Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка об’єктів Virtual Reality (VR) засобами Unreal Engine4»

Студента 2м курсу, 6 групи, спеціальності 121 «Інженерія програмного забезпечення» спеціалізації «Інженерія програмного забезпечення»

_____
підпис студента

Радько Максима Андрійович

Науковий керівник
кандидат технічних наук, доцент кафедри інженерії програмного забезпечення та кібербезпеки

_____
підпис керівника

Рассамакін Володимир Якович

Гарант освітньої програми
доктор технічних наук, професор кафедри інженерії програмного забезпечення та кібербезпеки

_____
підпис гаранта

Криворучко Олена Володимирівна

КИЇВ - 2020

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення та кібербезпеки
Освітній ступінь магістр
Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую
Зав. кафедри інженерії програмного забезпечення та кібербезпеки
Криворучко О. В.
8 листопада 2019 р.

Завдання
на випускний кваліфікаційний проект студентові
Радько Максиму Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка об'єктів Virtual Reality (VR) засобами Unreal Engine4»

Затверджена наказом ректора від "13" грудня 2019 р. № 4304

2. Строк здачі студентом закінченої проекту 01 грудня 2020

3. Цільова установка та вихідні дані до проекту

Мета проекту дослідження існуючих підходів і концепцій та створення 3D образів музейних експонатів для віртуального музею КНТЕУ.

Об'єкт дослідження віртуалізація музейного простору історії КНТЕУ.

Предмет дослідження технічні та програмні засоби відтворення 3D-образів музейних експонатів КНТЕУ

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ І КОНЦЕПЦІЙ

1.1. Віртуальна реальність в сучасній іт- індустрії

1.2. Аналіз існуючих програмних засобів

1.3. Розробка загальної концепції проекту

1.4. Висновок до розділу 1

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ОБ*ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE4

2.1. Вимоги до технічного завдання

2.2. Опис етапів розробки 3d моделей

2.2.1. Modeling

2.2.2. Retopology

2.2.3. UV Mapping

2.2.4. Rigging

2.3. Висновки до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОБ*ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE4

3.1. Використане ПЗ

3.2. Етапи створення об`єктів vr засобами unreal engine4

3.3. Експорт готового проекту

3.4. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
•	*Вибір теми випускного кваліфікаційного проект*	*20.09.2019*	*20.09.2019*
•	*Розробка та затвердження завдання на проект магістра (Наказ №185 від 07.11.19)*	*08.11.2019*	*08.11.2019*
•	*Вступ та перелік літературних джерел*	*24.01.2020*	*24.01.2020*
•	*Наукова стаття*	*01.09.2020*	*01.09.2020*
•	*Технічне завдання*	*28.02.2020*	*28.02.2020*
•	*Розділ 1. Дослідження існуючих підходів і концепцій*	*25.06.2020*	*25.06.2020*
•	*Розділ 2. Технічне завдання на розробку об*єктів vr засобами unreal engine4*	*07.09.2020*	*07.09.2020*
•	*Розділ 3. Реалізація об*єктів vr засобами unreal engine4*	*19.10.2020*	*19.10.2020*
•	*Програма та методика тестування*	*21.10.2020*	*21.10.2020*
•	*Керівництво користувача*	*23.10.2020*	*23.10.2020*
•	*Висновки та пропозиції*	*30.10.2020*	*30.10.2020*
•	*Здача випускного кваліфікаційного проекту на кафедру (перша перевірка)*	*05.11.2020*	*05.11.2020*
•	*Підготовка автореферату та презентації доповіді*	*05.11.2020*	*05.11.2020*
•	*Попередній захист випускного кваліфікаційного проекту*	*25.11.2020-27.11.2020*	*25.11.2020-27.11.2020*
•	*Зовнішнє рецензування випускного кваліфікаційного проекту*	*01.12.2020*	*01.12.2020*
•	*Підготовка до публічного захисту випускного кваліфікаційного проекту*	*08.12.2020-09.12.2020*	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту Рассамакін В.Я.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Криворучко О.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Радько М.А.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проекту

_____

Науковий керівник випускного кваліфікаційного проекту

_____

(підпис, дата)

Відмітка про попередній захист _____Рассамакін В.Я._____

(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проект

Випускний кваліфікаційний проект студента _____Радько М.А.____

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____Криворучко О. В.

(прізвище, ініціали, підпис)

Завідувач кафедри _____Криворучко О. В.

(підпис, прізвище, ініціали)

«_____»_____20 ___ р.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці програмного забезпечення музею КНТЕУ у віртуальній реальності. Ця програма є покращеним варіантом музею КНТЕУ у 3д та є сумісною з сучасними контролерами віртуальної реальності. Обрана мова програмування – C++ та BluePrints Visual Scripting. Готове програмне забезпечення було успішно протестовано відповідно до функціональних вимог.

Ключові слова: Віртуальна реальність, програмне забезпечення, КНТЕУ, музей, 3д.

ABSTRACT

In accordance with the purpose of the study, the work is devoted to the development of software of the KNUTE museum in virtual reality. This program is an improved version of the KNUTE Museum in 3D and is compatible with modern virtual reality controllers. Selected programming language - C ++ and BluePrints Visual Scripting. The finished software has been successfully tested in accordance with the functional requirements.

Keywords: Virtual reality, software, KNUTE, museum, 3d.

KHTEY KNUTE

ЗМІСТ

ВСТУП **3**
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ І КОНЦЕПЦІЙ... ..... **4**
1.1 ВІРТУАЛЬНА РЕАЛЬНІСТЬ В СУЧАСНІЙ ІТ- ІНДУСТРІЇ **4**
1.2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ **5**
1.3 РОЗРОБКА ЗАГАЛЬНОЇ КОНЦЕПЦІЇ ПРОЕКТУ **7**
1.4 ВИСНОВКИ ДО РОЗДІЛУ 1 ..... **7**
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ОБ*ЄКТІВ VR ЗАСОБАМИ……..
UNREAL ENGINE4 ..... **8**
2.1 ВИМОГИ ДО ТЕХНІЧНОГО ЗАВДАННЯ **8**
2.2 ОПИС ЕТАПІВ РОЗРОБКИ 3D МОДЕЛЕЙ **9**
2.2.1 MODELING 9
2.2.2 RETOPOLOGY 10
2.2.3 UV MAPPING 10
2.2.4 RIGGING 11
2.3 ВИСНОВКИ ДО РОЗДІЛУ 2 ..... **11**
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОБ*ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE4 ..... **12**
3.1 ВИКОРИСТАНЕ ПЗ **12**
3.2 ЕТАПИ СТВОРЕННЯ ОБ*ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE4 ..... **13**
3.3 ЕКСПОРТ ГОТОВОГО ПРОЕКТУ **26**
3.4 ВИСНОВКИ ДО РОЗДІЛУ 3 ..... **28**
ВИСНОВКИ ТА ПРОПОЗИЦІЇ **29**
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **30**
ТЕХНІЧНЕ ЗАВДАННЯ……**31**
ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ……**32**

ВСТУП

VR - це наше майбутнє. Як тільки будуть виправлені всі недоліки, і буде вирішено питання з адаптацією людини до нового середовища, ця технологія отримає масове застосування в усіх сферах нашого життя, спростить її і зробить краще. Лікарі зможуть підготуватися до складних операцій, авіація отримає нові симулятори, які дозволять набагато ефективніше готувати персонал, а прості люди зможуть максимально комфортно проводити своє дозвілля, наприклад, за переглядом фільму або занурившись в цікаву гру.

Але умови для масового впровадження практики створення віртуальних музеїв в наш час, зокрема у закладах вищої освіти повною мірою не створені. Тому подальше дослідження спрямоване на пошук можливостей створення умов для його реалізації, в першу чергу, на прикладі створення віртуального музейного простору університету.

● Точність, яку містять тривимірні моделі, не має аналогів і жоден інший звичайний підхід до моделювання або візуалізації не може бути настільки точним, як 3D-моделювання.

Мета роботи – дослідження існуючих підходів і концепцій та створення 3D образів музейних експонатів для віртуального музею КНТЕУ.

Об'єкт дослідження – віртуалізація музейного простору історії КНТЕУ.

Предмет дослідження: технічні та програмні засоби відтворення 3D-образів музейних експонатів КНТЕУ

Завдання дослідження: аналіз вітчизняних і зарубіжних публікацій та інформаційних джерел по темі науково-дослідної роботи.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ І КОНЦЕПЦІЙ

1.1 ВІРТУАЛЬНА РЕАЛЬНІСТЬ В СУЧАСНІЙ ІТ- ІНДУСТРІЇ

VR - це вигаданий плід розумової діяльності людини. Це те, що існує не в нашій голові, а в комп'ютерному коді. Віртуальна реальність і доповнена реальність вражають своїми масштабами: відео з оглядом в 360 градусів, цілі ігрові світи зі своїми законами, особливостями і персонажами.

Іноді складно зрозуміти, де реальний об'єкт, а де творіння комп'ютера - настільки високого рівня досягли технології.

Багато вузів переводять частину предметів на дистанційне навчання. Їм простіше і вигідніше запросити іменитого професора і транслювати його лекції по скайпу, ніж возити його по всіх містах, де є філії цього вузу.

Недовго залишиться чекати, поки ця модель навчання прийде в наші школи. Вона дуже вигідна для учнів, адже куди простіше надіти окуляри віртуальної реальності, ніж сидіти в задушливому класі, лежати вдома на дивані, але при цьому перебувати в оточенні однокласників і вчителів.

VR активно використовується в медицині: хірургії, психології та реабілітації. Як отримати досвід, якщо не робити операції, найчастіше такої можливості у студентів немає. Але проблема вирішується, якщо хірурги починають використовувати ВР для тренувань.

VR дозволяє відновлюватися після операції і травм, наприклад, системи використовуються для додаткової стимуляції з травмами спинного мозку.

1.2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ

Adobe Photoshop - багатофункціональний графічний редактор, розроблений фірмою Adobe Systems. В основному працює з растровими зображеннями, проте має деякі векторні інструменти. Продукт є лідером ринку в області комерційних засобів редагування растрових зображень і найбільш відомим продуктом фірми Adobe.

Blender - відкрите програмне забезпечення для створення тривимірної комп'ютерної графіки, що включає в себе засоби моделювання, анімації, рендеринга, постобробки і монтажу відео зі звуком, а також для створення інтерактивних ігор. В даний час користується найбільшою популярністю серед безкоштовних 3D-редакторів в зв'язку з його швидким і стабільним розвитком, якому сприяє професійна команда розробників. Характерною особливістю пакету Blender є його невеликий розмір в порівнянні з іншими популярними пакетами для 3D-моделювання.

Mixcraft - це програма багатоканального запису для Windows. Це програмне забезпечення для запису музики функціонує як робоча станція цифрового звуку, MIDI секвенсор, віртуальний хост приладу, нелінійний аранжувальник відео та програма запису музичних циклів.

Основні можливості програми

- MIDI: Ця функція дозволяє записувати MIDI-дані, які відтворюються на одній або декількох клавіатурах MIDI-контролера.
- Multitrack Recording: Ця функція дозволяє користувачам створювати записи з декількох звукових / аудіовходів одночасно.

Unreal Engine 4 - ігровий рушій, розроблений і підтримуваний компанією Epic Games. Написаний на мові C ++, рушій дозволяє створювати ігри для більшості операційних систем і платформ: Microsoft

Windows, Linux, Mac OS і Mac OS X, консолі Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast і Nintendo GameCube.

Код рушія може використовуватися в будь-яких комерційних проектах, але з розробників даних проектів здійснюється відрахування у розмірі 5% від доходу, отриманого від продажу гри користувачам. Збір відрахувань з доходів від продажу робить рушій Unreal Engine 4 цікавим рішенням не тільки для великих ігрових проектів, але і для невеликих стартапів і авторів безкоштовних ігор.

Adobe Fuse CC - (раніше Fuse Character Creator) - це програмне забезпечення для комп'ютерної графіки 3D, розроблене компанією Mixamo, що дозволяє користувачам створювати 3D-персонажі. Її головна новизна - можливість імпортувати та інтегрувати вміст, створений користувачем, до створення персонажа. Fuse є частиною набору продуктів Mixamo, і призначений для розробників відеоігор, модераторів відеоігор та 3D-ентузіастів.

