

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики  
Кафедра інженерії програмного забезпечення в енергетиці

ДО ЗАХИСТУ ДОПУЩЕНО

В.о. завідувача кафедри

Олександр КОВАЛЬ

«\_\_» \_\_\_\_\_ 2025 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інженерія програмного забезпечення  
інтелектуальних кібер-фізичних систем в енергетиці»  
спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Служба виявлення аномалій у моделях даних часових рядів»**

Виконав:

студент IV курсу, групи ТВ-11

Каленський Олександр Миколайович

(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Керівник:

доцент, к.т.н., Залевська Ольга Валеріївна

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Консультант з віртуальної та доданої реальності

доцент, к.т.н., Залевська Ольга Валеріївна

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Рецензент:

\_\_\_\_\_ (посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій дипломній роботі  
немає запозичень із праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2025

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Навчально-науковий інститут атомної та теплової енергетики  
Кафедра інженерії програмного забезпечення в енергетиці  
Рівень вищої освіти перший (бакалаврський)  
Спеціальність 121 Інженерія програмного забезпечення  
Освітньо-професійна програма «Інженерія програмного забезпечення  
інтелектуальних кібер-фізичних систем в енергетиці»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ Олександр КОВАЛЬ

«\_\_» \_\_\_\_\_ 2025р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

\_\_\_\_\_ Каленському Олександру Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Віртуальний світ IT-компанії в енергетиці»

керівник роботи Залевська Ольга Валеріївна, доцент, к.т.н.

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

затверджені наказом по університету від «\_\_» \_\_\_\_\_ 2025р. № \_\_\_\_\_

2. Строк подання студентом роботи «\_\_» \_\_\_\_\_ 2025р.

3. Вихідні дані до роботи: ігровий рушій Unreal Engine 5, програмний пакет для  
створення 3D-об'єктів Blender, мова програмування C++

4. Зміст (дипломної роботи) пояснювальної записки (перелік завдань, які потрібно розробити): створити інтерактивні 3D-моделі офісів та виробничих приміщень IT-  
компанії. Розробити систему взаємодії користувачів з віртуальними об'єктами  
компанії.

5. Перелік ілюстративного матеріалу: приклад інтерфейсу користувача. Фото  
реального офісу та побудованої 3D-моделі

6. Дата видачі завдання «31» жовтня 2024р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Строки виконання етапів роботи	Примітка
1	Отримання завдання	30.10.2024	виконано
2	Дослідження предметної області	31.10.2024 – 01.11.2024	виконано
3	Дослідження існуючих рішень	02.11.2024 – 01.12.2024	виконано
4	Постановка вимог до проектування системи	02.12.2024 – 12.01.2025	виконано
5	Розробка програмного продукту	13.01.2025 – 11.05.2025	виконано
6	Тестування	12.05.2025 – 15.05.2025	виконано
7	Захист програмного продукту	12.05.2025 – 15.05.2025	виконано
8	Оформлення дипломної роботи	19.05.2025 – 01.06.2025	виконано
9	Передзахист	02.06.2025 – 06.06.2025	виконано
10	Захист	16.06.2025 – 27.06.2025	виконано

Студент

---

(підпис)

Олександр Каленський

---

(ім'я, прізвище)

Керівник роботи

---

(підпис)

Ольга Залевська

---

(ім'я, прізвище)

## РЕФЕРАТ

**Структура та обсяг дипломної роботи.** Робота містить 69 сторінок, 6 рисунків, 3 додатки та 9 посилань.

**Метою роботи** є створити інтерактивну 3D-модель офісу IT-компанії, що працює в галузі енергетики, з можливістю перегляду та взаємодії у віртуальній реальності. Реалізація такого середовища дозволяє дослідити застосування сучасних цифрових технологій, зокрема Unreal Engine та Blender, у створенні віртуальних просторів, а також на практиці опрацювати принципи 3D-моделювання, побудови взаємодії з об'єктами та візуалізації офісного середовища у VR-форматі.

**Практичне значення одержаних результатів** полягає у створенні інтерактивного віртуального офісу IT-компанії, що може бути використаний для демонстрацій, навчання, ознайомлення нових співробітників із робочим середовищем або презентацій клієнтам. Результати роботи підтверджують ефективність використання сучасних інструментів, таких як Unreal Engine і Blender, для побудови віртуального простору з елементами взаємодії. Такий підхід може бути адаптований для інших галузей, де потрібна візуалізація офісних або виробничих середовищ у VR-форматі.

