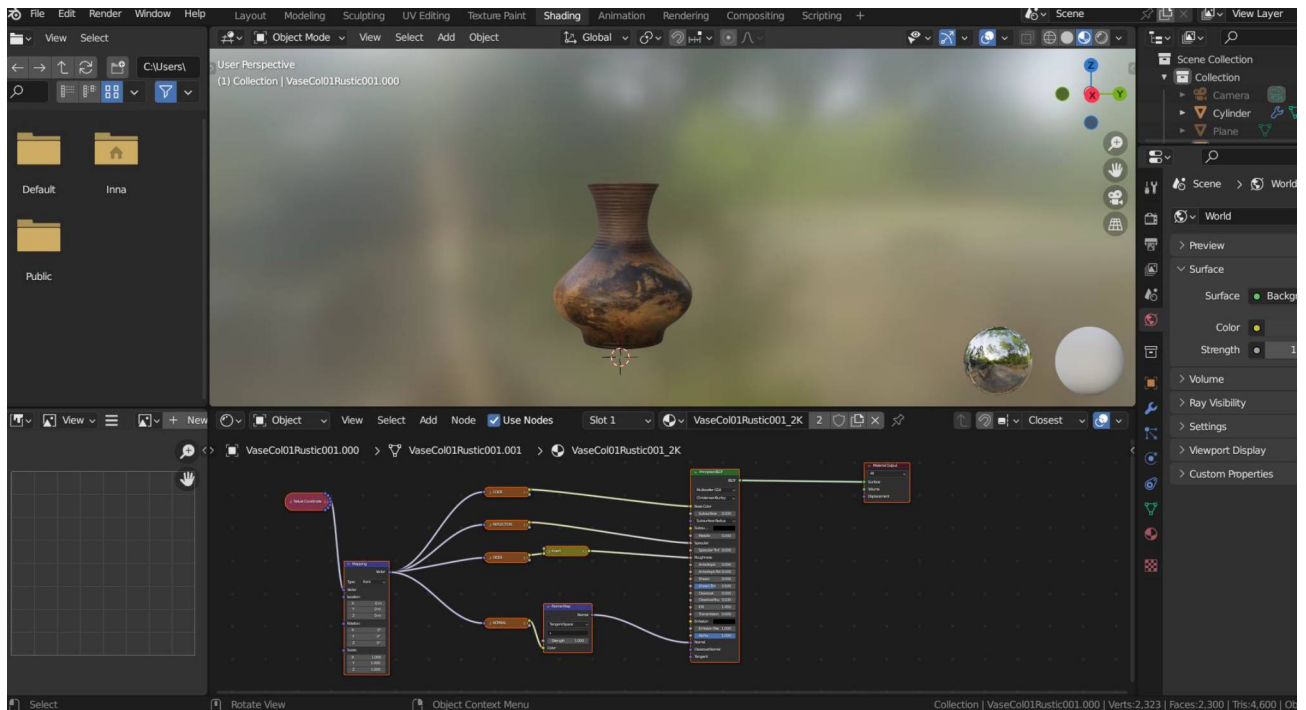


Рис 2.12 — Налаштування текстури 3d моделі

2.2.3 Імпортування та налаштування моделей у Unreal Engine

Готові 3D моделі в Blender експортуються в Unreal Engine в підтримуваних форматах, зазвичай це FBX або OBJ. Це робиться з метою передачі геометричної та текстурної інформації.

В Unreal Engine створюється новий проект. За допомогою вбудованого імпорту Unreal Engine, 3D моделі імпортуються у віртуальне середовище. У вікні імпорту налаштовуються параметри, такі як масштаб, система координат, текстури та матеріали. Імпортовані моделі розташовуються у віртуальній сцені Unreal Engine та налаштовуються для відповідного відображення. Це включає в себе розташування, обертання та масштабування моделей. Наступним кроком, необхідно налаштувати матеріали та текстури для моделей щоб вони відображалися правильно та реалістично. Для цього використовується редактор матеріалів Unreal Engine. Після цього налаштовується освітлення, додаються Point Lights (Точкові Світильники), розташовують їх у відповідних місцях та налаштовують параметри. Для реалістичного освітлення, налаштовуються тіні, щоб вони створювали м'які тіні.

Після завершення налаштувань об'єктів, отримуємо готову модель віртуального музею Рис 2.13 , залишається лише додати функціонал.

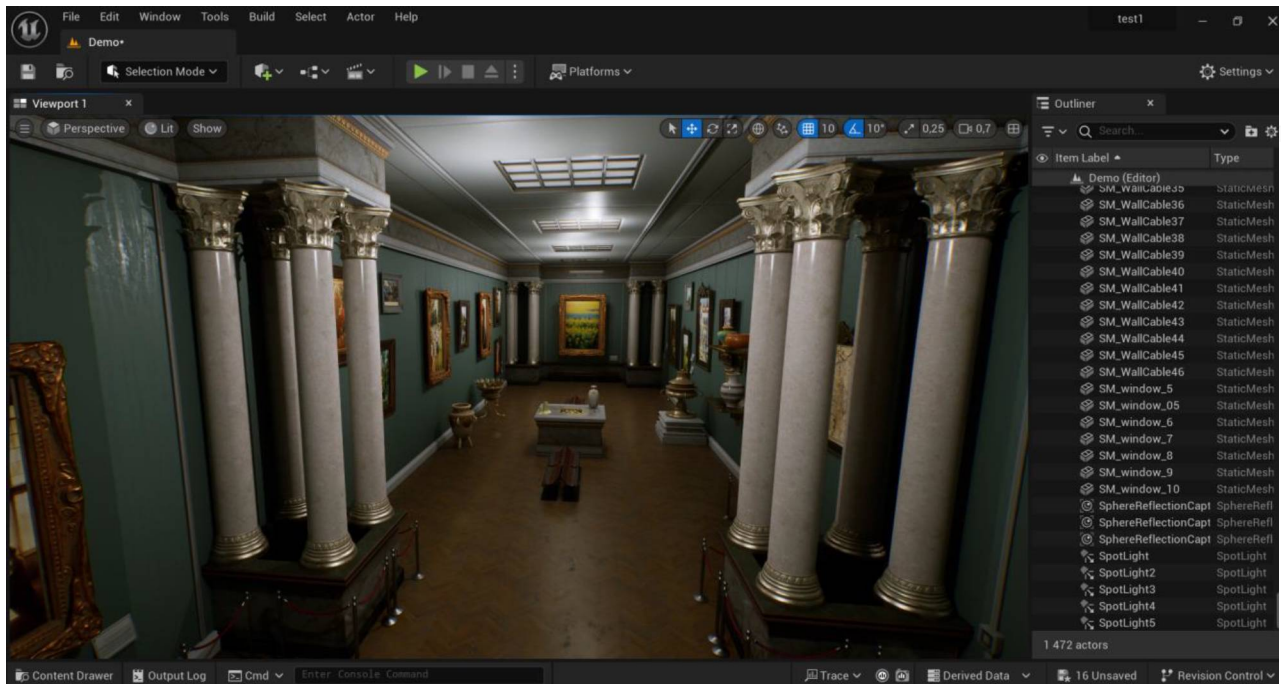


Рис 2.13 — Готова модель віртуального музею в Unreal Engine 5

2.2 Розробка функціоналу віртуального музею за допомогою с++

2.2.1 Визначення функціональних можливостей користувача

Для того щоб визначити функціональні можливості системи з точки зору користувача, які дії система має підтримувати, створимо Use Case діаграму.

Створення Use Case діаграми для віртуального музею допоможе візуалізувати різні взаємодії, які користувачі можуть мати з системою. Use Case діаграма спрощує процес розробки, подаючи стислий огляд того, що система повинна робити, без вдавання в технічні деталі реалізації. Вона вказує на сценарії, які потрібно протестувати, щоб переконатися, що всі вимоги користувача виконуються.



Рис 2.3 — Діаграма варіантів використання (UseCase diagram) “Комп’ютерної моделі віртуального музею з елементами віртуального середовища”.

На представленій діаграмі Use Case Рис 2.3 для віртуального музею, ми маємо актора "Користувач" та взаємодії, які він може здійснити з системою:

- **Обрати пристрій** - можливість відвідувача вибрати між VR-обладнанням та комп’ютером для навігації музеєм.
- **Налаштувати Мову** - налаштування бажаної мови інтерфейсу віртуального музею.
- **Навігація по Музею** - ця функція надає користувачу можливість переміщатися по різних по різних залах та експонатах віртуального музею.

- Масштабувати експонати - збільшення або зменшення зображення експонатів для детальнішого огляду.
- Обертати експонати 360 - відвідувач має змогу обертати експонати на 360 градусів.
- Перегляд текстової інформації про експонати - доступ до інформативних текстів про експонати, їх історію та значення.
- Прослуховувати аудіогіди - можливість прослуховувати аудіо описи експонатів.
- Переглядати меню - дає змогу користувачам переглядати навігаційне меню музею для вибору різних опцій та розділів.

2.3.1 Навігація та переміщення в середовищі VR

Для реалізації функціоналу віртуального музею в середовищі VR за допомогою C++ в Unreal Engine, потрібно створити VR персонажа, який дозволить користувачам переміщатися та взаємодіяти з середовищем Рис 2.14.

Перед написанням коду потрібно визначити вхідні сигнали в Unreal Engine. це можна зробити в редакторі проекту у розділі "Edit" -> "Project Settings" -> "Input". Тут встановлюємо осі для руху та обертання, які будуть використовуватися для VR контролерів, клавіатури та миші.

```

1 // VRCharacter.h
2 #include "CoreMinimal.h"
3 #include "GameFramework/Character.h"
4 #include "VRCharacter.generated.h"
5
6 UCLASS()
7 class VIRTUALMUSEUM_API AVRCharacter : public ACharacter
8 {
9     GENERATED_BODY()
10
11 public:
12     // конструктор для створення VR персонажа
13     AVRCharacter();
14
15 protected:
16     // Функція, яка ініціалізує VR персонажа під час створення
17     virtual void BeginPlay() override;
18
19 public:
20     // Функція оновлення кожного кадру
21     virtual void Tick(float DeltaTime) override;
22
23     // Функції для переміщення персонажа
24     void MoveForward(float Value);
25     void MoveRight(float Value);
26 };
27
28

```

Рис 2.14 — Клас користувача

Потім були реалізовані функції навігації користувача Рис 2.15. Ці функції дозволять користувачу переміщатися у віртуальному музеї як за допомогою VR контролерів, так і за допомогою клавіатури та миші. Для клавіатури та миші, використовуються стандартні сигнали руху та обертання.