Fuse - продукт на основі клієнта, що дозволяє користувачам вибирати та змінювати компоненти символів, такі як частини тіла, в режимі реального часу. Основною новинкою Fuse є можливість імпортувати та автоматично інтегрувати свій власний вміст у систему створення персонажа, використовуючи всі функції попередньо завантаженого контенту.

Fuse використовується в основному для розробників ігор і модераторів ігор. Fuse доступний на Mixamo, на Unity Asset Store і на ринку Steam, де з вересня 2014 року він має 40 000 користувачів. Персонажів Fuse можна імпортувати через певну серію кроків у ігри, створені за допомогою Source SDK.

1.3 РОЗРОБКА ЗАГАЛЬНОЇ КОНЦЕПЦІЇ ПРОЕКТУ

В даній роботі реалізується рухома модель персонажу зі змогою взаємодії з навколишнім середовищем. Із навколишнього середовища буде створений музей КНТЕУ, відтворена його архітектурна форма, внутрішній інтер'єр та оздоблення.

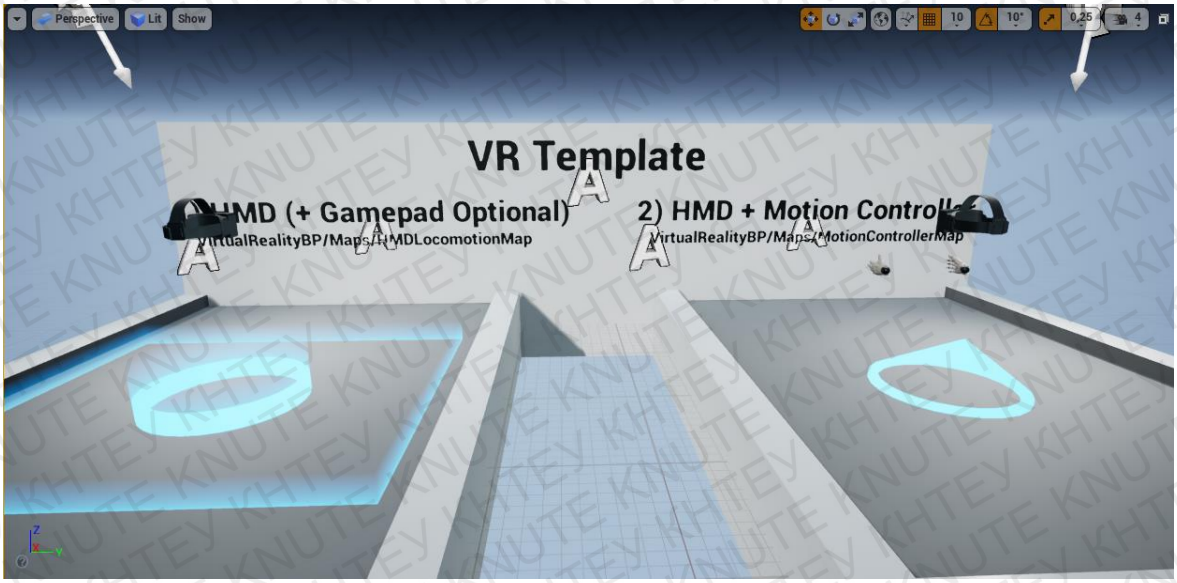

Рис. 1.1 віртуальні контролери персонажа

1.4 ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ

В першому розділі була вивчена потрібність віртуальної реальності та 3D моделей в IT індустрії, проаналізовані існуючі програмні засоби для розробки 3D моделей та розроблена загальна концепція музею КНТЕУ.

РОЗДІЛ 2 ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ОБ*ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE 4

2.1. ВИМОГИ ДО ТЕХНІЧНОГО ЗАВДАННЯ

Мета - підвищення ефективності освітнього процесу в сучасних умовах за рахунок використання нових інформаційних технологій (VR), що забезпечують багатофункціональність, багатоваріантність, гнучкість і оперативність;

Завдання дослідження: аналіз вітчизняних і зарубіжних публікацій та інформаційних джерел по темі науково-дослідної роботи.

1. Вимоги по розробці об*єктів:

- Об’єкти повинні відповідати дійсності.
- Об’єкти повинні надавати достовірну інформацію.
- Персонаж повинен мати змогу керуватися апаратними пристроями та контролерами віртуальної реальності.

2. Розміри

Музей у своєму фінальному вигляді повинен відповідати приблизним розмірам справжньої кімнати.

3. Анімація

В готовому проекті має бути анімований персонаж, для того щоб пересуватись по музею без перешкод. Також повинна бути створена анімація відчинення та зачинення дверей.

4. Взаємодія

В готовому проекті мають бути декілька експонатів музею з якими персонаж зможе взаємодіяти, наприклад отримати детальну

інформацію про історичну подію або постать.

2.2 ОПИС ЕТАПІВ РОЗРОБКИ 3D МОДЕЛЕЙ

2.2.1 Modeling

Моделювання - це процес візуального зображення об'єктів і поверхонь для подальших маніпуляцій. Якщо потрібно створити статичний 3D-актив, який не буде виконувати складний набір анімації, моделювання стане необхідним способом його створення. Весь процес моделювання може бути завершений багатьма способами, залежно від того, який з них є найбільш привабливим для 3D-художника.

- Примітивне моделювання означає створення простих 3D-об'єктів, таких як сфери, кубики, конуси, змінюючи їх атрибути. Використовуючи Modeling Toolkit, виконавець може розділяти, витягувати, об'єднувати або видаляти компоненти з об'єктів, щоб зробити примітивний об'єкт більш-менш складним.
- Моделювання кривих - це метод створення 3D-активів, коли поверхні маніпулюють кривими, які контролюються зваженими контрольними точками. Змінюючи вагу для певної точки, художник може наблизити криві до певної точки. Ці криві включають сплайни, патчі та NURBS.
- Конвертація - це метод створення полігонів за допомогою моделей поверхні NURBS.
- Окремі полігони можуть бути зроблені шляхом розміщення окремих вершин для демонстрації окремих полігонів. Інструмент "Створити багатокутник" або "Quad Draw Tool" може або витягнути, або розбити вид полігональної сітки.

2.2.2 Retopology

Враховуючи, скільки полігонів має висока деталізована модель (іноді до 10M), часто стає неможливим створювати належні анімації з кількістю точок впливу, з якими має займатися 3D-художник.

Використовуючи функцію Quad Draw в Toolkit моделювання, у вас є доступ до маркування по всій моделі, що дозволяє створювати значно спрощену нову топологію на основі поверхні еталонної моделі.

2.2.3 UV Mapping

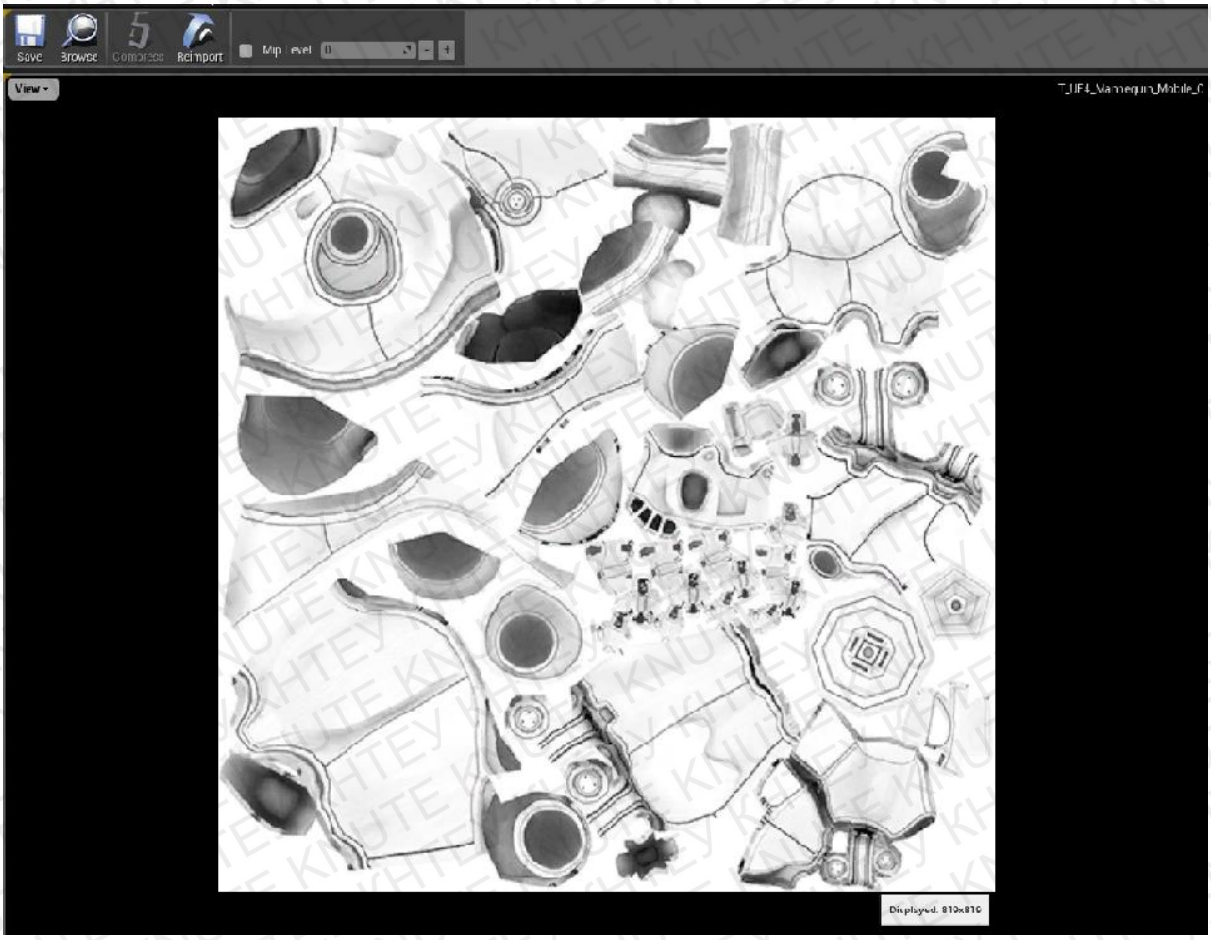

Рис. 2.1 *UV розгортка*

Цей етап означає, що виконавець може зробити 2D-зображення і застосувати його до поверхні 3D-моделі для подальшого відображення текстур.

Розфарбовуючи полігони 3D-моделі кольором та іншими параметрами відображення, UV Mapping призначає пікселі зображення поверхневим відображенням багатокутника шляхом копіювання та вставки фрагмента зображення на об'єкт.

2.2.4 Rigging

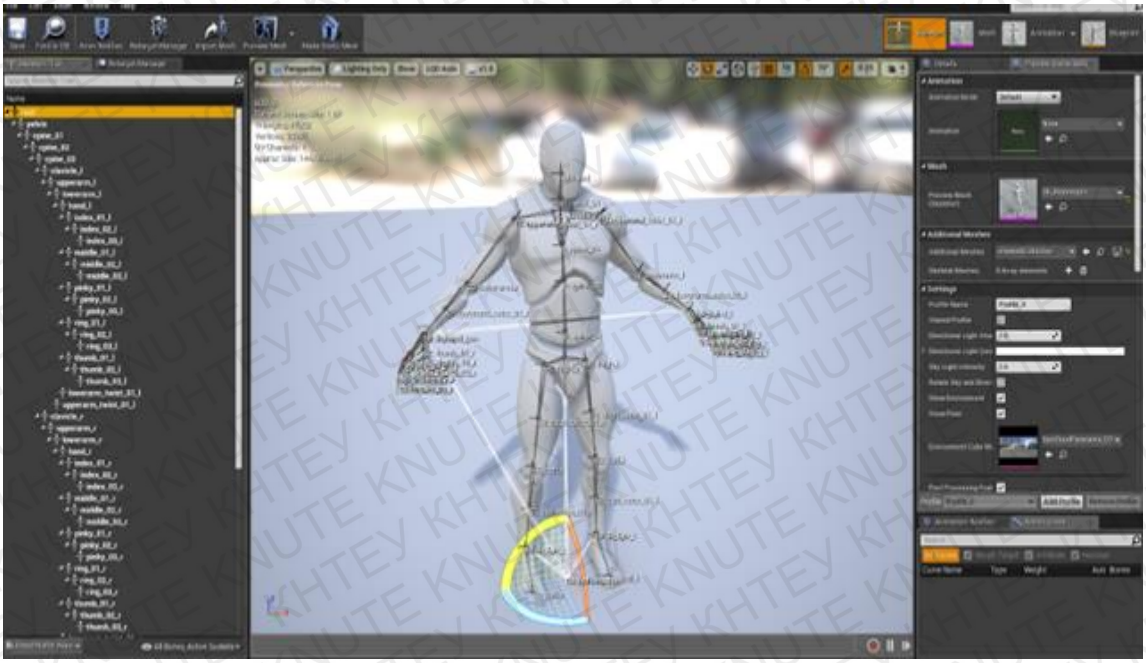

Рис. 2.2 *Всі рухомі частини персонажу*

Rigging - це процедура створення "скелета" і прив'язка його до 3D-сітки. Як скелет в реальному житті, він складається з ряду швів, які можуть маніпулювати моделлю, щоб згинатися або нахилятися в певному напрямку.