**Ключові слова:** віртуальна реальність, VR, Unreal Engine 5, Blender, 3D-моделювання, інтерактивне середовище, офіс IT-компанії, енергетика, візуалізація, користувацька взаємодія.

# ABSTRACT

**Structure and Volume of the Thesis.** The thesis consists of 69 pages, 6 figures, 3 appendices, and 9 references.

**The goal of the work** is to create an interactive 3D model of an IT company's office operating in the energy sector, with the ability to view and interact with it in virtual reality. The implementation of such an environment makes it possible to explore the application of modern digital technologies, such as Unreal Engine and Blender, in building virtual spaces, as well as to practice principles of 3D modeling, interaction logic, and office visualization in a VR format.

**The practical significance of the obtained results** lies in the development of an interactive virtual office of an IT company, which can be used for demonstrations, training, onboarding of new employees, or client presentations. The results confirm the effectiveness of using modern tools like Unreal Engine and Blender for building interactive virtual environments. This approach can also be adapted for other industries that require visualization of office or production spaces in a VR format.

**Keywords:** virtual reality, VR, Unreal Engine 5, Blender, 3D modeling, interactive environment, IT company office, energy sector, visualization, user interaction.

# ЗМІСТ

ВСТУП.....	8
1 ЗАДАЧА ПОБУДОВА РЕАЛІСТИЧНОЇ МОДЕЛІ ОФІСУ ІТ-КОМПАНІЇ В ЕНЕРГЕТИЦІ У ВІРТУАЛЬНІЙ РЕАЛЬНОСТІ.....	10
1.1 Постановка задачі.....	10
1.2 Вибір інструментів і середовища розробки.....	11
1.2.1 Вибір ігрового рушія для віртуального середовища.....	11
1.2.2 Вибір засобів 3D-моделювання.....	13
1.2.3 Висновок: обґрунтування вибору Unreal Engine та Blender.....	15
1.3 Висновок до Розділу 1.....	16
2 АНАЛІЗ ІНСТРУМЕНТІВ РОЗРОБКИ ТА РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	18
2.1 Вибір платформи розробки та мови програмування.....	18
2.2 Мова програмування C++.....	20
2.3 Програмний пакет для створення тривимірної комп'ютерної графіки Blender.....	24
2.4 Налаштування та генерація колізій об'єктів в Unreal Engine 5.....	28
2.4.1 Ручне налаштування колізій.....	28
2.4.2 Канали та реакції колізій.....	28
2.4.3 Прості vs. складні колізії.....	30
2.4.4 Інструмент Auto Convex Collision.....	32
2.4.5 Типові проблеми з колізіями у VR-проектах.....	33
2.4.6 Вирішення проблем у VR через налаштування колізій.....	34
2.5 Висновок до Розділу 2.....	37
3 ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	38
3.1 Проектування архітектури віртуального офісу.....	38
3.2 Створення 3D-моделей та текстурування.....	40
3.3 Розробка інтерактивних елементів.....	42
3.3.1 Взаємодія з об'єктами (підбір предметів).....	42
3.3.2 Тестування взаємодії користувача з віртуальним середовищем.....	44
3.3.3 Зміна колізій об'єктів через некоректну роботу.....	45

3.3.4 Створення VRrawn (модель користувача, переміщення та меню).....	46
3.4 Висновок до Розділу 3.....	46
4 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ .....	48
4.1 Використання в режимі VR.....	48
4.2 Класичний режим 3D-переглядання .....	50
4.3 Висновок до Розділу 4 .....	51
ВИСНОВКИ .....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	53
ДОДАТОК А .....	55
ДОДАТОК Б.....	58
ДОДАТОК В .....	64

## ВСТУП

Сучасний розвиток технологій віртуальної реальності (VR) відкриває нові можливості для різних сфер діяльності, включаючи енергетику та інформаційні технології. Віртуальні середовища дозволяють відтворювати складні просторові моделі, що можуть використовуватися для тренувань, симуляцій, презентацій та віртуальних турів. Одним із перспективних напрямів є створення віртуальних офісів, що дає змогу взаємодіяти з об'єктами, переміщатися в просторі та відтворювати реалістичні умови роботи в цифровому форматі.