Метод `SetupPlayerInputComponent` встановлює зв'язок між різними видами вводу та функціями, які будуть викликатися при зміні цих вхідних значень. У цьому випадку він встановлює обробники для руху вперед/назад (`MoveForward`), руху вліво/вправо (`MoveRight`), обертання персонажа за осі `Yaw` (`Turn`) та за осі `Pitch` (`LookUp`).

Функція `MoveForward` відповідає за рух персонажа вперед або назад. Вона отримує вхідне значення `Value`, яке представляє собою напрямок руху, і якщо це значення не дорівнює нулю, обчислює напрямок руху (`ForwardDirection`) на основі вектора вперед персонажа та додає рух у цьому напрямку.

```
// VRCharacter.cpp
#include "VRCharacter.h"
void AVRCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
{
    Super::SetupPlayerInputComponent(PlayerInputComponent);

    // Об'єднане управління для VR контролерів та клавіатури/миші
    PlayerInputComponent->BindAxis("MoveForward", this, &AVRCharacter::MoveForward);
    PlayerInputComponent->BindAxis("MoveRight", this, &AVRCharacter::MoveRight);

    // Обертання персонажа
    PlayerInputComponent->BindAxis("Turn", this, &APawn::AddControllerYawInput);
    PlayerInputComponent->BindAxis("LookUp", this, &APawn::AddControllerPitchInput);
}

void AVRCharacter::MoveForward(float Value)
{
    if (Value != 0.0f)
    {
        // Отримання напрямку руху вперед/назад
        FVector ForwardDirection = GetActorForwardVector();
        // Додаємо рух у напрямку цього вектора
        AddMovementInput(ForwardDirection, Value);
    }
}

void AVRCharacter::MoveRight(float Value)
{
    if (Value != 0.0f)
    {
        // Отримання напрямку руху вліво/вправо
        FVector RightDirection = GetActorRightVector();
        // Додаємо рух у напрямку цього вектора
        AddMovementInput(RightDirection, Value);
    }
}
```

Рис 2.15 — Функції навігації користувача

Значення Value автоматично адаптується в залежності від способу вводу: джойстик на VR контролері або клавіші W, A, S, D на клавіатурі. Аналогічно, обертання камери може контролюватися як за допомогою VR контролерів, так і за допомогою миші. Коли користувач рухає джойстик вперед, Value буде позитивним, що призведе до руху персонажа вперед. Коли джойстик рухається назад, Value стає негативним, що призведе до руху персонажа назад. Аналогічно, рух джойстика вліво чи вправо призведе до відповідного переміщення персонажа. Система вхідних даних Unreal Engine обробляє ці сигнали і передає відповідні значення в функції..

2.3.2 Інтерактивні елементи: вибір, перегляд та маніпуляція об'єктами

Для реалізації інтерактивних елементів у віртуальному музеї, таких як вибір об'єкта, його масштабування на весь екран, та можливість обертання на 360 градусів, були використані наступні підходи:

Спершу створили клас експоната Рис 2.16. Цей клас представляє об'єкти, з якими можна взаємодіяти у віртуальному музеї.

```
// ExhibitObject.h
#include "CoreMinimal.h"
#include "Gameframework/Actor.h"
#include "ExhibitObject.generated.h"

UCLASS()
class VIRTUALMUSEUM_API AExhibitObject : public AActor
{
    GENERATED_BODY()

public:
    AExhibitObject();

    // Функції для управління станом об'єкта
    void SelectObject();
    void DeselectObject();

private:
    // Зберігаємо початкові параметри об'єкта
    FVector OriginalScale;
    FVector OriginalLocation;
};
```

Рис 2.16 — Клас експоната музею

AExhibitObject є класом, який успадковується від AActor, який є базовим класом для об'єктів. Він також позначений макросом UCLASS() та використовує GENERATED_BODY() для генерації необхідного коду Unreal Engine.

OriginalScale та OriginalLocation зберігають початкові параметри масштабу та розташування об'єкта. Це корисно для повернення об'єкта в його початковий стан після взаємодії.

Далі прописали функції цього класу Рис 2.17.

```
// ExhibitObject.cpp
#include "ExhibitObject.h"

AExhibitObject::AExhibitObject()
{
    // Конструктор класу
    // Ініціалізація компонентів, якщо потрібно
}

void AExhibitObject::SelectObject()
{
    // Запам'ятовуємо початкові параметри об'єкта
    OriginalScale = GetActorScale3D();
    OriginalLocation = GetActorLocation();

    // Можна також змінити вигляд об'єкта, наприклад, підсвітити його
}

void AExhibitObject::DeselectObject()
{
    // Відновлюємо початкові параметри об'єкта
    SetActorScale3D(OriginalScale);
    SetActorLocation(OriginalLocation);

    // Вимикаємо підсвітку, якщо вона була включена
}

```

Рис 2.17 — Функції експоната музею

Функція SelectObject() зберігає початкові параметри об'єкта (масштаб та розташування) та, опціонально, може змінювати вигляд об'єкта (наприклад, підсвічувати його). Функція DeselectObject() відновлює початкові параметри об'єкта, включаючи масштаб та розташування, і може вимикати підсвітку, якщо вона була включена.

Механізм вибору об'єкта був реалізований за допомогою системи трасування променів (raycasting), яка дозволяє визначити, на який об'єкт вказує користувач Рис 2.18.

```

// Визначення об'єкта, на який вказує користувач
void AVRCharacter::SelectObject()
{
    FVector Start, Dir;
    FHitResult Hit;

    // Отримання позиції та напрямку вказівника
    GetPointerData(Start, Dir);

    // Виконання трасування променів
    if (<GetWorld()->LineTraceSingleByChannel(Hit, Start, Start + Dir * TraceDistance, ECC_Visibility))
    {
        // Перевірка, чи було щось виявлено
        if (AExhibitObject* SelectedObject = Cast<AExhibitObject>(Hit.Actor))
        {
            // Збереження вибраного об'єкта для подальшої обробки
            CurrentSelectedObject = SelectedObject;
        }
    }
}

// Функція для отримання позиції та напрямку вказівника
void AVRCharacter::GetPointerData(FVector& OutStart, FVector& OutDir)
{
    // Отримання позиції та орієнтації VR контролера або курсора миші
}

```

Рис 2.18 — Функції визначення об'єкта, на який вказує користувач

Після вибору об'єкта, потрібно здійснити його масштабування та відображення на весь екран. Для цього була прописана функція масштабування об'єкта 2.19.

```

void AVRCharacter::ScaleSelectedObject()
{
    if (CurrentSelectedObject)
    {
        // Масштабування об'єкта для його відображення на весь екран
        CurrentSelectedObject->SetActorScale3D(FVector(ZoomScale));

        // Визначаємо позицію перед користувачем на певній відстані
        FVector NewLocation = GetActorLocation() + GetActorForwardVector() * DistanceInFrontOfPlayer;

        // Переміщення об'єкта до відповідного місця для перегляду
        CurrentSelectedObject->SetActorLocation(NewLocation);
    }
}

```

Рис 2.19 — Функція масштабування об'єкта

Цей метод забезпечує те, що коли об'єкт масштабується, він з'являється перед користувачем, що зручно для спостереження та взаємодії.

Об'єкт можна обертати на 360 градусів, для цього була прописана функція Рис 2.20. Для обертання об'єкта на 360 градусів, можна використовувати введені дані, такі як рух миші або VR контролера.

```

void AVRCharacter::RotateSelectedObject(float Horizontal, float Vertical)
{
    if (CurrentSelectedObject)
    {
        // Обертання об'єкта
        CurrentSelectedObject->AddActorLocalRotation(FRotator(Vertical, Horizontal, 0.0f));
    }
}

```

Рис 2.20 — Функція обертання об'єкта

Для того щоб вийти з перегляду експоната і повернутись до нормального перегляду музею, була реалізована функція CloseObjectView Рис 2.21.

```

void AVRCharacter::CloseObjectView()
{
    if (CurrentSelectedObject)
    {
        // Повернення об'єкта до його початкового розміру та позиції
        CurrentSelectedObject->SetActorScale3D(OriginalScale);
        CurrentSelectedObject->SetActorLocation(OriginalLocation);

        // Очищення вибраного об'єкта
        CurrentSelectedObject = nullptr;
    }
}

```

Рис 2.21 — Функція закриття Масштабування

2.3.3 Реалізація аудіо гіда та текстової інформації

Для реалізації аудіо гіда та текстової інформації, які доступні при виборі та масштабуванні об'єкта, додаємо інтерактивні елементи (іконки) та функціонал для їх обробки Рис 2.22. На етапі дизайну були створенні іконки для аудіо гіда та текстової інформації. Вони знаходяться поруч з об'єктом після його масштабування.

UMuseumInterface успадковується від UUserWidget. Він містить функції та змінні, які використовуються для управління аудіогідом та текстовою інформацією.