2.3 ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ

У другому розділі були описані основні етапи розробки музею, такі як моделювання, конвертація, ретопологія та створення UV розгортки. Також було встановлено завдання та вимоги для створення музею.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ОБ`ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE4

3.1 ВИКОРИСТАНЕ ПЗ

Для реалізації алгоритму було обрано такий набір програмного забезпечення:

Назва	Плюси	Мінуси
Mixcraft 8	• Зрозумілий інтерфейс; • Можливість запис одразу з декілько аудіовходів; • Можливість роботи усіма аудіо форматами; • Висока надійність • Унікальні інструменти.	• За час використанн суттєвих недоліків н було виявлено.
Unreal Engine 4	• Безкоштовний; • Можливість візуальног програмування; • Потужний редакто матеріалів та штучни інтелект; • Фото-реалістична візуалізація.	• Високі системн вимоги.

Таблиця 3.1.

Кінець таблиці 3.1.

3.2 ЕТАПИ СТВОРЕННЯ ОБ*ЄКТІВ VR ЗАСОБАМИ UNREAL ENGINE4

- Створення кімнати;
- Створення базових елементів інтер'єру;
- Моделювання експонатів;
- Взаємодія з тривимірними об'єктами.

Перше, що необхідно зробити при створенні 3Д музею – це створити кімнату в якій в подальшому будуть розміщуватись експонати. Для цього необхідно провести розрахунки розмірів експонатів та самої кімнати.

Для початку необхідно створити підлогу. Для цього необхідно відкрити вкладку «Geometry», в якій треба вибрати фігуру «Box».

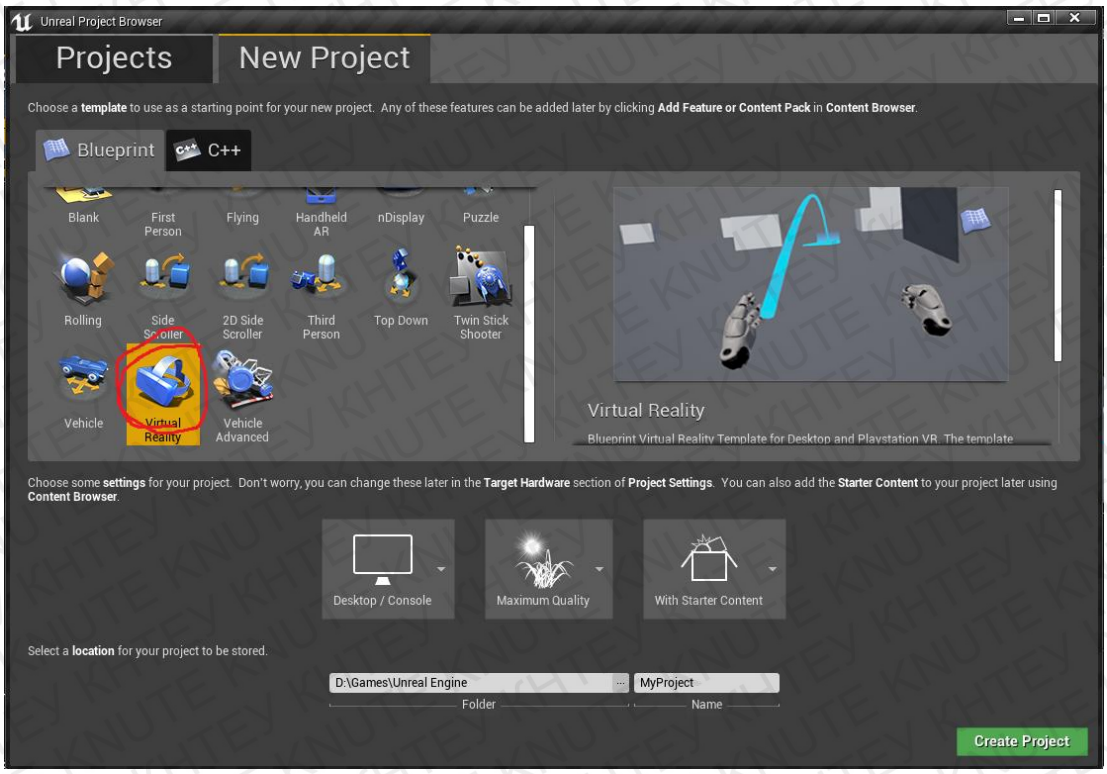

Рис. 3.1. створення проекту віртуальної реальності

Далі необхідно розширити простір. За допомогою вкладки «Geometry Editing» розширюємо куб по осям.

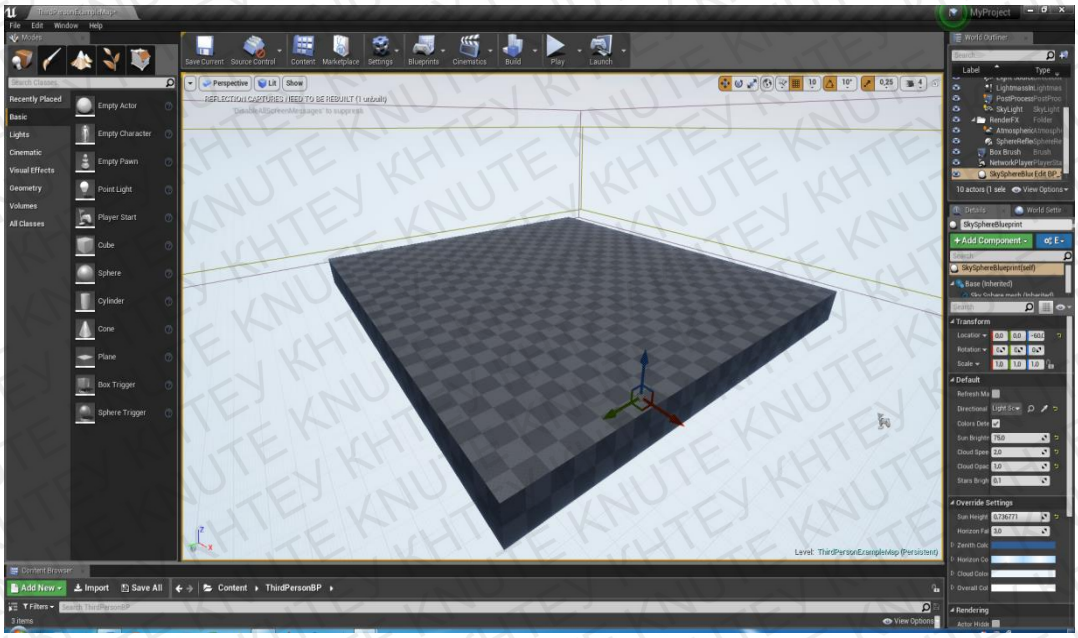

Рис. 3.2. Створення підлоги

Наступний етап у створенні кімнати – це стіни. Для створення стіни необхідно відкрити вкладку «Geometry», в якій треба знову вибрати фігуру «Box».

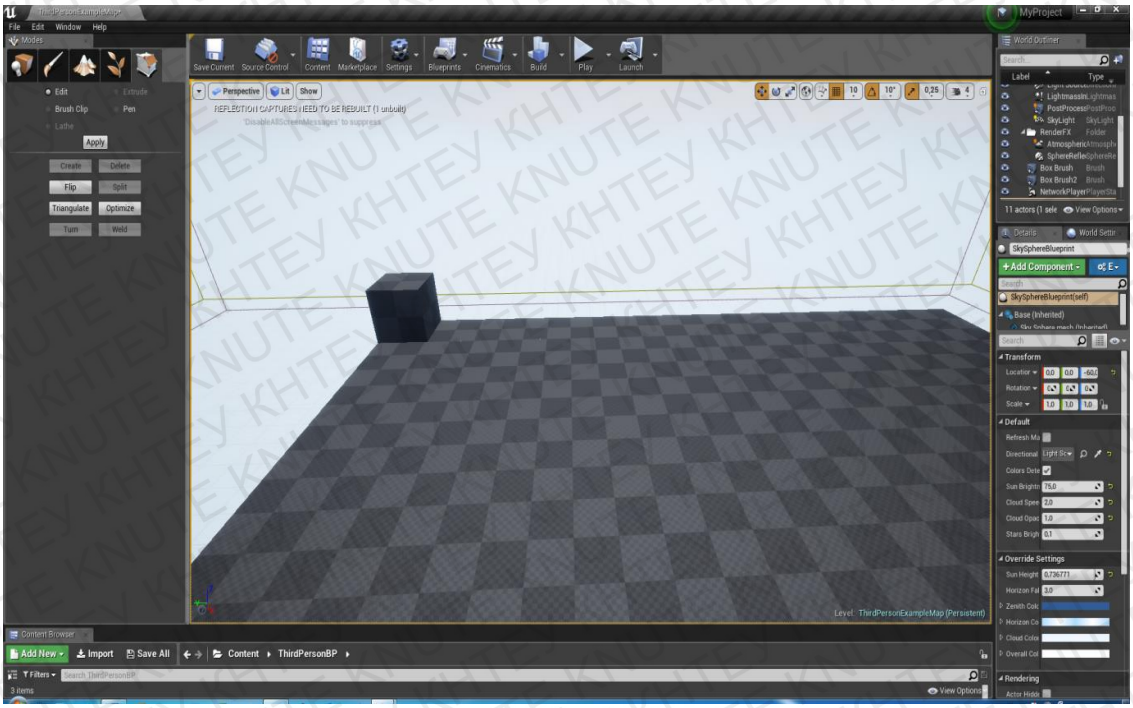

Рис. 3.3. Створення стіни

Далі необхідно розтягнути куб по осям.

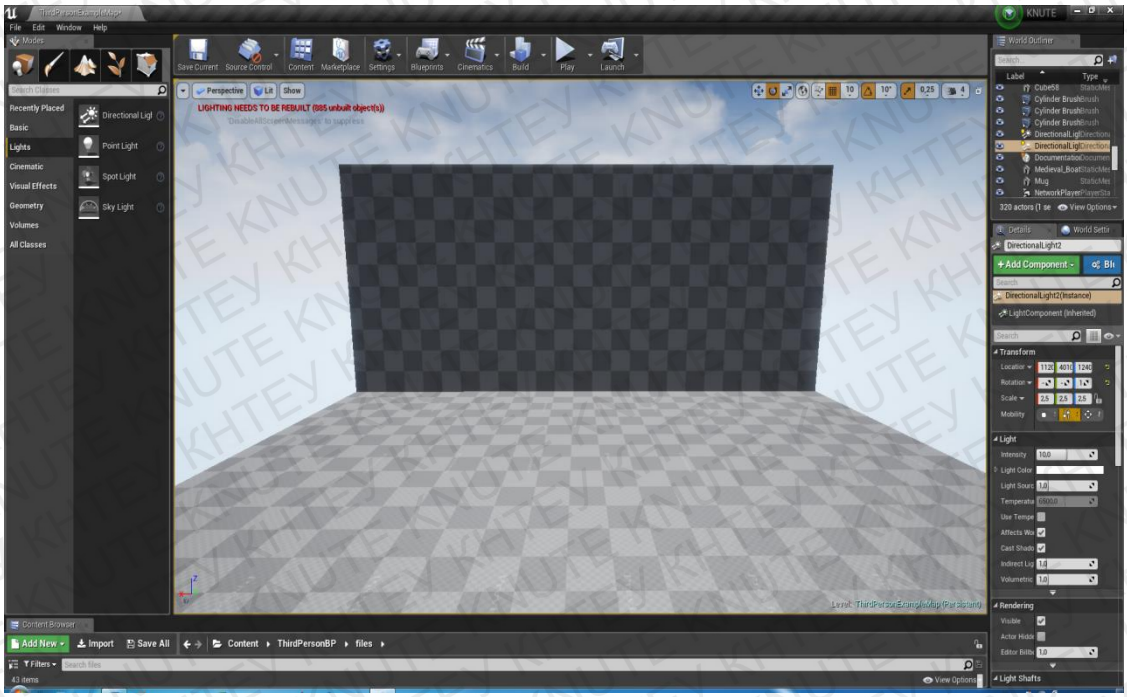

Рис. 3.4. Створення стіни

Копіюємо та розміщуємо стіни так, щоб вийшла коробка.