Метою даної роботи є розробка віртуальної моделі офісу ІТ-компанії в енергетичній сфері, яка дозволить користувачам повністю зануритися у віртуальне середовище, досліджувати його, взаємодіяти з предметами та використовувати інструменти управління за допомогою VR-технологій. Такий підхід надає можливість не тільки відчувати присутність у цифровому просторі, але й оптимізувати процеси роботи та навчання, зменшити витрати на фізичні приміщення та підвищити рівень безпеки при навчальних симуляціях.

Під час розробки віртуального офісу були використані сучасні технології, зокрема рушій Unreal Engine 5.5.4 для створення інтерактивного середовища та Blender для моделювання тривимірних об'єктів. Це дозволило відтворити реалістичну графіку, забезпечити фізично коректну взаємодію з об'єктами та інтегрувати віртуальну реальність у фінальну модель. Віртуальний офіс підтримує два режими роботи: VR-режим із повним зануренням та 3D-проглядання на екрані монітора, що розширює можливості його використання.

Розроблений проєкт включає інтерактивні елементи, такі як підбір предметів, їх переміщення, взаємодія з панелями управління, а також підтримку реалістичної фізики. Для зручної навігації у VR було створено модель VRpawn, яка відповідає за переміщення користувача, відображення віртуальних рук та інтерактивне меню.

У даній роботі детально описано процес розробки віртуального офісу: від проектування архітектури до тестування інтерактивних елементів і оптимізації під



VR-пристрої. Також розглянуто методи моделювання в Blender, імпорт об'єктів в Unreal Engine, налаштування фізики та інтеграцію з віртуальною реальністю. Проведене тестування забезпечило стабільність роботи додатку та коректність взаємодії користувача з об'єктами віртуального середовища.

Результатом роботи стала інтерактивна віртуальна модель офісу, що дозволяє ефективно взаємодіяти з цифровим простором, забезпечуючи користувачу повне занурення та реалістичну взаємодію з предметами. Це відкриває нові можливості для використання VR-технологій в енергетичній сфері та ІТ-компаніях, сприяючи розвитку віртуальних робочих місць та цифрових симуляцій.

# **1 ЗАДАЧА ПОБУДОВА РЕАЛІСТИЧНОЇ МОДЕЛІ ОФІСУ ІТ-КОМПАНІЇ В ЕНЕРГЕТИЦІ У ВІРТУАЛЬНІЙ РЕАЛЬНОСТІ**

## **1.1 Постановка задачі**

Метою даної задачі є створення віртуальної моделі офісу ІТ-компанії, що працює в сфері енергетики. Така модель повинна виглядати максимально реалістично та давати змогу користувачу повноцінно зануритися у віртуальний простір за допомогою VR-технологій. Офіс має бути не просто «3D-приміщенням», а відображати справжнє робоче середовище, в якому щоденно працюють працівники ІТ-компаній.

Основні вимоги до проєкту:

- Створити тривимірну (3D) модель офісу з врахуванням сучасного дизайну, зонування приміщення, наявності робочих місць, техніки, меблів та елементів декору.
- Реалізувати перенесення моделі у VR-середовище, щоб користувач міг оглядати простір у віртуальній реальності з можливістю пересування.
- Зробити інтерфейс зручним для користувача: щоб легко можна було навігувати у просторі, не виникало складнощів у керуванні.
- Додати базові інтерактивні елементи, які дадуть змогу взаємодіяти з об'єктами — наприклад, піднімати та переміщувати речі.
- Забезпечити фізично коректну поведінку об'єктів — тобто, щоб предмети падали, відбивались, реагували на дотики так, як у реальному житті.

Цей проєкт повинен продемонструвати, як за допомогою сучасних VR-технологій можна візуалізувати роботу офісу, що займається розробкою ІТ-рішень у галузі енергетики. Така віртуальна модель може бути використана для демонстрацій, навчання нових співробітників або навіть для віддаленої роботи у VR.

## **1.2 Вибір інструментів і середовища розробки**

Проект створення віртуального офісу IT-компанії в енергетичній сфері потребує ретельного вибору технологій. Потрібно забезпечити реалістичну графіку та підтримку VR для повного ефекту присутності, а також ефективно працювати з тривимірними моделями інтер'єру. З огляду на ці вимоги, як основні інструменти обрано ігровий рушій Unreal Engine 5.5.4 та систему 3D-моделювання Blender. Нижче представлено аналіз цих рішень і порівняння з основними конкурентами (рушії Unity, Godot та пакети Autodesk 3ds Max, Maya тощо) за ключовими критеріями: якість графіки, підтримка VR, зручність у використанні, наявність безкоштовних ресурсів, документація та придатність до студентських/освітніх проєктів. На основі аналізу зроблено висновки на користь Unreal Engine і Blender та наведено приклади типових завдань, які зручно реалізувати в цих середовищах.