ToggleAudioGuide() та ToggleTextInfo(): Функції для перемикання аудіогіда та текстової інформації. Вони визначені як BlueprintCallable, щоб їх можна було викликати з Blueprint-скриптів.

```

// MuseumInterface.h
#pragma once

#include "CoreMinimal.h"
#include "Blueprint/UserWidget.h"
#include "MuseumInterface.generated.h"

UCLASS()
class VIRTUALMUSEUM_API UMuseumInterface : public UUserWidget
{
    GENERATED_BODY()

public:
    // Функції для перемикання аудіогіда та текстової інформації
    UFUNCTION(BlueprintCallable, Category = "UI")
    void ToggleAudioGuide();

    UFUNCTION(BlueprintCallable, Category = "UI")
    void ToggleTextInfo();

private:
    // Елементи інтерфейсу: іконки, текстовий блок, аудіокомпонент
    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UTexture2D* AudioGuideIcon;

    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UTexture2D* TextInfoIcon;

    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UTextBlock* TextInfoWidget;

    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UAudioComponent* AudioGuideAudioComponent;

    // Змінні для відстеження стану аудіогіда та текстової інформації
    bool bAudioGuideActive;
    bool bTextInfoActive;
};

```

Рис 2.22 — Інтерфейс аудіогіда та текстової інформації

Реалізація Функціоналу Рис 2.23. `ToggleAudioGuide()` визначає функціональність для включення та вимикання аудіогіда. `bAudioGuideActive` визначає поточний стан аудіогіда (увімкнений чи вимкнений). При увімкненні аудіогіда, метод викликає `Play()` для початку відтворення звукового компонента, а також змінює іконку інтерфейсу. При вимиканні аудіогіда, метод викликає `Stop()` для припинення відтворення звукового компонента, а також змінює іконку інтерфейсу (можливо, на попередню іконку). `ToggleTextInfo()` визначає функціональність для включення та вимикання текстової інформації. `bTextInfoActive` визначає поточний стан текстової інформації (відображена чи прихована). При включенні текстової інформації, метод встановлює текст та зробиє його видимим на інтерфейсі. При вимиканні текстової інформації, метод приховує текст на інтерфейсі.

```

// MuseumInterface.cpp
#include "MuseumInterface.h"
#include "Components/Image.h"
#include "Components/TextBlock.h"
#include "Components/AudioComponent.h"

void UMuseumInterface::ToggleAudioGuide()
{
    // Перемикаємо стан аудіогіда
    bAudioGuideActive = !bAudioGuideActive;

    if (bAudioGuideActive)
    {
        // Запускаємо аудіогід та змінюємо іконку
        AudioGuideAudioComponent->Play();
        Cast<UImage>(GetWidgetFromName(TEXT("AudioGuideIcon")))->SetBrushFromTexture(AudioGuideIcon);
    }
    else
    {
        // Вимикаємо аудіогід та змінюємо іконку
        AudioGuideAudioComponent->Stop();
        Cast<UImage>(GetWidgetFromName(TEXT("AudioGuideIcon")))->SetBrushFromTexture(/*previous icon*/);
    }
}

void UMuseumInterface::ToggleTextInfo()
{
    // Перемикаємо стан текстової інформації
    bTextInfoActive = !bTextInfoActive;

    if (bTextInfoActive)
    {
        // Відображаємо текстову інформацію та робимо видимим
        TextInfoWidget->SetText(/*your text data*/);
        TextInfoWidget->SetVisibility(ESlateVisibility::Visible);
    }
    else
    {
        // Приховуємо текстову інформацію
        TextInfoWidget->SetVisibility(ESlateVisibility::Hidden);
    }
}

```

Рис 2.23 — Інтерфейс аудіогіда та текстової інформації

Висновки за розділом 2

У рамках роботи було визначено структуру музею та проведено підбір і аналіз необхідних експонатів, що враховувалися при створенні концепції віртуального музею. Програмне забезпечення Meshroom використовувалося для створення 3D-моделей із зібраних сканів, що є свідченням ефективності фотограмметрії як методу цифрової реконструкції об'єктів. Окреслено основні функціональні можливості майбутнього віртуального музею, які включають здатність користувача вибирати пристрої для перегляду, налаштування мови інтерфейсу, навігацію, масштабування та обертання експонатів, а також перегляд текстової інформації та прослуховування аудіогідів.

РОЗДІЛ 3 РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ

3.1 Рекомендації для Користувача з віртуальним шоломом

1. Ознайомлення з Інтерфейсом

Перед тим, як почати дослідження віртуального музею, варто ознайомитись з базовими елементами управління.

При вході в музей, користувач потрапляє на головний екран Рис. 3.1, де йому буде запропоновано вибрати між двома режимами:

- VR-режим (для користувачів з VR-обладнанням)
- Стандартний режим (для користування через комп'ютер з клавіатурою та мишею)

Потрібно обрати "VR-режим", щоб ввімкнути режим з керування клавіатурою та мишею.

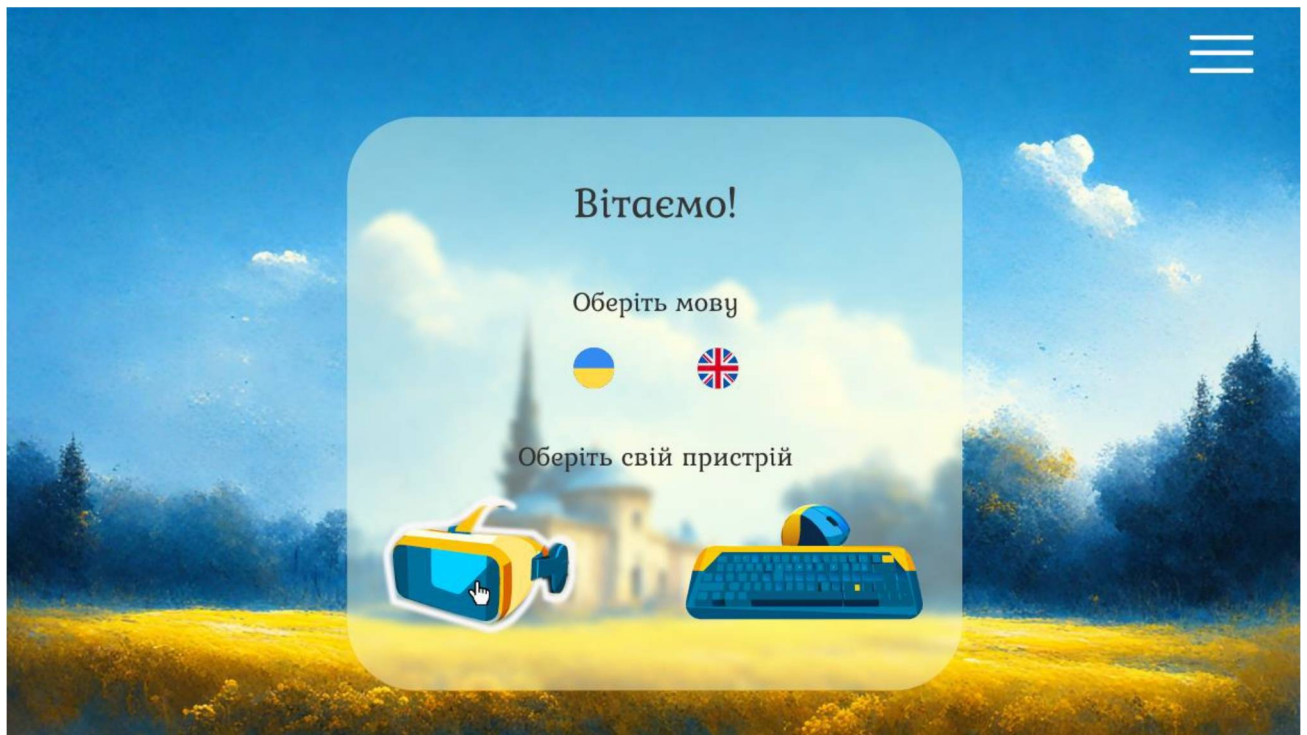


Рис 3.1 — Головний екран віртуального музею

Щоб змінити мову необхідно на головному екрані натиснути на іконку з прапором Рис 3.2, або через меню Рис 3.3.