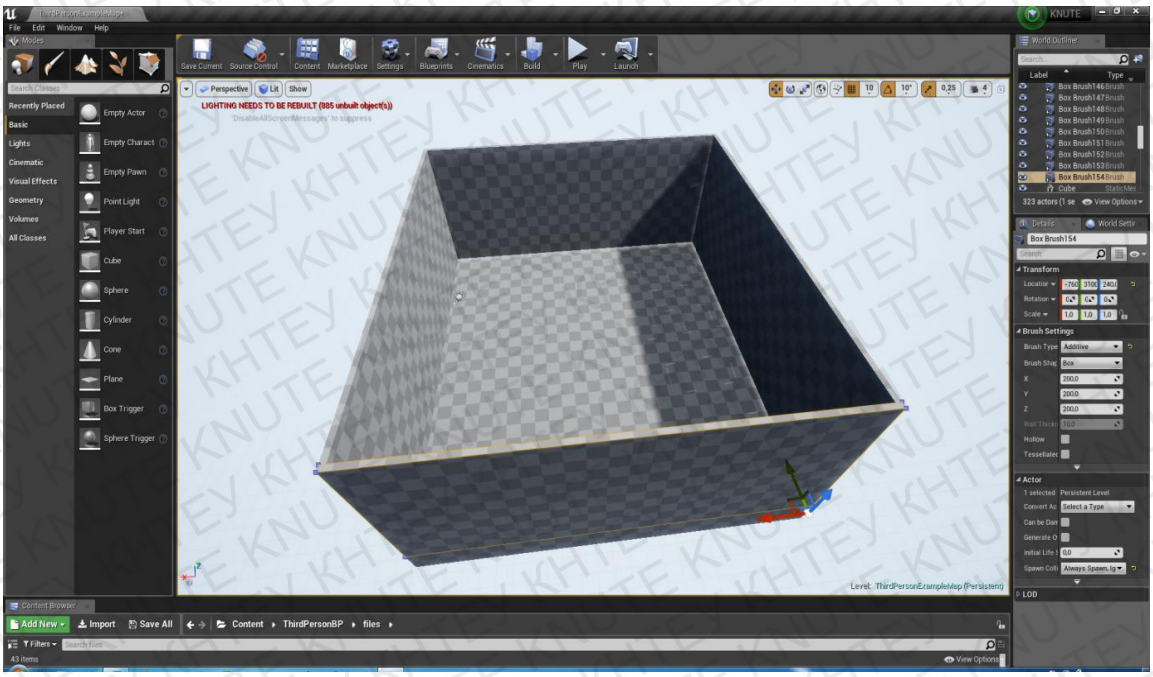

Рис. 3.5. Необроблена кімната

Після виконання цих пунктів можна поступово переходити до базових елементів інтер'єру.

Для підлоги було використано дерев'яну текстуру.

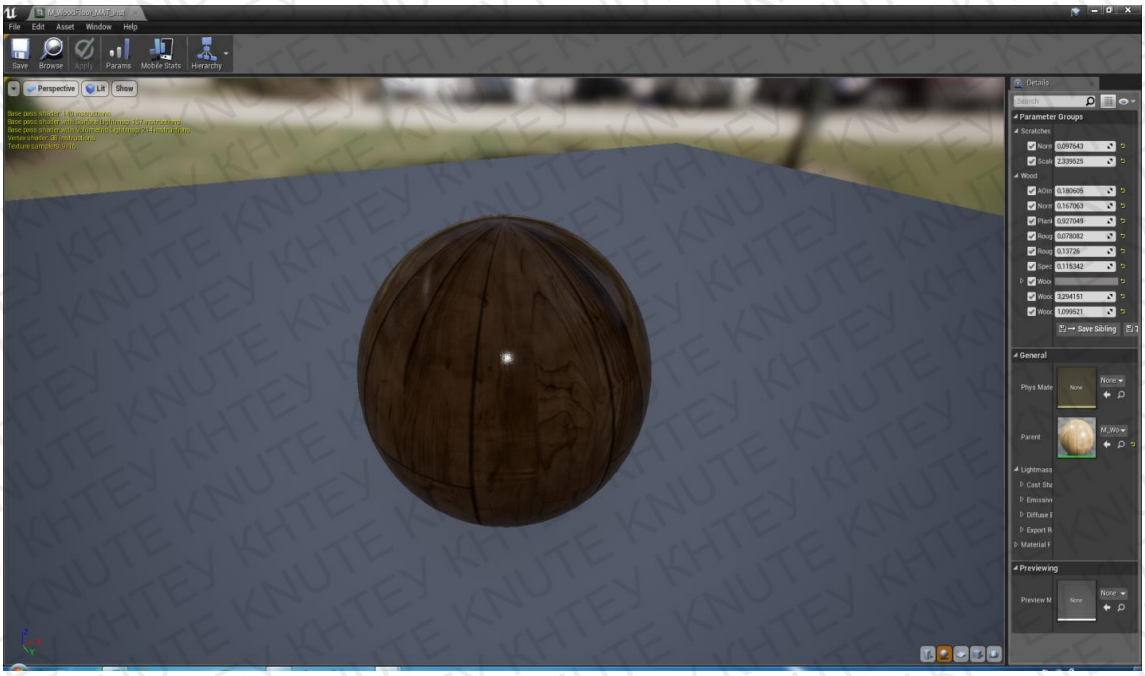

Рис. 3.6. Матеріал підлоги

Так виглядає підлога після покриття дерев'яним матеріалом.

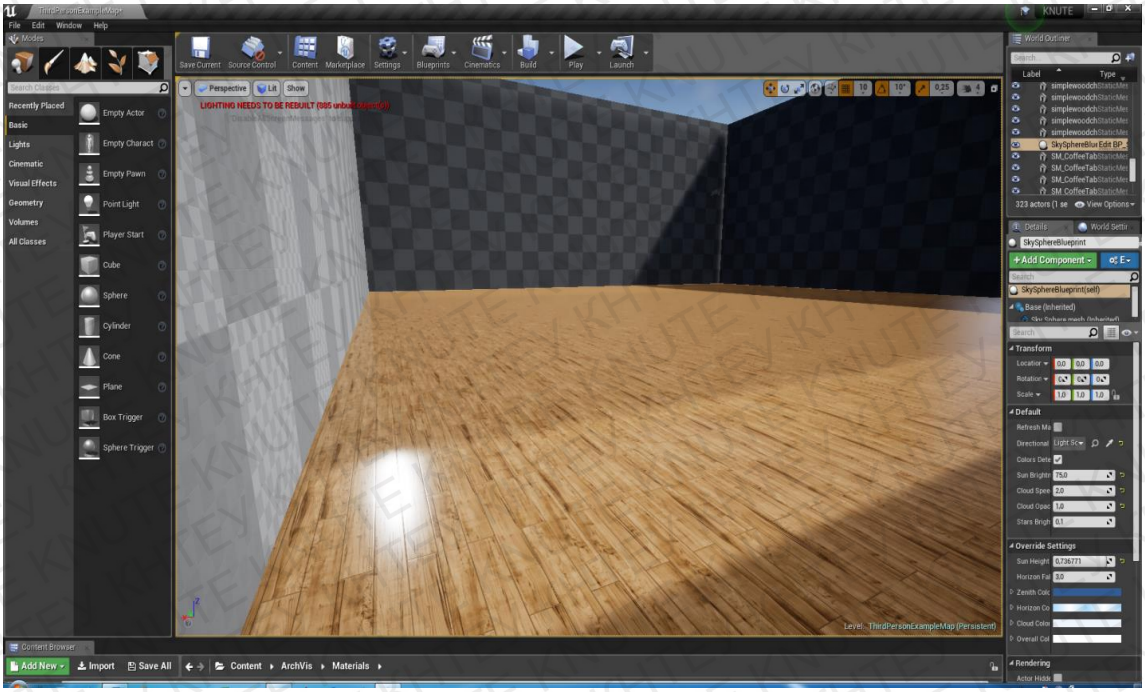

Рис. 3.7. Матеріал підлоги

Далі стіни покриваємо таким самим способом, але іншим матеріалом.

Рис. 3.8. Матеріал стіни

Після облаштування стін та стелі кімната має наступний вигляд.

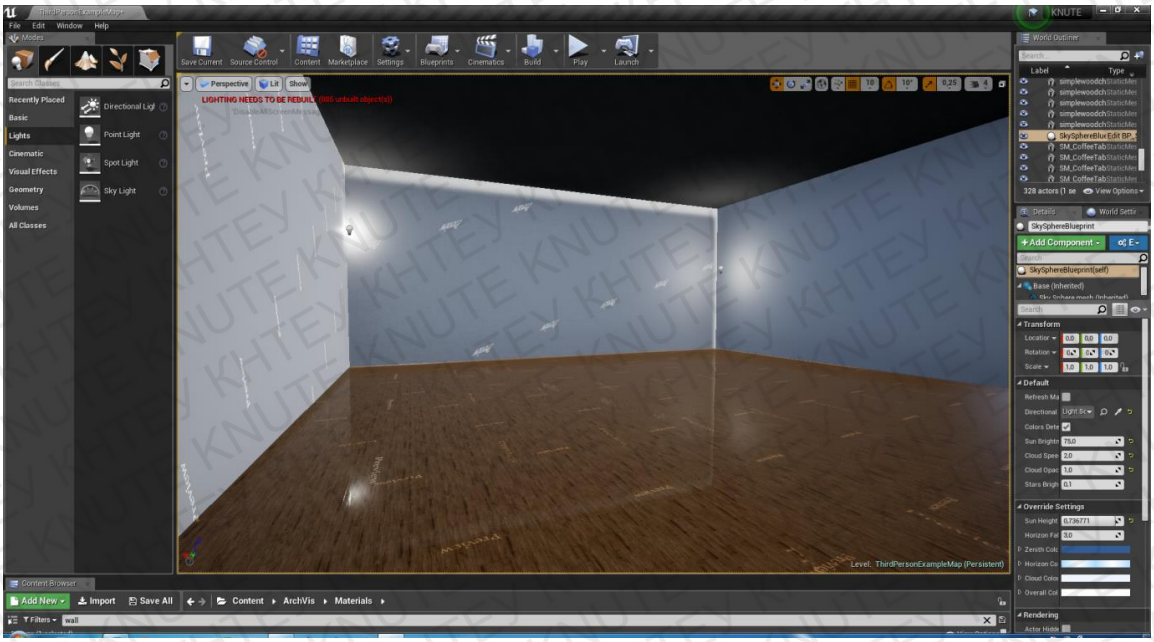

Рис. 3.9. Вигляд кімнати

Для кращого освітлення необхідне потрапляння зовнішніх джерел світла.

Для створення вікон необхідно в одній зі стін проробити отвори. Для цього відкриваємо вкладку «Geometry», де вибираємо фігуру «Box». Далі в

параметрах пензля (*Brush Settings*) вибираємо параметр «subtractive», після чого куб стає невидимим і його необхідно пересунути так, щоб він перетинався з іншим геометричним об'єктом.