### **1.2.1 Вибір ігрового рушія для віртуального середовища**

Якість графіки та VR. Unreal Engine 5 славиться неперевершеною якістю візуальних ефектів та реалістичною графікою – цей рушій є вибором багатьох AAA-проєктів, що потребують передових технологій рендерингу. Unity також пропонує сучасні графічні можливості (зокрема, через HDRP-пайплайн), проте за максимальним рівнем деталізації він все ж поступається Unreal Engine у випадку висококласних сцен. Відкритий рушій Godot Engine придатний для менш ресурсоємних 3D-проєктів; хоча його функціональність у 3D постійно зростає, наразі за складною графікою та VR-можливостями Godot помітно відстає від Unity й Unreal. Щодо VR-підтримки, то і Unity, і Unreal Engine мають повноцінні засоби для створення VR-додатків на різних платформах (PC, мобільні пристрої, шоломи тощо). Зокрема, Unity відома як одна з найкращих платформ для розробки VR/AR із відмінною крос-платформною підтримкою, тоді як Unreal Engine націлений на найвищу продуктивність і реалізм у VR-середовищах. Для нашого проєкту це означає, що Unreal Engine здатен забезпечити фотореалістичний інтер'єр офісу та

стабільну роботу навіть зі складними сценами, тоді як Unity могла б вимагати ретельнішої оптимізації для досягнення подібної якості, а можливостей Godot могло б виявитися недостатньо.

Зручність та навчання. Криві навчання різних рушіїв помітно різняться. Unity традиційно вважається більш дружнім для новачків: цей рушій має відносно простий інтерфейс і зрозумілу структуру редактора, тож початківець швидше опанує базові навички розробки. Unreal Engine, навпаки, характеризується крутішою кривою навчання – потужність цього інструменту обумовлює й більшу складність, що може стати перепоною для невеликих команд чи студентів без достатнього досвіду. Втім, для зменшення порогу входження Unreal надає систему візуальної скриптової мови Blueprints, яка дозволяє створювати ігрову логіку без написання коду (шляхом з'єднання блок-схем). Це суттєва перевага для освітніх проєктів: багато взаємодій (наприклад, відкриття дверей, натискання кнопок чи телепортація у VR) можна реалізувати засобами Blueprint, не залучаючи C++-програмування. Godot Engine, своєю чергою, відзначається мінімалістичним інтерфейсом та власною мовою сценаріїв GDScript (синтаксично подібною до Python), що робить його дуже легким у вивченні навіть для початківців. Проте така простота поєднується з обмеженим функціоналом, тож для складних 3D/VR-завдань Godot використовується рідше.

Спільнота, ресурси та документація. Обидва провідні рушії мають величезні спільноти користувачів і багаторічну базу знань. Unity може похвалитися одним із найбільших каталогів готових ресурсів – Asset Store, де доступні тисячі моделей, текстур, скриптів та інших об'єктів (значна частина з них безкоштовна або умовно безкоштовна). Наявність такого обширного репозиторію значно прискорює розробку, дозволяючи студентам використовувати готові рішення замість створення всього з нуля. Unreal Engine також має свій Marketplace із великою кількістю контенту; до того ж Epic Games надає унікальну можливість безоплатного використання бібліотеки фотореалістичних мегасканів Quixel Megascans для проєктів на Unreal, що є важливою перевагою при створенні

інтер'єрів (можна брати готові якісні 3D-моделі об'єктів офісу). Обидва рушії добре задокументовані: є офіційні посібники, навчальні портали, відеоуроки та активні форуми підтримки. Godot має меншу, але зростаючу спільноту користувачів; як наслідок, кількість готових напрацювань і плагінів для нього менша. Проте, будучи відкритим проєктом, Godot теж має докладну документацію та ентузіастів-учасників, які допомагають одне одному. Щодо ліцензійної політики, всі три рушії доступні студентам без прямої оплати: Unity Personal є безкоштовним (до досягнення певного доходу проєкту), Unreal Engine також безкоштовний доти, доки проєкт не почне приносити значні прибутки (роялті 5% стягуються лише після перевищення ліміту \$3000 поквартально), а Godot є повністю безкоштовним і відкритим. Така доступність робить їх привабливими для навчальних проєктів.