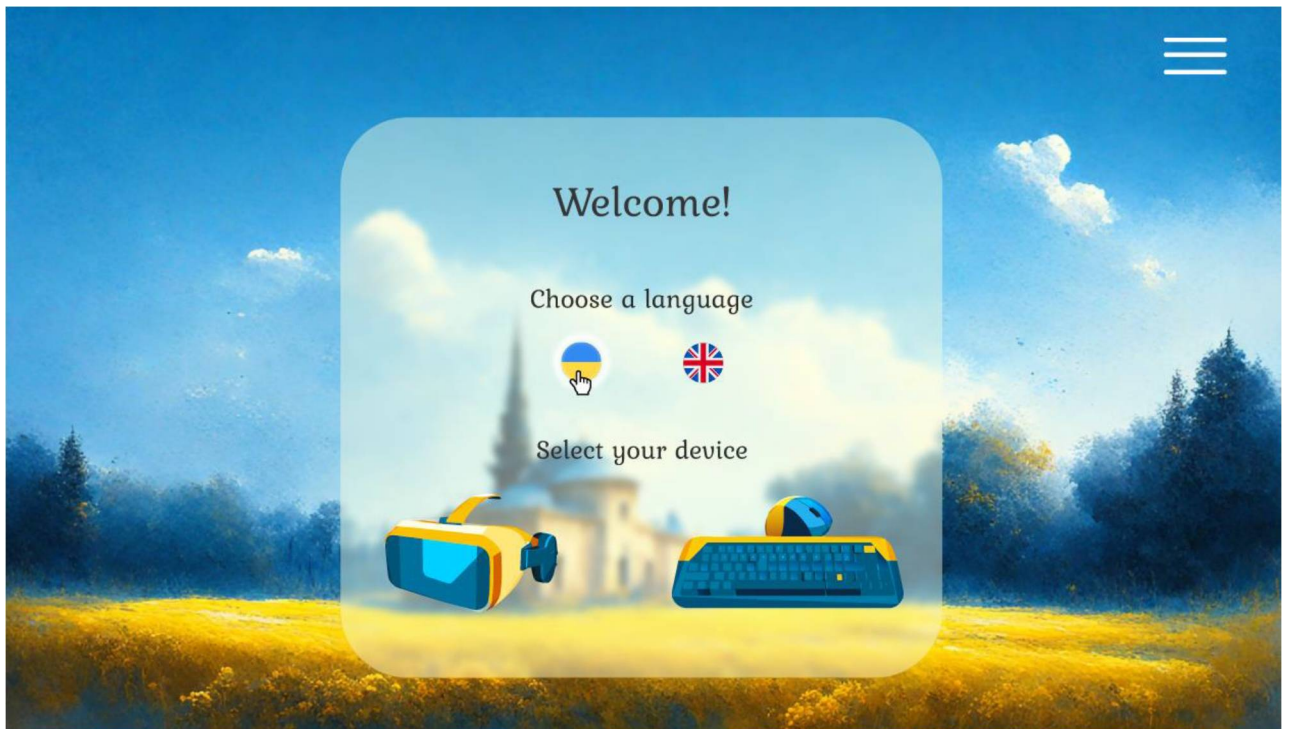


Рис 3.2 — Зміна мови

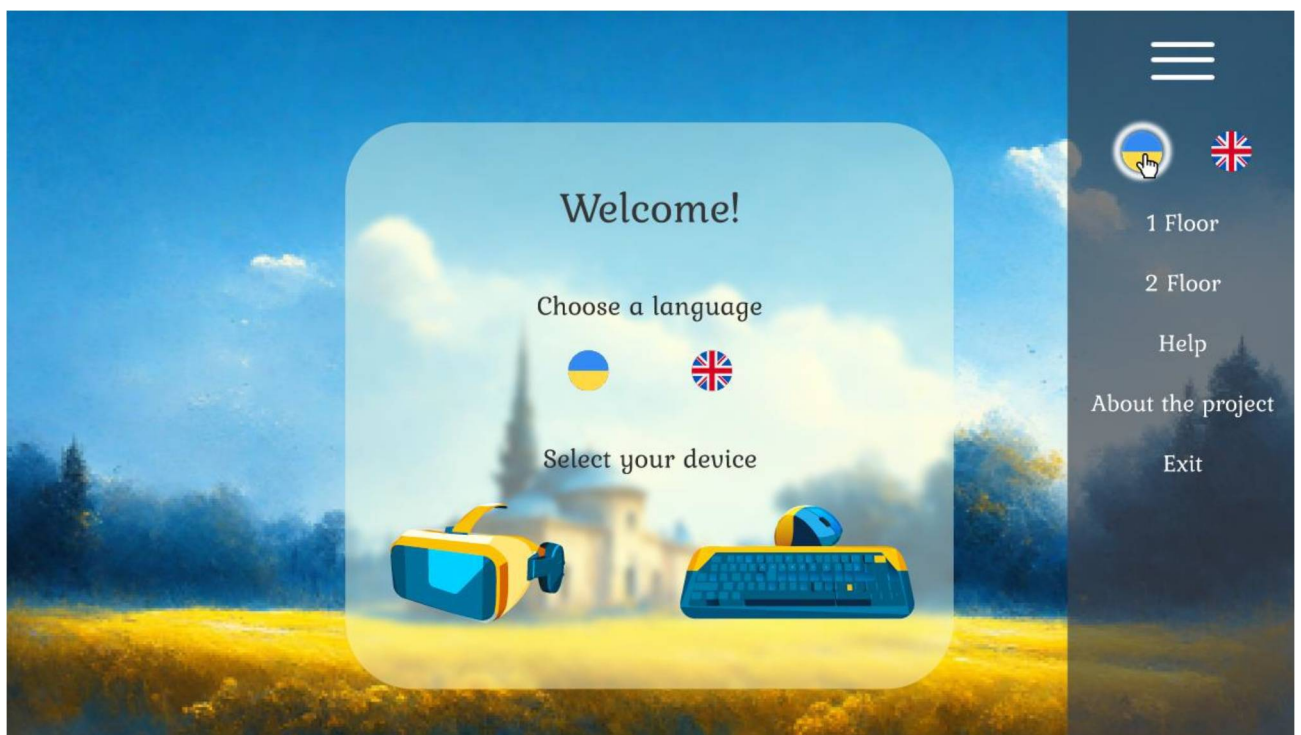


Рис 3.3 — Меню

Меню:

У разі виникненні запитань потрібно скористатись розділом “Допомога”. Щоб дізнатись більше про сам проект, необхідно натиснути “Про проект”. Для

швидкої навігації можна використовувати пункти меню “Головний екран”, “1 Поверх”, “2 Поверх” Рис 3.3.

Для переміщення вперед, назад та вбік потрібно використовувати джойстик або сенсорну панель на одному з контролерів. Перед тим, як переміститися, необхідно переконатися, що у фізичному просторі достатньо місця і воно безпечне для пересування. Напрямок переміщення залежить від положення джойстика: вперед, назад, вліво, вправо. Швидкість переміщення може залежати від сили натискання на джойстик. Для повороту камери необхідно використати джойстик на іншому контролері, натискати джойстик вліво або вправо.

Натискайте джойстик вліво або вправо для обертання персонажа в музеї.

Взаємодія з Експонатами

Для того щоб вибрати експонат, необхідно вибрати його за допомогою контролера та натиснути та натиснути тригер (зазвичай передня кнопка на контролері) для вибору. Після чого вибраний об'єкт з'явиться на весь екран з можливістю детального перегляду, його можна обертати на 360 градусів Рис.3.3-3.4, для цього необхідно використовувати джойстик або сенсорну панель для обертання експонату [31].



Рис 3.4 — Вибір експоната



Рис 3.5 — Взаємодія з експонатами

Після вибору експоната з'явиться іконка, схожа на книгу Рис 3.6. Для того щоб переглянути текстову інформацію про обраний експонат необхідно натиснути на цю іконку, щоб відкрити детальну текстову інформацію про обраний об'єкт. Щоб закрити текстову інформацію потрібно натиснути хрестик.

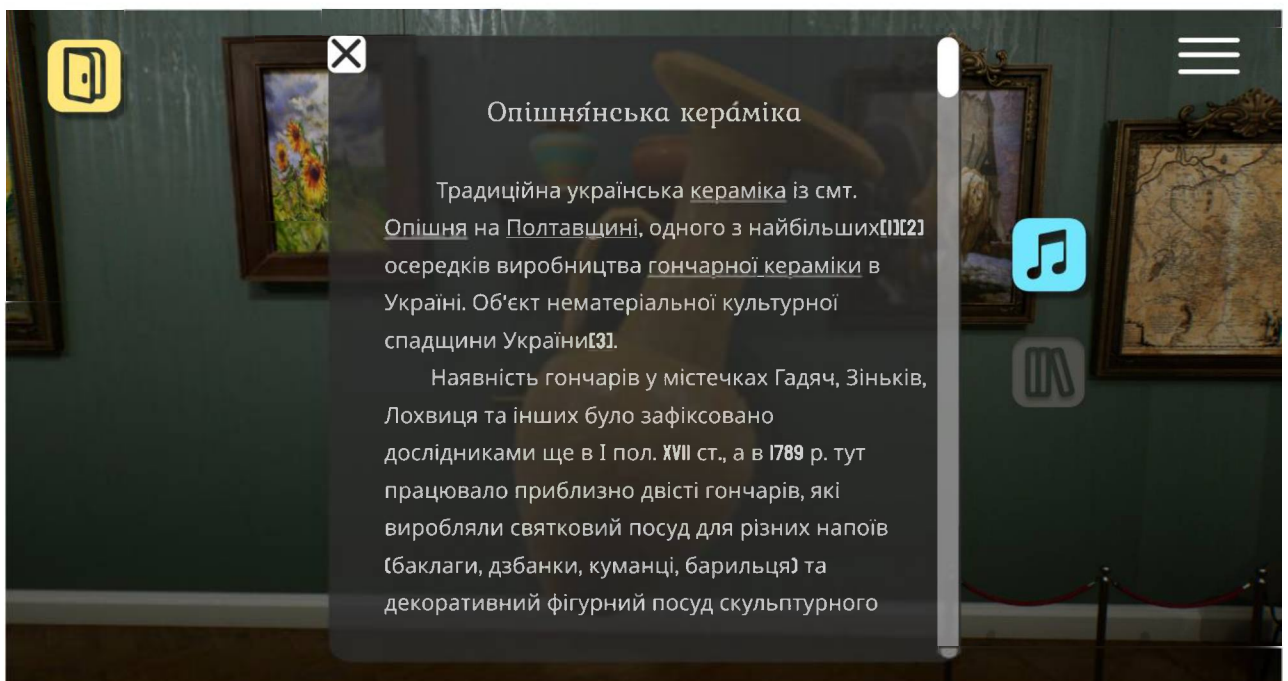


Рис 3.6 — Текстовий гід

Ця інформація може включати історію, призначення, характеристику та інші цікаві факти. Окрім текстової інформації є можливість активувати аудіогід

для обраного експоната. Це можна зробити, натиснувши на відповідну іконку аудіогіда, яка з'являється разом з текстовою інформацією, щоб вимкнути аудіо, потрібно повторно натиснути іконку музики. Аудіогід надасть вербальний опис експонату, що допоможе глибше зрозуміти його історію та значення.



Рис 3.7 — Ввімкнений та вимкнений аудіогід

3.2 Рекомендації для Користувача без Віртуального шолома

Для користувачів з клавіатурою та мишею принцип користування майже такий як і для користувачів із VR-шоломом, відмінність лише тому що замість контролера використовується миша та клавіатура.