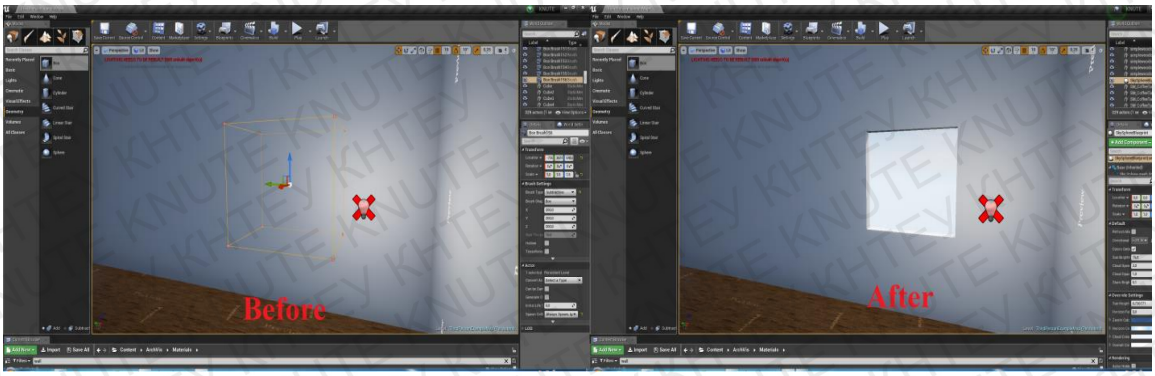

Рис. 3.10. Створення вікна

Наступним кроком просто копіюємо вікна та розміщуємо їх як це потрібно.

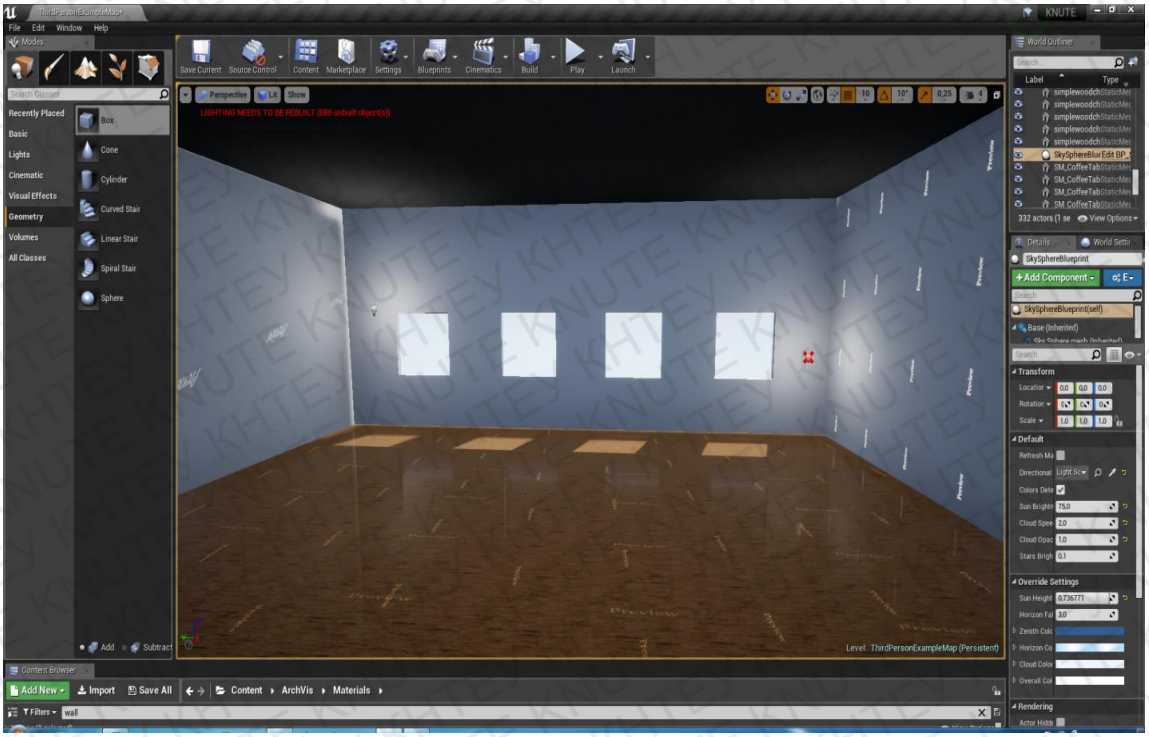

Рис. 3.11. Послідовне розміщення вікон

Отвори готові, але залишати вікна без рами та скла не можна. Для рами використаємо 3D-модель металопластикового вікна.

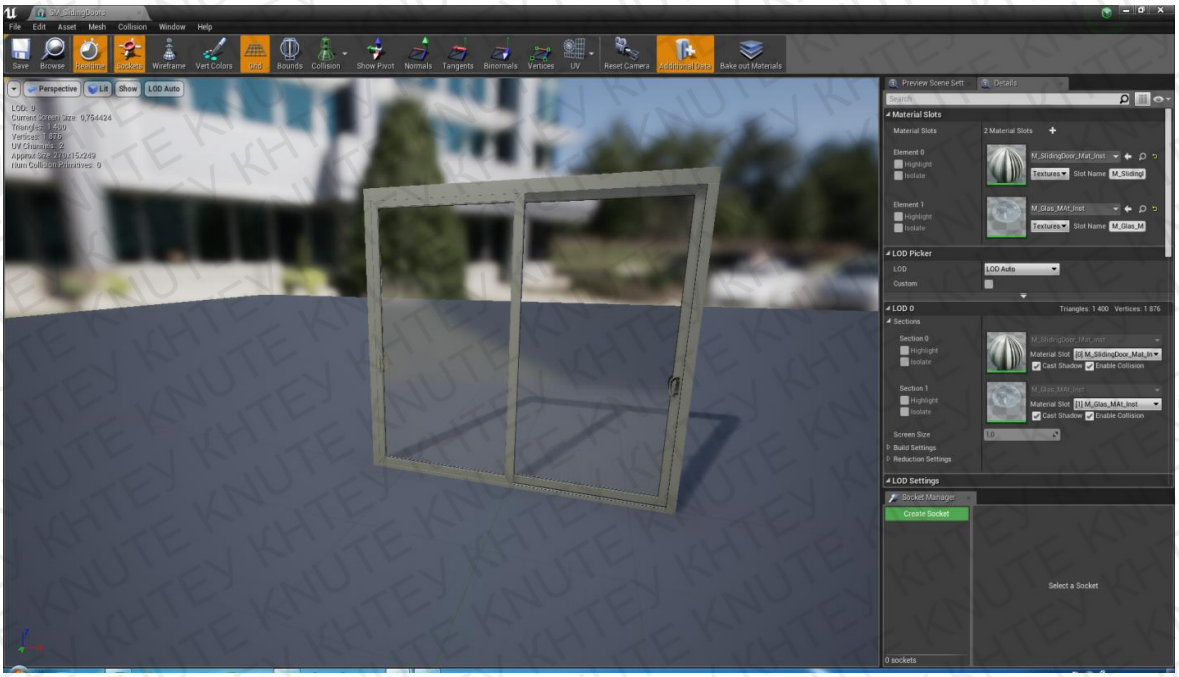

Рис. 3.12. Модель вікна

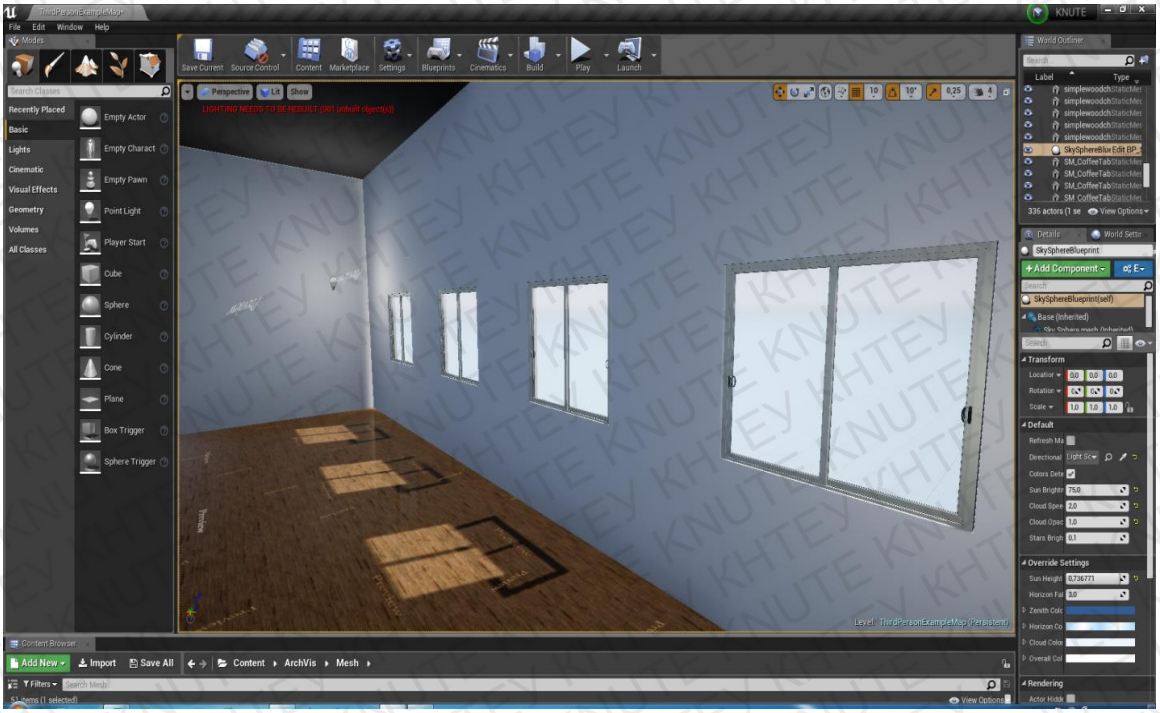

Рис. 3.13. Моделі вікон

Після створення кімнати з базовим інтер'єром можна перейти до її заповнення елементами декору. В музеї КНТЕУ є декілька видів стендів і вітрин. Розглянемо п'ятикутну та прямокутну вітрини які займають своє місце біля вікна.

Рис. 3.14. П'ятикутна та прямокутна вітрини

В арсеналі платформи Unreal Engine 4 немає складних геометричних фігур, таких як додекаедр, пентагондодекаедр або будь-який об'ємний п'ятикутник, тому довелося створити його використовуючи куб.

Створимо паралелепіпед який буде складатися с куба та паралелепіпеда менших розмірів.

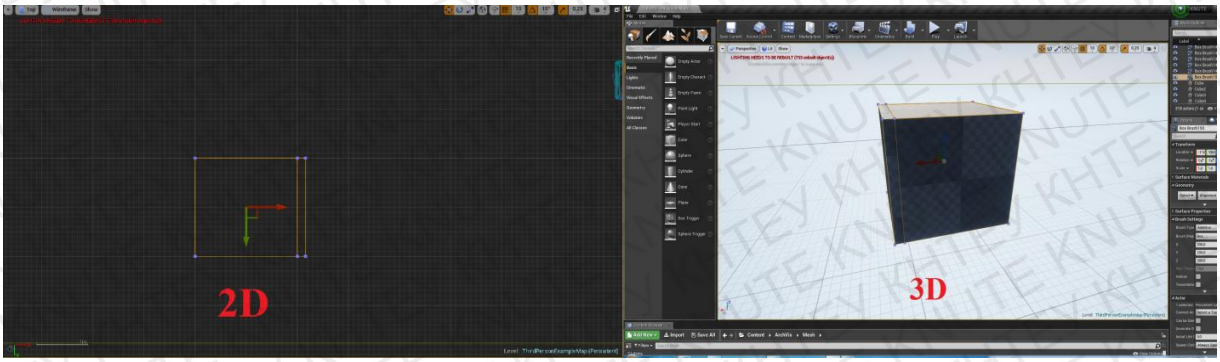

Рис. 3.15. Прямокутний кубоїд

В 2D редакторі за допомогою мишки перетворюємо прямокутник в правильний п'ятикутник – пентагон.

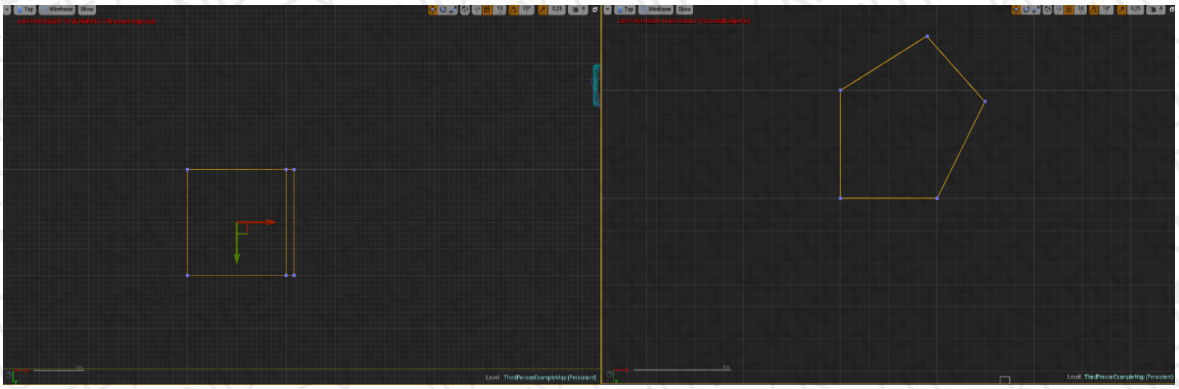

Рис. 3.16. Пентагон

Отримавши п'ятикутник покриваємо його дерев'яним матеріалом.

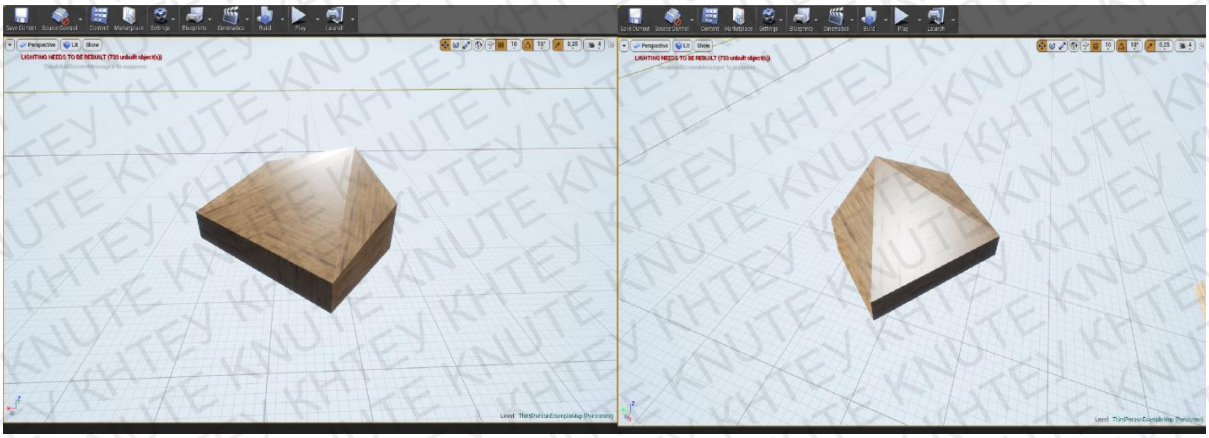

Рис. 3.17. Дерев'яний пентагон

Для того, щоб зробити з даної фігури вітрину, необхідно проробити в ній п'ятикутний отвір. Для цього необхідно копіювати пентагон та за допомогою панелі з редактором розміру зменшити фігуру відповідно до пропорцій які потребуються.

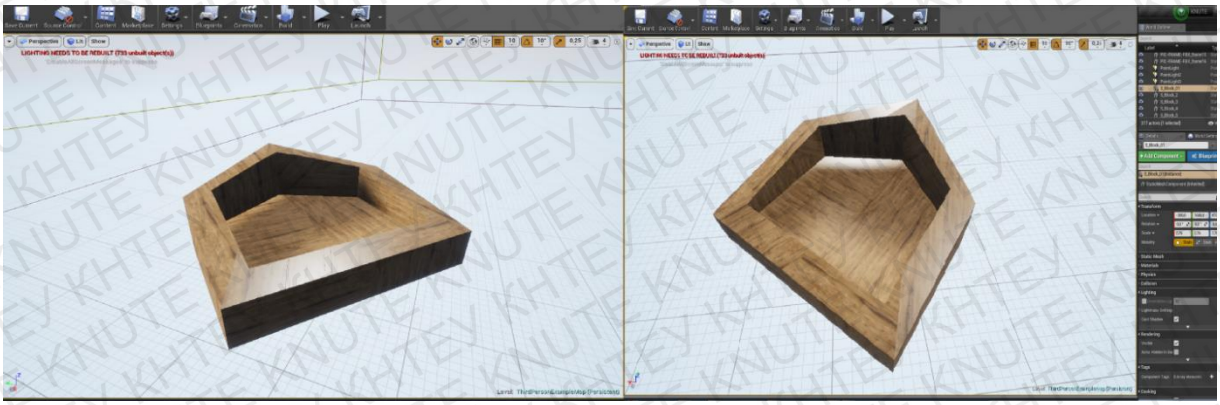

Рис. 3.18. Заготовка вітрини

Щоб закінчити п'ятикутну вітрину – необхідно вкрити її склом та поставити на підставку. Для створення скляної панелі копіюємо пентагон та за допомогою панелі редактора розмірів робим його максимально пласким.

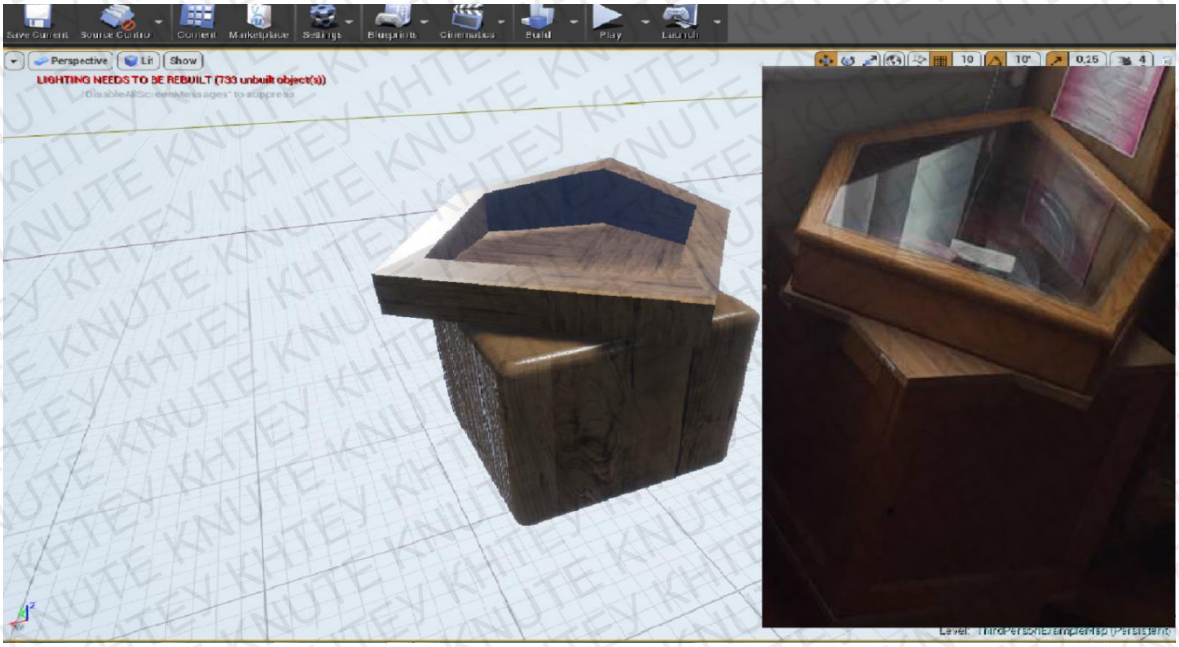

Рис. 3.19. Модель готової вітрини

Дотримуючись даного методу було створено решту елементів віртуального музею.

Наступною дією буде створення рухомого актора та його взаємодія з елементами музею. Базові рухи актора, такі як: рухи вперед, назад, в боки та стрибки генеруються програмою автоматично. Складні анімації рухів та взаємодія з навколишнім середовищем програмується власноруч.

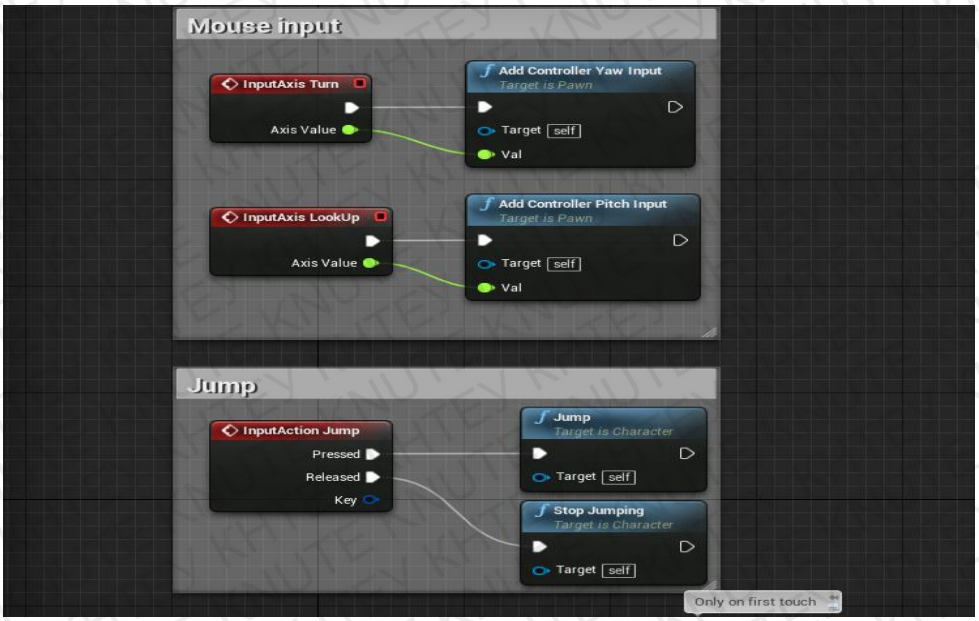

Рис.3.20. Blueprint системи руху актора

Одним з елементів взаємодії між актором та середовищем були двері, що автоматично відкриваються при наближенні актора на певну відстань, анімація для яких створювалась власноруч.