Щоб обрати керування клавіатурою на головному екрані потрібно обрати клавіатуру та мишку, як на Рис. 3.8

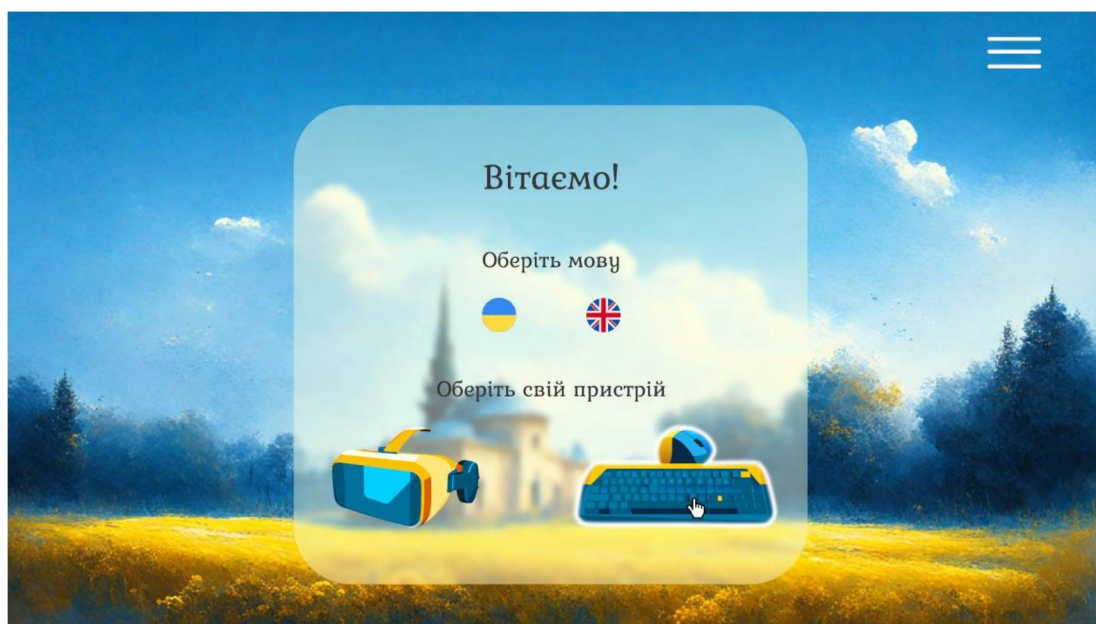


Рис 3.8 — Головний екран, клавіатура як керуючий пристрій

Для переміщення по простору музею використовуються клавіші W, A, S, D для переміщення вперед, вліво, назад та вправо відповідно.

Для обертання камери, зміни напрямку погляду використовується миша. Також можна використовувати клавіші зі стрілками для повороту. Рис.3.9



Рис 3.8 — Навігація та меню

Щоб взаємодіяти з експонатами, необхідно навести курсор на об'єкт та натиснути ліву кнопку миші. Інші функції розписані вище в підпункті 3.1

2.3 Тестування віртуального музею

Тестування віртуального музею проводилось з використанням різних методів, щоб забезпечити його надійність та ефективність. Ось ключові етапи:

1. Інтерактивне Юзер-Тестування: Цей етап включає в себе залучення різних користувачів для оцінки інтерфейсу та взаємодії. Користувачам було запропоновано відвідати музей та взаємодіяти з експонатами. Вони могли вільно переміщатися по музею, вивчати експонати та користуватися всіма доступними функціями. Після відвідування музею, користувачам було запропоновано заповнити анкету, де вони могли висловити свої враження, вказати на проблеми, з якими вони зіткнулися, та дати загальну оцінку свого досвіду. Отримані відгуки

аналізувалися для виявлення слабких місць, виявлення паттернів у використанні, проблем з навігацією та доступністю інформації.

Результат тестування: Користувачі оцінили інтуїтивність інтерфейсу та зручність навігації.

Вони високо оцінили якість візуальних елементів та взаємодію з експонатами. Загальний досвід був позитивним, що свідчить про успішність дизайну, інтерфейсу та реалізації музею.

2. Крос-Платформне Тестування: Музей було протестовано на різних пристроях, що допомогло забезпечити його широку сумісність та доступність. Ось деталі цього процесу:

VR-шоломи: Музей було протестовано на різних моделях VR-шоломів. Це включало в себе перевірку якості графіки, зручності керування та загального досвіду користувача. Було виявлено, що музей працює добре на всіх протестованих VR-шоломах, забезпечуючи високоякісний віртуальний досвід.

Стандартні комп'ютери: Музей також було протестовано на стандартних комп'ютерах з використанням клавіатури та миші для керування. Було перевірено, що навігація та взаємодія з експонатами працюють гладко. Користувачі мали змогу легко переміщатися по музею та вивчати експонати.

Було виявлено що для

Всі ці тести допомогли забезпечити, що віртуальний музей працює ефективно та надає користувачам високоякісний досвід. Завдяки цьому музей може служити важливим інструментом для освіти та культурного обміну. Завдяки успішному проходженню всіх тестів, можна стверджувати, що віртуальний музей готовий до використання. коректної роботи віртуального музею, обладнання користувача повинно мати достатню потужність для рендерингу в реальному часі. Це означає, що графічний процесор (GPU) та центральний процесор (CPU) повинні бути достатньо потужними для обробки великої кількості даних, необхідних для створення віртуального середовища.

Крім того, об'єм оперативної пам'яті (RAM) також впливає на здатність системи ефективно обробляти віртуальні сцени.

Завдяки успішному проходженню всіх тестів, можна стверджувати, що віртуальний музей готовий до використання.

Висновки за розділом 3

У третьому розділі кваліфікаційної роботи описані рекомендації щодо використання віртуального музею, які забезпечують зручність та легкість в управлінні для користувачів. Основні аспекти включають інтуїтивно зрозумілий інтерфейс для користувачів з VR-шоломами та тих, хто використовує клавіатуру та мишу. Універсальність навігації та взаємодії з експонатами дозволяє користувачам легко переміщатися по музею та взаємодіяти з експонатами, використовувати аудіо гідів та текстові описи.

ВИСНОВКИ

У цій квалі роботі було розроблено віртуальний музей, який використовує сучасні технології Blender, Unreal Engine 5 та C++ для створення інтерактивного віртуального середовища, що зберігає та популяризує історичні та культурні об'єкти України. Метою проекту було створити віртуальний музей, який не тільки демонструє експонати в детальній 3D-графіці, але й надає користувачам можливість взаємодіяти з ними, слухати аудіогіди та читати текстову інформацію. Також проект сприяє розвитку віртуального туризму, дозволяючи людям з усього світу відвідувати віртуальний музей без фізичного присутності.

Для досягнення цієї мети було проведено наступні етапи дослідження та розробки:

Аналіз вимог та обмежень для віртуального музею, включаючи вибір тематики, структури, експонатів, функціональності та обладнання для VR.

Вибір програмного забезпечення для розробки віртуального музею, зокрема Blender та Unreal Engine 5, які надають потужні інструменти для моделювання, візуалізації та створення інтерактивних VR-середовищ.

Використання фотограмметрії як найефективнішого методу для сканування експонатів, оскільки вона дозволяє точно відтворити деталі та текстури об'єктів. Для перетворення зображень у 3D-об'єкти була використана програма Meshroom, яка ефективно використовує фотограмметричні методи для створення деталізованих 3D-моделей.

Розробка моделі віртуального музею в Blender, включаючи створення архітектури, освітлення, текстур, матеріалів та анімації. Для імпорту 3D-моделей експонатів було використано плагін BlenderGIS, який дозволяє завантажувати географічні дані з різних джерел.

Розробка інтерактивності та функціональності віртуального музею в Unreal Engine 5, включаючи створення сценаріїв, інтерфейсів, звуків, навігації, масштабування, обертання, аудіогідів та текстової інформації. Для

програмування було використано мову C++, яка є основною мовою для розробки в Unreal Engine 5.

Тестування та оцінка віртуального музею, включаючи перевірку якості 3D-графіки, інтерактивності, функціональності, зручності та легкості в управлінні. Для тестування було використано різні пристрої, такі як VR-шоломи, комп'ютери, смартфони та планшети.

Результати дослідження та розробки показали, що віртуальний музей є успішним проектом, який відповідає поставленим вимогам та обмеженням. Віртуальний музей демонструє високу якість 3D-графіки, інтерактивності та функціональності, а також забезпечує зручність та легкість в управлінні для користувачів. Віртуальний музей є важливим освітнім ресурсом, який дозволяє людям з усього світу досліджувати українську історію та культуру без фізичного відвідування музею. Віртуальний музей також сприяє розвитку віртуального туризму, забезпечуючи можливість відвідування та вивчення об'єктів з усього світу, сприяючи освіті, туризму, збереженню національної ідентичності та історичної пам'яті.

Таким чином, ця кваліфікаційна робота є внеском у сферу розробки віртуальних музеїв, які використовують сучасні технології Blender, Unreal Engine 5 та C++ для створення інтерактивного віртуального середовища, що зберігає та популяризує історичні та культурні об'єкти України. Проект має великий потенціал для подальшого розвитку та вдосконалення, а також може слугувати прикладом для інших віртуальних музеїв, які хочуть використовувати сучасні технології для збереження та популяризації культурної спадщини.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розгон О. В. Віртуальна версія музею як засіб впровадження цифрових технологій. Право та інноваційне суспільство: електронне наукове видання, 2019. с. 10-15. URL: <http://apir.org.ua/wp-content/uploads/2019/12/Rozghon13.pdf> (дата звернення: 13.12.2022 р.).
2. Віртуальний тур українськими музеями просто неба: веб-сайт. URL: <https://museums.authenticukraine.com.ua/ua/> (дата звернення: 13.12.2022 р.).
3. Топ-10 найкращих віртуальних екскурсій в Україні: веб-сайт. URL: <https://covid.unian.ua/10922969-top-10-naykrashchih-virtualnih-ekskursiy-v-ukrajini.html> (дата звернення: 15.12.2022 р.).
4. Pop I., Borza A. Technological innovations in museums as a source of competitive advantage. In Proceedings of the 2nd International Scientific Conference SAMRO, 2016; Volume 1, pp. 398–405 (дата звернення: 20.12.2022 р.).
5. Волинець В. О. Новий зміст і потенціал віртуального музею. Цифрова платформа: інформаційні технології в соціокультурній сфері, 2020. с. 143-154. URL: https://www.researchgate.net/publication/347980705_New_Content_and_Potential_of_the_Virtual_Museum (дата звернення: 23.12.2022 р.).
6. Maria Shehade. Virtual Reality in Museums: Exploring the Experiences of Museum Professionals. Virtual Reality and Its Application in Cultural Heritage, 2020. с.37-43 URL: <https://www.mdpi.com/2076-3417/10/11/4031> (дата звернення: 25.12.2022 р.).
7. Guazzaroni G. Role of Emotions in Interactive Museums: How Art and Virtual Reality Affect Emotions. In Virtual and Augmented Reality in Education, Art, and Museums, 2020. с. 174–193 (дата звернення: 04.01.2023 р.).
8. Dewdney A. What Is the Current Fascination with VR on the Part of Museums and Art Galleries? 2020. с. 56-64 URL: <https://openresearch.lsbu.ac.uk/item/86quq> (дата звернення: 08.01.2023 р.).