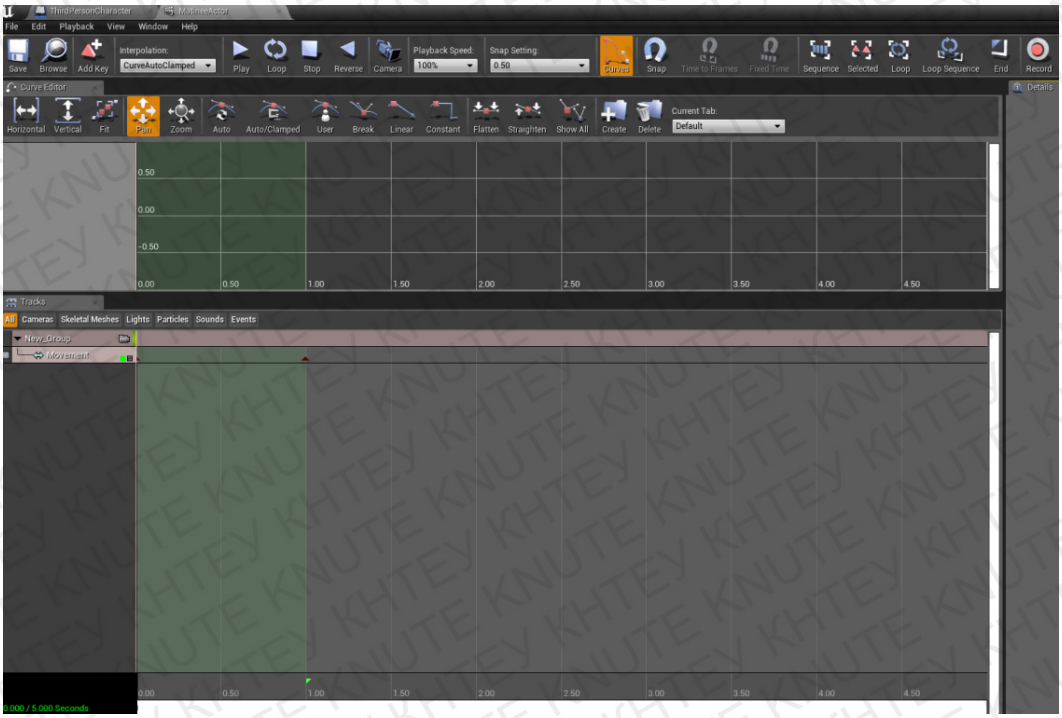

Рис. 3.21. Інтерфейс вікна для створення анімації

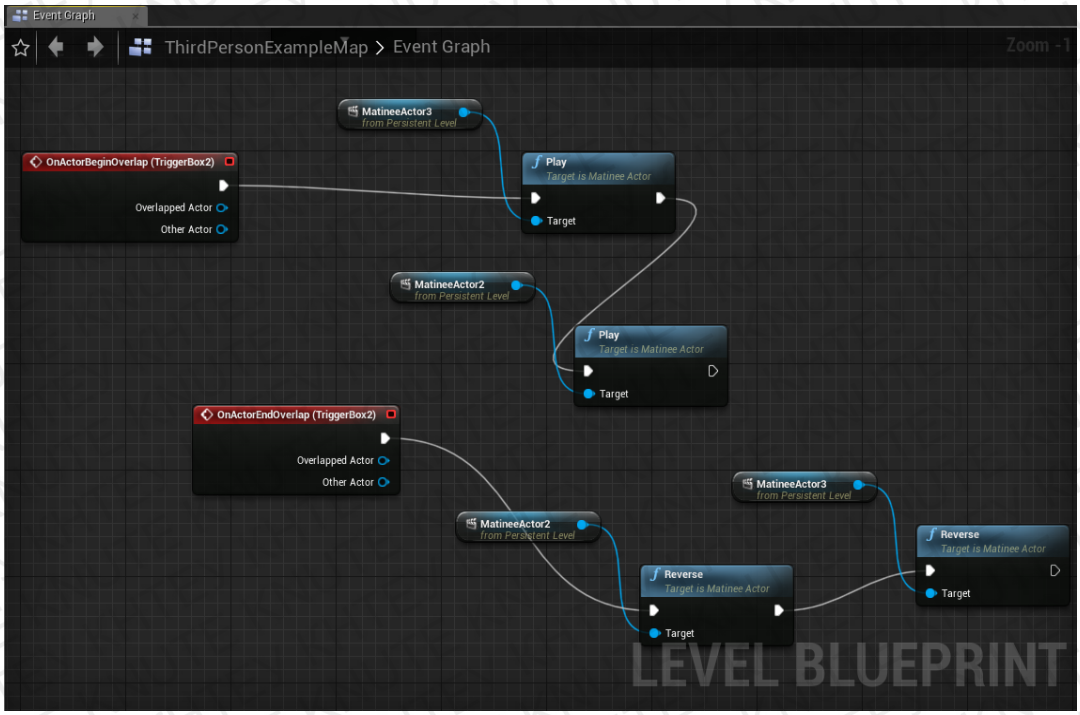

Рис. 3.22. Blueprint для анімації дверей

Принцип дії цієї анімації полягає в тому, що на ігровому рівні розміщується певна область (*Trigger Box*), перетнувши яку гравець надсилає системі сигнал про те, що вданий момент часу необхідно увімкнути анімацію, а коли актор перестає знаходитись в зоні, тоді починає діяти такий Blueprint як *OnActorEndOverlap*, який дає системі знак що необхідно запустити ту ж саму анімацію, але задом наперед.

Наступним прикладом взаємодії є приклад взаємодії актора та експонатів, в даному випадку – картиною. Принцип роботи даного скрипта полягає в тому, що створюється окрема область при перетині якої на екрані з'являється віджет з вказівкою натиснути на вказану клавішу, щоб прослухати інформацію про вказаний експонат. Коли ця клавіша натискається у спеціальній створеній області – то все відбувається за сценарієм і звук відтворюється без помилок, але як тільки актор покине цю зону віджет одразу ж зникає і звук не буде відтворюватися.

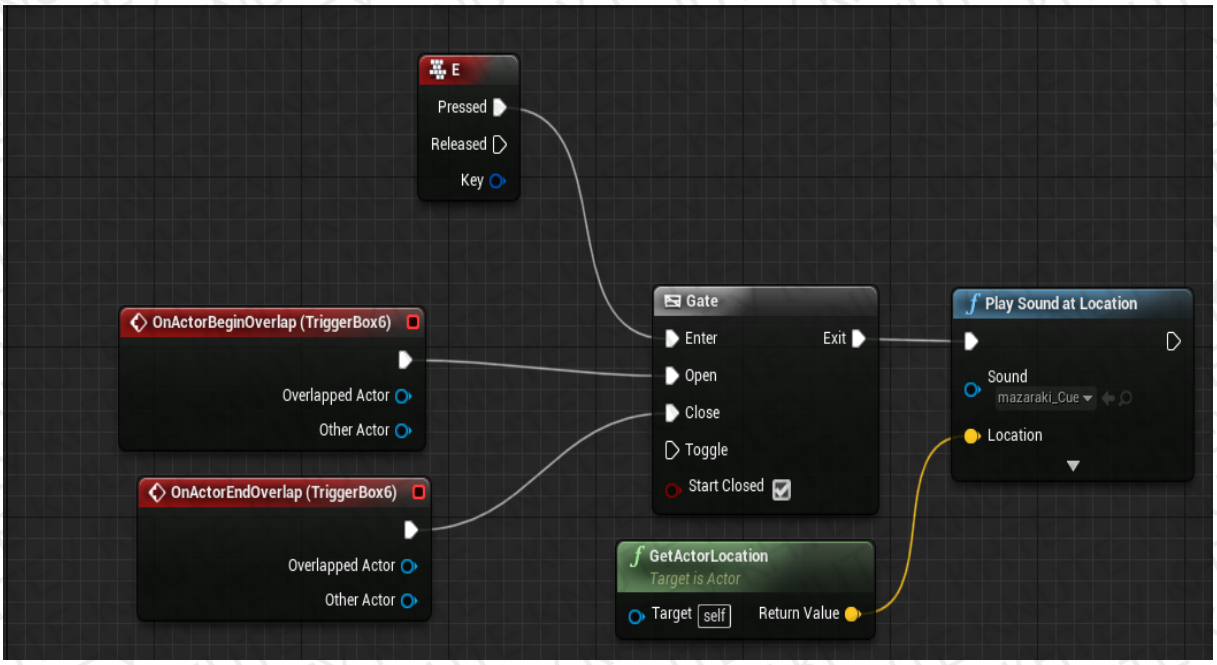

Рис. 3.23. Приклад роботи взаємодії з одним експонатом

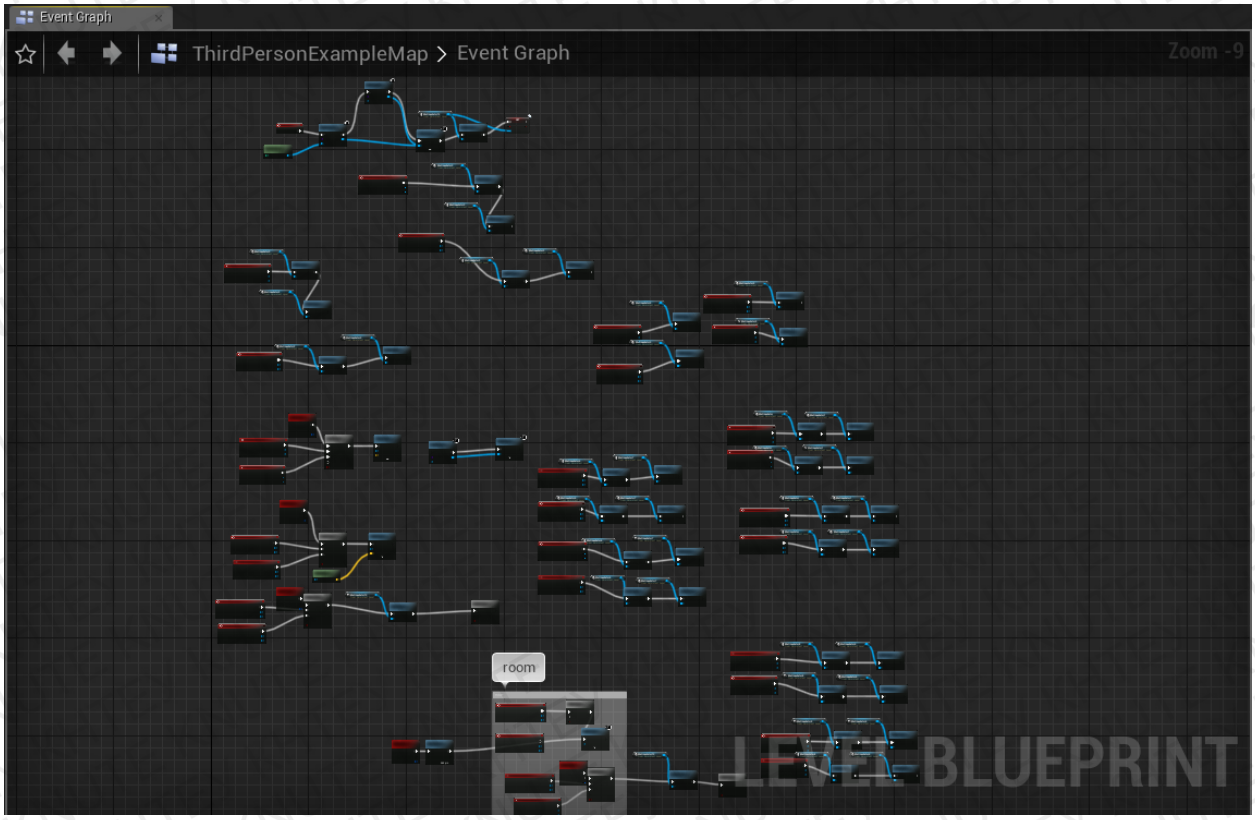

Рис. 3.24. Загальна кількість зв'язків в усій програмі

3.3 ЕКСПОРТ ГОТОВОГО ПРОЕКТУ

Перш ніж експортувати свій проект, вам спочатку потрібно встановити мапу за замовчуванням гри, яка буде завантажуватися, коли починається ваша гра. Якщо ви не встановили карту і використовуєте порожній проект, ви побачите лише чорний екран, коли починається гра.

Щоб встановити мапу за замовчуванням гри, натисніть головне меню Edit> Project Settings> Maps & Modes:

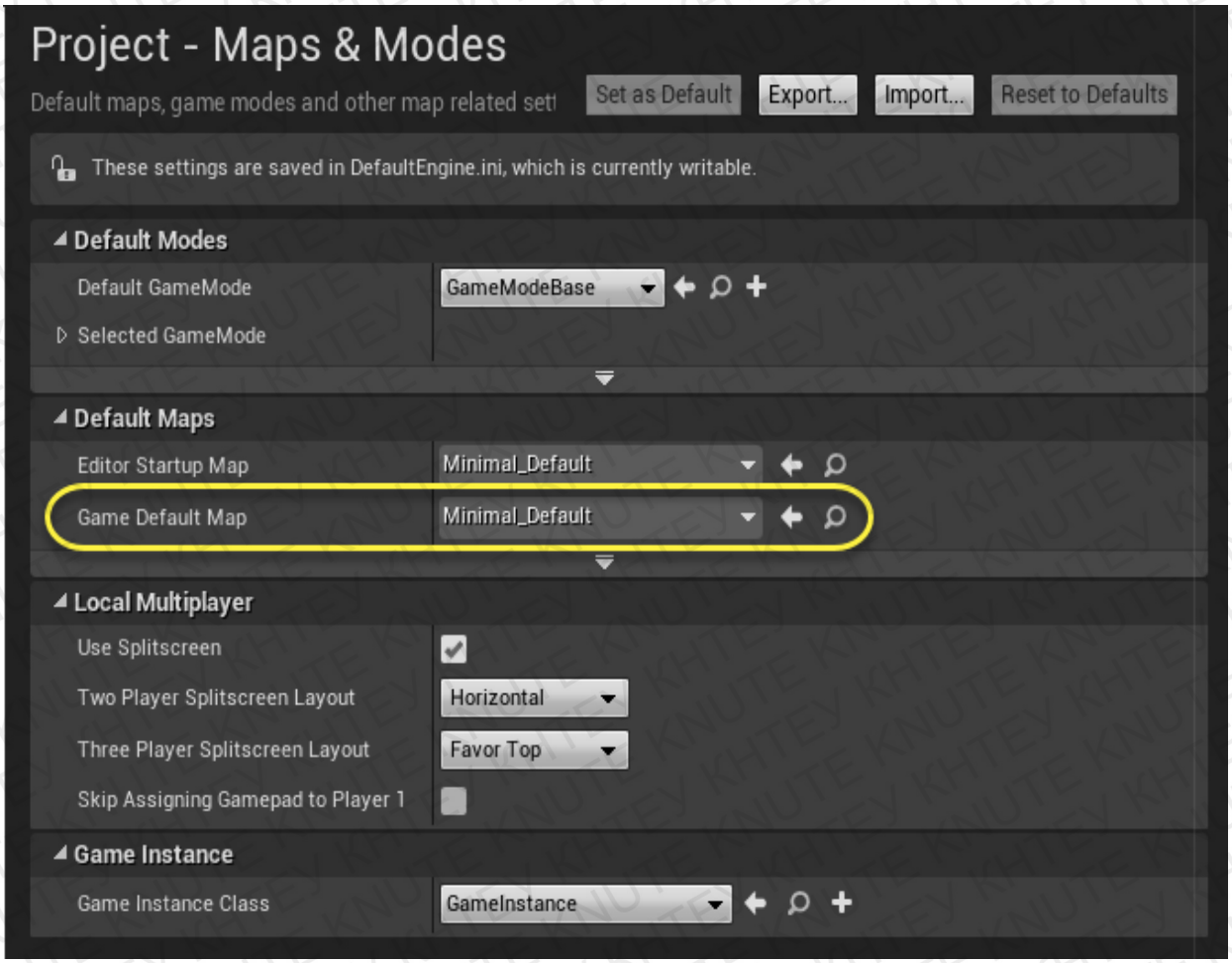

Рис 3.25 *Вікно «Карти та Режими»*

Щоб упакувати проект для певної платформи, у головному меню редактора натисніть File> Package Project> [PlatformName]:

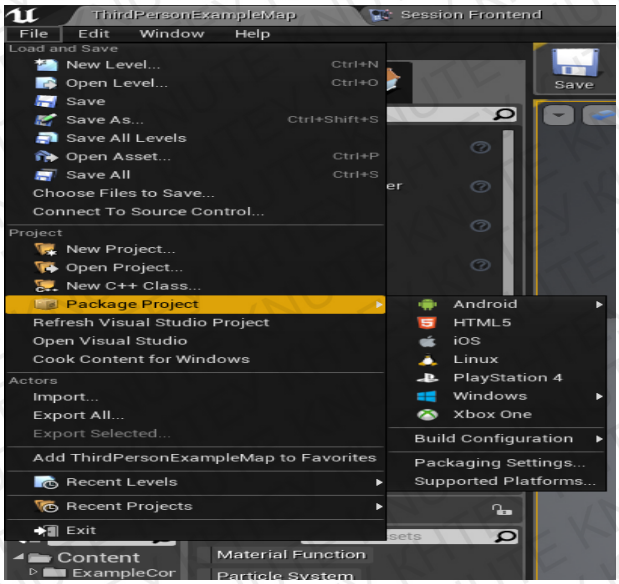

Рис 3.26 *Вибір платформи*

Після підтвердження цільового каталогу буде ініційовано поточний процес, який упаковує проект для вибраної платформи. Оскільки пакування може займати багато часу, цей процес виконується у фоновому режимі, і ви можете продовжувати використовувати редактор. Індикатор стану відображатиметься в нижньому правому куті редактора, щоб вказати прогрес:

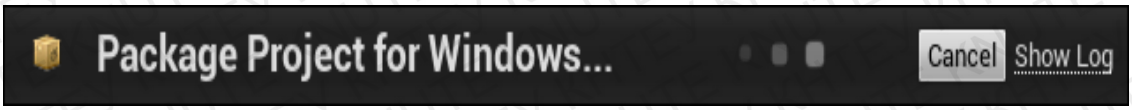

Рис 3.27 *Індикатор стану*

Індикатор стану також надає кнопку Скасувати, щоб зупинити процес упаковки. Крім того, посилання "Показати журнал" може використовуватися для відображення інформації про розширений вихідний журнал, що є корисним для з'ясування того, що пішло не так, якби процес упаковки вийшов з ладу, або для виявлення попереджень, які могли б виявити потенційні помилки в продукті:

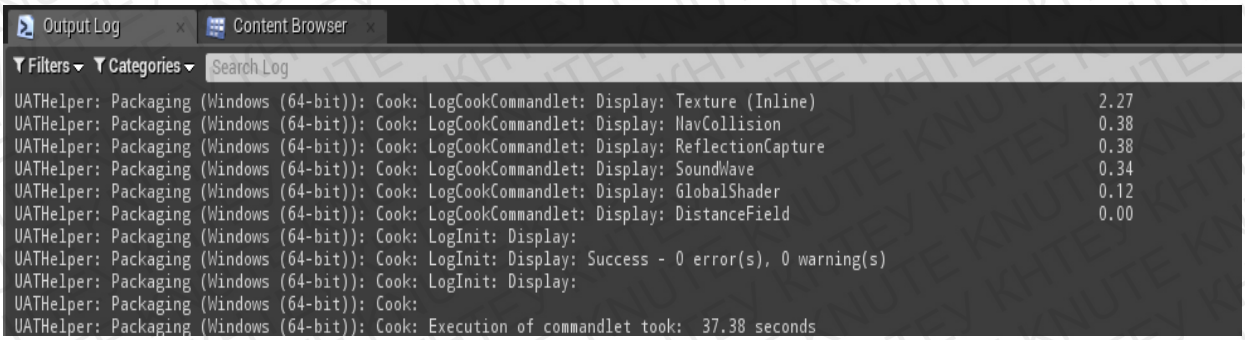

Рис 3.28 *Розширений вихідний журнал*

3.4 ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ

В третьому розділі був описаний повний цикл створення об’єктів віртуальної реальності 3D-музею КНТЕУ, покроково описаний процес експорту проекту та зроблено порівняння плюсів та мінусів обраних програмних засобів. Був реалізований спосіб контролю персонажа використовуючи контролери HTC Vive.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У цьому дипломному проекті була проведена робота з розробки алгоритму та створення музею у віртуальній реальності.

В першому розділі була описана потрібність віртуальної реальності в сучасній сфері IT, проаналізовані програмні засоби та побудована загальна концепція.

У другому розділі були описані основні етапи розробки музею, такі як моделювання, конвертація, ретопологія та створення UV розгортки. Також було встановлено завдання та вимоги для створення об’єктів віртуальної реальності.

У третьому розділі був описаний повний цикл створення інтерактивного VR-музею КНТЕУ, покроково описаний процес експорту проекту та зроблено порівняння плюсів та мінусів обраних програмних засобів.

В рамках проведеного ВКП опрацьовані методи створення VR-образів музейних експонатів з використанням:

- фотограмметрії
- 3D-сканування
- шляхом розробки та реалізації алгоритму створення тривимірної моделі з використанням сучасних графічних програм.

Результати даної роботи можуть бути впроваджені при розробці WEB-сайту віртуального музею КНТЕУ.

Результати цієї роботи пройшли апробацію на Міжнародних науково-практичних конференціях. Робота має номер державної реєстрації 0117U000057.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tom Shannon - Unreal Engine 4 for Design Visualization
   T. Shannon. – Packt Publishing, 2016. — 506p.
2. Matt Edmonds - Mastering Game Development with Unreal Engine 4
   M. Edmonds - Packt Publishing, 2018. — 356p.
3. Satheesh PV - Unreal Engine 4 Game Development Essentials
   Satheesh PV - Packt Publishing, 2016. — 506 p.
4. Benjamin Carnall - Unreal Engine 4.X By Example
   B. Carnall - Packt Publishing, 2016. — 506 p.
5. Joana Lee - Learning Unreal Engine Game Development
   J. Lee - Packt Publishing, 2016. — 274p.
6. Aram Cookson - Unreal Engine 4 Game Development in 24 Hours
   A. Cookson – Pearson Education, 2016. – 465p.
7. Alireza Tavakkoli - Game Development and Simulation with Unreal Technology
   A. Tavakkoli - CRC Press, 2015. — 741 p.
8. Brenden Sewell - Blueprints Visual Scripting for Unreal Engine
   B. Sewell - Packt Publishing, 2015. — 188p.

Інтернет-ресурси

9. Офіційний магазин 3D-моделей компанії Epic Games - https://www.unrealengine.com/marketplace/en-US/store
10. Офіційний сайт Unreal Engine - https://www.unrealengine.com/en-US/feed
11. Візуальне програмування - https://en.wikipedia.org/wiki/Visual_programming_language

ТЕХНІЧНЕ ЗАВДАННЯ

Однією з найголовніших запорук відповідності вимогам та високої якості розробленого програмного рішення є грамотно спроектоване до початку розробки технічне завдання. Отже, необхідно розкрити невід'ємні положення технічного завдання до розробки об'єктів Virtual Reality (VR) засобами Unreal Engine4.

1. Загальна інформація

Назва проекту: KNUTE museum.

Планові строки розробки: грудень 2020 – вересень 2020.

Призначення програмного рішення: віртуалізація музейного простору КНТЕУ.

Мета створення програмного рішення: опанування сучасних технологій розробки VR-додатків та їх закріплення на практиці.

2. Функціональні вимоги до додатку

До функціональних можливостей повинні входити – керування персонажем через клавіатуру та контролери віртуальної реальності, взаємодія з віртуальним середовищем.

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Unreal Engine містить вбудований набір інструментів для тестування і налагодження вашого проекту. Ви можете запустити програму прямо у вікні редактора, в реальному часі спостерігаючи за тим, як змінюються об'єкти, які виникають події і як протікає виконання заданих сценаріїв.

LogD3D11RHI: Driver Date: 7-3-2019

LogRHI: Texture pool is 1379 MB (70% of 1970 MB)

LogD3D11RHI: Async texture creation enabled

LogD3D11RHI: GPU Timing Frequency: 1000.000000 (Debug: 2 2)

LogTemp: Display: Module 'AllDesktopTargetPlatform' loaded TargetPlatform 'AllDesktop'

LogTemp: Display: Module 'MacClientTargetPlatform' loaded TargetPlatform 'MacClient'

LogTemp: Display: Module 'MacNoEditorTargetPlatform' loaded TargetPlatform 'MacNoEditor'

LogTemp: Display: Module 'MacServerTargetPlatform' loaded TargetPlatform 'MacServer'

LogTemp: Display: Module 'MacTargetPlatform' loaded TargetPlatform 'Mac'

LogTemp: Display: Module 'WindowsClientTargetPlatform' loaded TargetPlatform 'WindowsClient'

LogTemp: Display: Module 'WindowsServerTargetPlatform' loaded TargetPlatform 'WindowsServer'

LogTemp: Display: Module 'WindowsTargetPlatform' loaded TargetPlatform 'Windows'

ДОДАТКИ

Додаток А

Лістинг програмного коду для додавання скелетної сітки до контролерів руху

```
USkeletalMeshComponent* AcVRPlayerPawn::CreateHandMesh(UMotionControllerComponent*
a_compParent, FName a_strDisplayName, FName a_nameHandType)
{
USkeletalMeshComponent* refComponentHand = NULL;

//Find the default cube that ships with the engine content
static
ConstructorHelpers::FObjectFinderHandMeshObject(TEXT("SkeletalMesh'/Game/VirtualReality/Mannequi
n/Character/Mesh/MannequinHand_Right.MannequinHand_Right'"));
if (!HandMeshObject.Object)
{
UE_LOG(LogTemp, Error, TEXT("Could not load the default cube for hand mesh"));
return NULL;
}

//create the mesh component
refComponentHand = CreateDefaultSubobject(a_strDisplayName);
//set the mesh to the component
refComponentHand->SetSkeletalMesh(HandMeshObject.Object,true);

//Set the defaults
refComponentHand->SetAutoActivate(true);
refComponentHand->SetVisibility(true);
refComponentHand->SetHiddenInGame(false);
FQuat qRotation = FQuat::Identity;
FVector vec3Scale = FVector::OneVector;
if (a_nameHandType == FXRMotionControllerBase::LeftHandSourceId)
{
qRotation = FQuat(FVector(1, 0, 0), FMath::DegreesToRadians(90));
vec3Scale = FVector(1, -1, 1);
//a_refHand->SetVisibility(false);
//a_refHand->SetHiddenInGame(true);
}
else
{
qRotation = FQuat(FVector(1, 0, 0), FMath::DegreesToRadians(270));
}
refComponentHand->SetRelativeLocationAndRotation(FVector::ZeroVector, qRotation);
refComponentHand->SetRelativeScale3D(vec3Scale);
refComponentHand->SetupAttachment(a_compParent);

return refComponentHand;
}
```