9. Frédéric Mérienne. Virtual Reality: Principles and Applications. Encyclopedia of Computer Science and Technology, 2018. с. 1-11. URL: <https://hal.science/hal-01728062/document> (дата звернення: 13.01.2023 р.).
10. Unity: офіційний сайт. URL: <https://unity.com/> (дата звернення: 16.01.2023 р.).
11. 10+ Best Software for AR and VR Design in 2023. URL: <https://justcreative.com/best-software-for-ar-and-vr-design/> (дата звернення: 13.01.2023 р.).
12. Pinar Oruç. Rethinking Who ‘Keeps’ Heritage: 3D Technology, Repatriation and Copyright. 2022. с. 346-348 URL: <https://academic.oup.com/grurint/article/71/12/1138/6692637> (дата звернення: 19.01.2023 р.).
13. Christopher Dostal. Photogrammetric texture mapping: A method for increasing the Fidelity of 3D models of cultural heritage materials. Journal of Archaeological Science: Reports, Volume 18, 2018. с. 430-436. URL: <https://www.sciencedirect.com/science/article/abs/pii/S2352409X17306892> (дата звернення: 21.01.2023 р.).
14. Anna Fryskowska. A no-reference method of geometric content quality analysis of 3D models generated from laser scanning point clouds for hBIM. Journal of Cultural Heritage, Volume 34, 2018. с. 95-108. URL: <https://www.sciencedirect.com/science/article/abs/pii/S1296207417305897> (дата звернення: 23.01.2023 р.).
15. The Best 3D Scanning Software of 2023. URL: <https://all3dp.com/2/best-3d-scanning-software-tools/> (дата звернення: 24.01.2023 р.).
16. Federica Pellitteri. Comparative analysis of intraoral scanners accuracy using 3D software: an in vivo study. 2022. URL: <https://progressinorthodontics.springeropen.com/articles/10.1186/s40510-022-00416-5> (дата звернення: 26.01.2023 р.).

17. Use Meshroom To Create 3D Models from Photographs. URL: <https://www.gamedesigning.org/learn/meshroom-tutorial/?ssp=1&setlang=uk-UA&safesearch=moderate> (дата звернення: 30.01.2023 р.).
18. Villar. Learning Blender. 2021. с. 23-45 (дата звернення: 4.02.2023 р.).
19. JM Blain. The complete guide to Blender graphics: computer modeling & animation. 2019. (Дата звернення: 08.02.2023 р.).
20. MS Hosen. Mastering 3D Modeling in Blender: From Novice to Pro. 2019. с. 107-140 (дата звернення: 16.02.2023 р.).
21. Blender Documentation: офіційний сайт. URL: <https://docs.blender.org/> (дата звернення: 20.02.2023 р.).
22. Unreal Engine 5 Documentation: офіційний сайт. URL: <https://docs.unrealengine.com/5.0/en-US/> (дата звернення: 27.02.2023 р.).
23. Howard, M. C., Van Zandt, E. C. A meta-analysis of the virtual reality problem: Unequal effects of virtual reality sickness across individual differences. *Virtual Reality*, 2021. с. 1221-1246. URL: <https://doi.org/10.1007/s10055-021-00524-3> (дата звернення: 30.02.2023 р.).
24. Tom Looman. Unreal engine: Naming convention guide. 2022. с. 132-156 URL: <https://www.tomlooman.com/unreal-engine-naming-convention-guide> (дата звернення: 05.03.2023 р.).
25. Marcos Romero, Brenden Sewell, Luis Cataldi. Blueprints Visual Scripting for Unreal Engine 5: Unleash the true power of Blueprints to create impressive games and applications in UE5. 2023. (дата звернення: 20.03.2023 р.).
26. F Sapio. Hands-on artificial intelligence with Unreal Engine: Everything you want to know about game AI using blueprints or C++. 2019. (дата звернення: 25.03.2023 р.).
27. Zhen Yu George Li, Dr. E. Wyn Roberts. Unreal Engine 5 Game Development with C++ Scripting. 2023. (дата звернення: 01.04.2023 р.).
28. K Mack, R Ruud. Unreal Engine 4 Virtual Reality Projects: Build Immersive, Real-world VR Applications Using UE4, C++, and Unreal Blueprints. 2019. (дата звернення: 05.04.2023 р.).

29. Zhenyu George Li. Unreal Engine 5 Game Development with C++ Scripting. 2023. (дата звернення: 09.04.2023 р.).
30. Laura Inzerillo, Francesco Di Paola, Yuri Alogna. High quality texture mapping process aimed at the optimization of 3d structured light models. 2019. URL: <https://isprs-archives.copernicus.org/articles/XLII-2-W9/389/2019/isprs-archives-XLII-2-W9-389-2019.pdf> (дата звернення: 09.04.2023 р.)
31. Наконечна І.М., Толстолузька О.Г., Розробка комп'ютерної моделі віртуального музею з елементами віртуального середовища, Комп'ютерне моделювання у наукоємних технологіях : зб. наукових праць VIII Міжнар. наук.-техн. конф. Харків : ХНУ імені В.Н. Каразіна, 2023., с. 20-22 (дата звернення: 12.11.2023 р.)

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Магістр
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри теоретичної та
прикладної системотехніки
д.т.н., проф. Шматков С. І.



«08 » грудня 2022 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ Наконечна Інна Михайлівна

1. Тема роботи «Комп'ютерна модель віртуального музею з елементами віртуального середовища».

керівник роботи Толстолузька Олена Геннадіївна, д.т.н., с.н.с.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «10» листопада 2023 року № 4101-5/3197

2. Строк подання студентом роботи 28 .11.2023

2. Перелік питань, які потрібно розробити

1) Обґрунтування вибору технологій.

2) Огляд та аналіз існуючих методів створення віртуальних середовищ.

3) Розробка вимог та реалізація 3D-моделей.

4) Розробка алгоритмів взаємодії.

5) Програмна реалізація проекту.

6) Перевірка на працездатність та оцінка результатів.

7) Підготовка та оформлення пояснювальної записки.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Огляд та аналіз існуючих рішень у створенні віртуальних музеїв.	08.12.2022- 20.12.2022
2	Розробка концепції віртуального музею та його структури.	20.12.2022- 20.01.2023
3	Вибір і обґрунтування вибору технологічних засобів для створення віртуального середовища музею.	20.01.2023- 17.02.2023
4	Розробка вимог до 3D-моделей.	17.02.2023- 17.03.2023
5	Розробка 3D-моделей та текстур.	17.04.2023- 25.06.2023
6	Розробка алгоритмів взаємодії та навігації користувача.	25.06.2023- 02.08.2023
7	Програмна реалізація проекту.	02.08.2023- 04.09.2023
8	Тестування та оцінка результатів.	04.09.2023- 14.09.2023
9	Оптимізація та виправлення помилок.	14.09.2023- 30.09.2023
10	Оформлення пояснювальної записки.	30.09.2023- 28.11.2023

5. Дата видачі завдання 08.12.2022

Студентка

Наконечна І. М.

ініціали, прізвище


підпис

Керівник роботи

Толстолюзька О.Г.

ініціали, прізвище


підпис

Затверджую

«_____» _____ 2022 р.

Технічне завдання на розробку програмного виробу «Комп'ютерна модель віртуального музею з елементами віртуального середовища»

1.	Введення	1.1. Назва: Комп'ютерна модель віртуального музею з елементами віртуального середовища. 1.2. Галузь застосування: освіта, культура, туризм.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу магістра № __4101-5/3197_ від «_10»_ листопада 2023 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: Створити комп'ютерну модель віртуального музею, що відображає історичні та культурні об'єкти України в інтерактивному віртуальному середовищі. 3.2. Призначення розробки призначений для освітніх та туристичних цілей, а також для збереження та популяризації культурних та історичних цінностей України. 3.3. Вихідні дані розробки: Функціональна комп'ютерна модель віртуального музею, яка забезпечує інтерактивний перегляд 3D-моделей історичних та культурних об'єктів України у віртуальній реальності. Модель включає можливості VR навігації, обертання об'єктів на 360 градусів, аудіо гідів та текстову інформацію, доступна у двох мовах: українській та англійській. Відскановані та оптимізовані історичні та культурні пам'ятки України
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: немає 4.2. Вимоги до надійності: безперебійна робота виробу 4.3. Вимоги до умов експлуатації: немає 4.4. Вимоги до складу і параметрів технічних засобів: реальності (шолом) та контролери. Комп'ютер:

		<ul style="list-style-type: none"> • Процесор (CPU): Рекомендується використовувати мінімум Intel Core i5-4590 або AMD Ryzen 5 1500X. • Відеокарта (GPU): NVIDIA GTX 1060/AMD Radeon RX 480 або краща. Для високоякісного рендеринга в реальному часі рекомендуються карти з класу NVIDIA RTX 2080 або вище. • Оперативна пам'ять (RAM): Мінімум 8GB, але рекомендується 16GB або більше. • Порти: Мінімум 1x USB 3.0 або новіший порт, а також порти HDMI 1.3. <p>Гарнітура VR (Oculus Go):</p> <ul style="list-style-type: none"> • Дисплей: Fast-Switch LCD з роздільною здатністю 2560 x 1440. • Частота оновлення: 60Hz (з деякими додатками, що підтримують 72Hz). • Процесор: Qualcomm Snapdragon 821. • Оперативна пам'ять: 3GB RAM. • Зберігання: Версії на 32GB або 64GB. • Слідкування: 3-ступінчасте слідкування. • Батарея: Приблизно 2 години використання. <p>Контролер(Oculus Go):</p> <ul style="list-style-type: none"> • 3-ступінчасте слідкування, синхронізоване зі шоломом. • Кнопки: Сенсорний пульт з тригером, сенсорною поверхнею, кнопкою "Назад" та кнопкою меню. <p>4.5. Вимоги до інформаційної та програмної сумісності: немає</p> <p>4.6. Вимоги до маркування та упаковки: немає</p> <p>4.7. Вимоги до транспортування і зберігання: на звичайних носіях інформації</p> <p>4.8. Спеціальні вимоги: немає.</p>
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «Комп'ютерна модель віртуального музею з елементами віртуального середовища» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p>

		<p>2) Розроблена комп'ютерна модель інтерактивного віртуального музею (у вигляді розділу 2 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Рекомендацій по використанню (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
6.	Вимоги до техніко-економічних показників	<p>Програмною документацією до виробу «Комп'ютерна модель віртуального музею з елементами віртуального середовища» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Розроблена комп'ютерна модель інтерактивного віртуального музею (у вигляді розділу 2 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Рекомендацій по використанню (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		від 8 грудня 2022 до 20 грудня 2022	Огляд та аналіз існуючих методів створення віртуальних музеїв.
		від 20 грудня 2022 до 20 січня 2023	Розробка концепції віртуального музею та його структури.
		від 20 січня 2023 до 17 лютого 2023	Вибір і обґрунтування вибору технологічних засобів для створення
		від 17 лютого 2023 до 17 березня 2023	Розробка вимог до 3D-моделей.
		від 17 березня 2023 до 25 червень 2023	Розробка 3D-моделей та текстур.
		від 25 червня 2023 до 2 серпня 2023	Розробка алгоритмів взаємодії та навігації користувача.

		<p>від 2 серпня 2023 до 4 вересня 2023</p> <p>від 4 вересня 2023 до 14 вересня 2023</p> <p>від 14 вересня 2023 до 30 жовтня 2023</p> <p>від 30 жовтня 2023 до 23 листопада</p>	<p>Програмна реалізація проекту.</p> <p>Тестування та оцінка результатів.</p> <p>Оптимізація та виправлення помилок.</p> <p>Оформлення пояснювальної записки. Представлення кваліфікаційного проекту керівнику кваліфікаційної роботи та рецензенту.</p>
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> 1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику на CD-R компакт-диску. 	

Виконавець
студент групи КІ- 61
Наконечна І. М.



Замовник
Д. Т. Н.,
Толстолузька О. Г.



**Програма і методика випробувань
програмного виробу**

«Комп'ютерна модель віртуального музею з елементами віртуального
середовища»

1 Об'єкт випробувань

1.1 Найменування випробуваного програмного виробу: Комп'ютерна модель віртуального музею з елементами віртуального середовища.

1.2 Область його застосування: освіта, культура, туризм.

2. Мета випробувань

Перевірка працездатності продукту та правильності роботи розробленої моделі. Підтвердження характеристик розробленого програмного виробу вимогам, які сформульовані в ТЗ (Додаток Б).

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії».

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні».

4. Вимоги до програми або програмного виробу

Модель повинна відтворювати історичні та культурні пам'ятки України, представляти можливість користувачам переглядати їх на відстані, взаємодіяти з ними: обертати, масштабувати, надати текстову та аудіо інформацію про експонати.

5. Вимоги до програмної документації

Склад програмної документації, що подається на випробування, включає:

- 1) Технічне завдання на розробку програмного виробу (представлено в Додатку Б до пояснювальної записки до кваліфікаційної роботи).
- 2) Ця Програма і методика випробувань розробленого програмного виробу (представлена в Додатку В до пояснювальної записки до кваліфікаційної роботи).
- 3) Опис програмного виробу (представлено в розділі 3 пояснювальної записки до кваліфікаційної роботи).
- 4) Текст програми (представлений в Додатку Г до пояснювальної записки до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Комп'ютер:

- Процесор (CPU): Рекомендується використовувати мінімум Intel Core i5-4590 або AMD Ryzen 5 1500X.
- Відеокарта (GPU): NVIDIA GTX 1060/AMD Radeon RX 480 або краща. Для високоякісного рендеринга в реальному часі рекомендуються карти з класу NVIDIA RTX 2080 або вище.
- Оперативна пам'ять (RAM): Мінімум 8GB, але рекомендується 16GB або більше.
- Порти: Мінімум 1x USB 3.0 або новіший порт, а також порти HDMI 1.3.

Гарнітура VR (Oculus Go):

- Дисплей: Fast-Switch LCD з роздільною здатністю 2560 x 1440.
- Частота оновлення: 60Hz (з деякими додатками, що підтримують 72Hz).
- Процесор: Qualcomm Snapdragon 821.
- Оперативна пам'ять: 3GB RAM.

- Зберігання: Версії на 32GB або 64GB.
- Слідкування: 3-ступінчасте слідкування.
- Батарея: Приблизно 2 години використання.

Контролер(Oculus Go):

- 3-ступінчасте слідкування, синхронізоване зі шоломом.
- Кнопки: Сенсорний пульт з тригером, сенсорною поверхнею, кнопкою "Назад" та кнопкою меню.

6.2 Порядок проведення випробувань

6.2.1 Перевірку комплектності програмної документації.

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

- 1) Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
- 2) Перевірку комплектності складу технічних і програмних засобів.
- 3) Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

6.2.2 Перевірка роботоздатності моделі.

Користувач запускає музей, після чого потрапляє на головний екран, та вибирає пристрій клавіатуру та миш Рис В.1

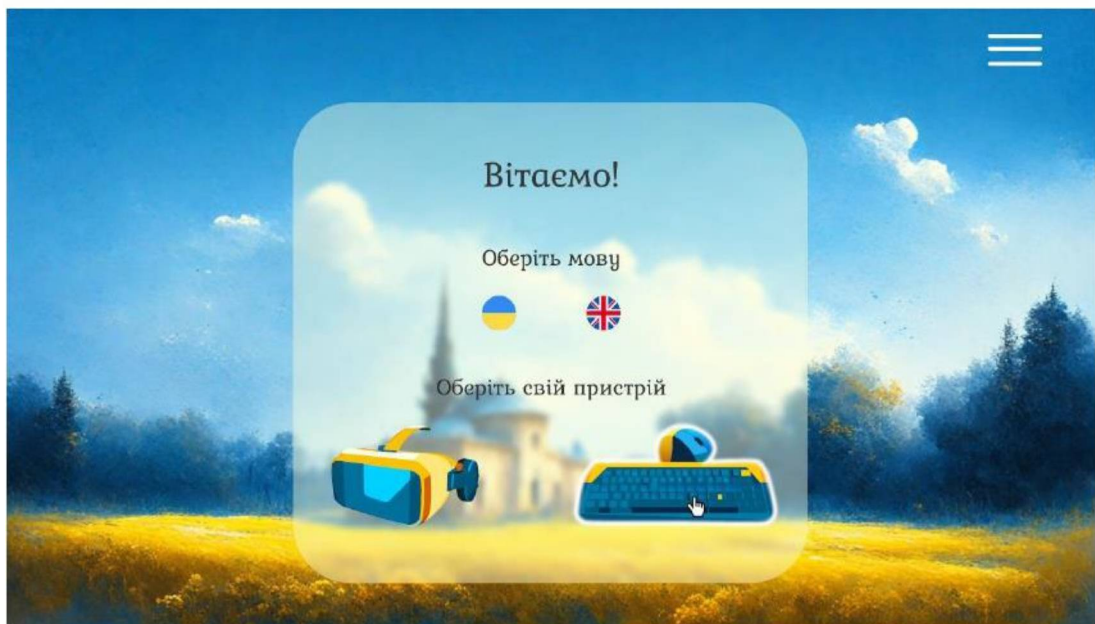


Рис В.1 — Головний екран, клавіатура та миш

Він легко переміщається по музею. Рис В.2



Рис В.2 — Навігація

Та взаємодіє з об'єктами: масштабує, обертає, отримує інформацію про експонат. Рис В.3



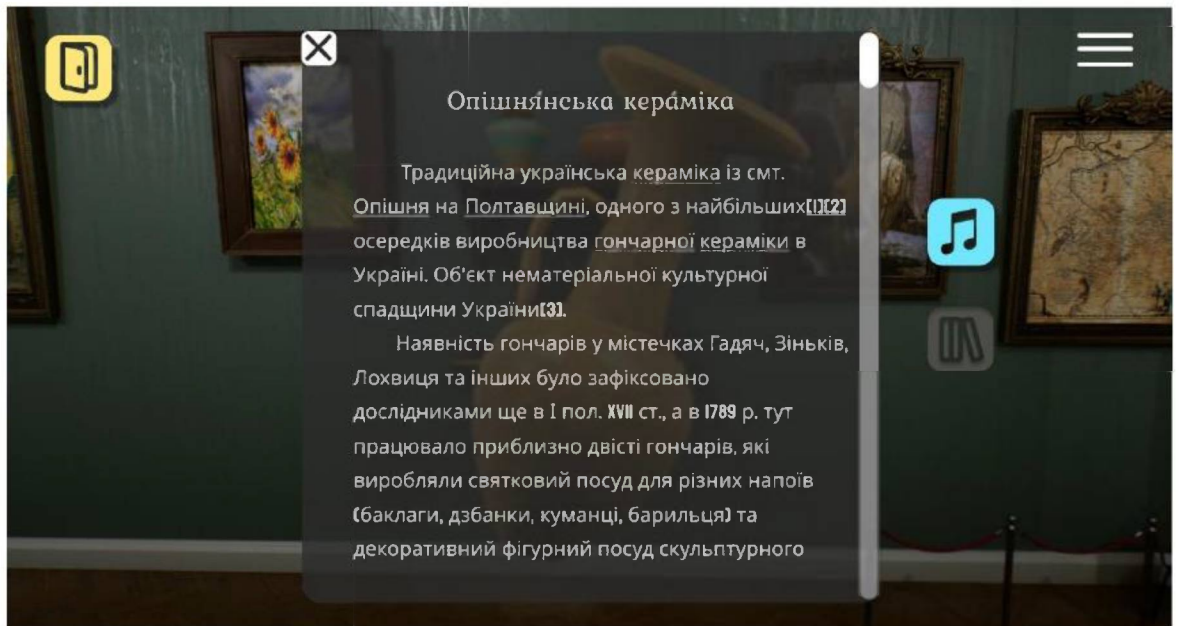


Рис В.3-4 — Взаємодія з експонатами

Виконавець

Наконачна І. М.

Код програми

VRCharacter.h:

```

1 // VRCharacter.h
2 #include "CoreMinimal.h"
3 #include "GameFramework/Character.h"
4 #include "VRCharacter.generated.h"
5
6 UCLASS()
7 class VIRTUALMUSEUM_API AVRCharacter : public ACharacter
8 {
9     GENERATED_BODY()
10
11 public:
12     // Конструктор для створення VR персонажа
13     AVRCharacter();
14
15 protected:
16     // Функція, яка ініціалізує VR персонажа під час створення
17     virtual void BeginPlay() override;
18
19 public:
20     // Функція оновлення кожного кадру
21     virtual void Tick(float DeltaTime) override;
22
23     // Функції для переміщення персонажа
24     void MoveForward(float Value);
25     void MoveRight(float Value);
26 };
27
28

```

Рис Г.1 — Клас користувача

VRCharacter.cpp:

```

// VRCharacter.cpp
#include "VRCharacter.h"
void AVRCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
{
    Super::SetupPlayerInputComponent(PlayerInputComponent);

    // Об'єднане управління для VR контролерів та клавіатури/миші
    PlayerInputComponent->BindAxis("MoveForward", this, &AVRCharacter::MoveForward);
    PlayerInputComponent->BindAxis("MoveRight", this, &AVRCharacter::MoveRight);

    // Обертання персонажа
    PlayerInputComponent->BindAxis("Turn", this, &APawn::AddControllerYawInput);
    PlayerInputComponent->BindAxis("LookUp", this, &APawn::AddControllerPitchInput);
}

void AVRCharacter::MoveForward(float Value)
{
    if (Value != 0.0f)
    {
        // Отримання напрямку руху вперед/назад
        FVector ForwardDirection = GetActorForwardVector();
        // Додаємо рух у напрямку цього вектора
        AddMovementInput(ForwardDirection, Value);
    }
}

void AVRCharacter::MoveRight(float Value)
{
    if (Value != 0.0f)
    {
        // Отримання напрямку руху ліво/вправо
        FVector RightDirection = GetActorRightVector();
        // Додаємо рух у напрямку цього вектора
        AddMovementInput(RightDirection, Value);
    }
}

```

Рис Г.2 — Функції навігації користувача

ExhibitObject.h:

```
// ExhibitObject.h
#include "CoreMinimal.h"
#include "GameFramework/Actor.h"
#include "ExhibitObject.generated.h"

UCLASS()
class VIRTUALMUSEUM_API AExhibitObject : public AActor
{
    GENERATED_BODY()

public:
    AExhibitObject();

    // Функції для управління станом об'єкта
    void SelectObject();
    void DeselectObject();

private:
    // Зберігаємо початкові параметри об'єкта
    FVector OriginalScale;
    FVector OriginalLocation;
};
```

Рис Г.3 — Клас експоната музею

ExhibitObject.cpp:

```
// ExhibitObject.cpp
#include "ExhibitObject.h"

AExhibitObject::AExhibitObject()
{
    // Конструктор класу
    // Ініціалізація компонентів, якщо потрібно
}

void AExhibitObject::SelectObject()
{
    // Запам'ятовуємо початкові параметри об'єкта
    OriginalScale = GetActorScale3D();
    OriginalLocation = GetActorLocation();

    // Можна також змінити вигляд об'єкта, наприклад, підсвітити його
}

void AExhibitObject::DeselectObject()
{
    // Відновлюємо початкові параметри об'єкта
    SetActorScale3D(OriginalScale);
    SetActorLocation(OriginalLocation);

    // Вимикаємо підсвітку, якщо вона була включена
}
}
```

Рис Г.4 — Функції експоната музею

Функції взаємодії з експонатами:

```
// Визначення об'єкта, на який вказує користувач
void AVRCharacter::SelectObject()
{
    FVector Start, Dir;
    FHitResult Hit;

    // Отримання позиції та напрямку вказівника
    GetPointerData(Start, Dir);

    // Виконання трасування променів
    if (GetWorld()->LineTraceSingleByChannel(Hit, Start, Start + Dir * TraceDistance, ECC_Visibility))
    {
        // Перевірка, чи було щось виявлено
        if (AExhibitObject* SelectedObject = Cast<AExhibitObject>(Hit.Actor))
        {
            // Збереження вибраного об'єкта для подальшої обробки
            CurrentSelectedObject = SelectedObject;
        }
    }
}

// Функція для отримання позиції та напрямку вказівника
void AVRCharacter::GetPointerData(FVector& OutStart, FVector& OutDir)
{
    // Отримання позиції та орієнтації VR контролера або курсора миші
}

```

Рис Г.5 — Функції визначення об'єкта, на який вказує користувач

```
void AVRCharacter::ScaleSelectedObject()
{
    if (CurrentSelectedObject)
    {
        // Масштабування об'єкта для його відображення на весь екран
        CurrentSelectedObject->SetActorScale3D(FVector(ZoomScale));

        // Визначаємо позицію перед користувачем на певній відстані
        FVector NewLocation = GetActorLocation() + GetActorForwardVector() * DistanceInFrontOfPlayer;

        // Переміщення об'єкта до відповідного місця для перегляду
        CurrentSelectedObject->SetActorLocation(NewLocation);
    }
}

```

Рис Г.6 — Функція масштабування об'єкта

```
void AVRCharacter::RotateSelectedObject(float Horizontal, float Vertical)
{
    if (CurrentSelectedObject)
    {
        // Обертання об'єкта
        CurrentSelectedObject->AddActorLocalRotation(FRotator(Vertical, Horizontal, 0.0f));
    }
}

```

Рис Г.7 — Функція обертання об'єкта

```

void AVRCharacter::CloseObjectView()
{
    if (CurrentSelectedObject)
    {
        // Повернення об'єкта до його початкового розміру та позиції
        CurrentSelectedObject->SetActorScale3D(OriginalScale);
        CurrentSelectedObject->SetActorLocation(OriginalLocation);

        // Очищення вибраного об'єкта
        CurrentSelectedObject = nullptr;
    }
}

```

Рис Г.8 — Функція закриття Масштабування

```

// MuseumInterface.h
#pragma once

#include "CoreMinimal.h"
#include "Blueprint/UserWidget.h"
#include "MuseumInterface.generated.h"

UCLASS()
class VIRTUALMUSEUM_API UMuseumInterface : public UUserWidget
{
    GENERATED_BODY()

public:
    // Функції для перемикання аудіогіда та текстової інформації
    UFUNCTION(BlueprintCallable, Category = "UI")
    void ToggleAudioGuide();

    UFUNCTION(BlueprintCallable, Category = "UI")
    void ToggleTextInfo();

private:
    // Елементи інтерфейсу: іконки, текстовий блок, аудіокомпонент
    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UTexture2D* AudioGuideIcon;

    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UTexture2D* TextInfoIcon;

    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UTextBlock* TextInfoWidget;

    UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = "UI", meta = (AllowPrivateAccess = "true"))
    class UAudioComponent* AudioGuideAudioComponent;

    // Змінні для відстеження стану аудіогіда та текстової інформації
    bool bAudioGuideActive;
    bool bTextInfoActive;
};

```

Рис Г.9 — Інтерфейс аудіогіда та текстової інформації

MuseumInterface.cpp:

```
// MuseumInterface.cpp
#include "MuseumInterface.h"
#include "Components/Image.h"
#include "Components/TextBlock.h"
#include "Components/AudioComponent.h"

void UMuseumInterface::ToggleAudioGuide()
{
    // Перемикаємо стан аудіогіда
    bAudioGuideActive = !bAudioGuideActive;

    if (bAudioGuideActive)
    {
        // Запускаємо аудіогід та змінюємо іконку
        AudioGuideAudioComponent->Play();
        Cast<UImage>(GetWidgetFromName(TEXT("AudioGuideIcon"))->SetBrushFromTexture(AudioGuideIcon));
    }
    else
    {
        // Вимикаємо аудіогід та змінюємо іконку
        AudioGuideAudioComponent->Stop();
        Cast<UImage>(GetWidgetFromName(TEXT("AudioGuideIcon"))->SetBrushFromTexture(/*previous icon*/));
    }
}

void UMuseumInterface::ToggleTextInfo()
{
    // Перемикаємо стан текстової інформації
    bTextInfoActive = !bTextInfoActive;

    if (bTextInfoActive)
    {
        // Відображаємо текстову інформацію та робимо видимим
        TextInfoWidget->SetText(/*your text data*/);
        TextInfoWidget->SetVisibility(ESlateVisibility::Visible);
    }
    else
    {
        // Приховуємо текстову інформацію
        TextInfoWidget->SetVisibility(ESlateVisibility::Hidden);
    }
}
```

Рис Г.10 — Інтерфейс аудіогіда та текстової інформації

Виконавець

Накочна І. М.