Апаратні вимоги. Слід згадати, що Unreal Engine потребує продуктивнішого обладнання для комфортної роботи: його редактор і проєкти вимагають більше ресурсів комп'ютера (потужного графічного процесора, оперативної пам'яті тощо). Unity менш вибагливий до “заліза”, а Godot є найлегшим з-поміж рушіїв і може запускатися на скромних конфігураціях. У студентських умовах, де апаратні ресурси обмежені, це може бути важливим фактором вибору рушія. Утім, з огляду на поставлену мету (якісний VR у реалістичному офісі), команда схиляється до використання потужнішого Unreal Engine, попри вищі вимоги до системи.

### **1.2.2 Вибір засобів 3D-моделювання**

Функціональні можливості 3D-моделювання. Усі три провідні пакети (Blender, 3ds Max, Maya) мають потужні інструменти для створення і редагування 3D-моделей. Втім, їх спеціалізація дещо різниться. Autodesk 3ds Max традиційно використовується для створення детальних тривимірних середовищ і об'єктів, особливо в архітектурній візуалізації та дизайні інтер'єрів. Цей пакет пропонує відмінні полігональні та поверхневі моделюючі засоби, потужні текстурювання і

рендеринг (включно з інтеграцією рендерів на кшталт V-Ray, Arnold). Autodesk Maya більше відомий як індустріальний стандарт для високоякісної анімації та ригінгу персонажів; його інструментарій моделювання також сильний, але особливо Maya цінується за можливості створення складної кінематики, спецефектів і анімаційних сцен. Blender є універсальним рішенням з відкритим кодом: він поєднує інструменти полігонального і скульптурного моделювання, має систему процедурних модифікаторів та вузлів (Geometry Nodes), і навіть вбудовані засоби анімації та симуляції. Blender особливо відомий своїми можливостями цифрової скульптури (що корисно для органічних форм), тоді як для завдань технічного моделювання (будівлі, меблі) він так само ефективний, як і комерційні аналоги. У контексті моделювання офісного приміщення (переважно неорганічні об'єкти — стіни, меблі, прилади) можливостей Blender і 3ds Max цілком достатньо для створення високодеталізованих моделей; спеціалізовані функції Maya з персонажної анімації тут не є критичними.

Зручність використання та навчальні ресурси. Інтерфейс і криві навчання 3D-пакетів відрізняються. 3ds Max вважається досить дружнім для початківців, особливо в галузі архітектурного моделювання – його інтерфейс більш інтуїтивний (принаймні у порівнянні з Maya) і організований під типові технічні задачі. Maya є надзвичайно потужним, але й складним інструментом: новачкам важко одразу зорієнтуватися у багатстві функцій Maya, тому навчання потребує більше часу та спеціальних курсів (недарма Maya – індустріальний стандарт, який часто викладають у професійних школах). Blender донедавна критикували за незвичний інтерфейс, який відрізняється від продуктів Autodesk (його навіть називали «спірним» щодо зручності). Однак після оновлення інтерфейсу (версії 2.8+) Blender став значно зрозумілішим для нових користувачів; до того ж, він має налаштовуваний UI і величезну спільноту, що створила безліч навчальних матеріалів. В результаті, при достатній мотивації студент може опанувати Blender самостійно, спираючись на відкриті курси та онлайн-уроки. Варто відзначити, що як 3ds Max, так і Maya офіційно доступні безкоштовно для студентів (через освітні

ліцензії), проте складність їх освоєння і вища потреба в ресурсах підтримки може бути стримуючим чинником у рамках студентського проєкту.

Доступність та вартість. Тут Blender має очевидну перевагу: це повністю безкоштовне і відкрите ПЗ, яке можна легально завантажити й використовувати без ліцензійних обмежень. Відсутність плати за програму та відкритий код роблять Blender привабливим вибором для студентів і молодих команд, дозволяючи уникнути витрат на дорогі підписки. Натомість, 3ds Max і Maya є комерційними продуктами Autodesk з досить високою вартістю (понад \$1600 на рік кожен у разі стандартної ліцензії). Для освітнього використання Autodesk надає студентські ліцензії без оплати, але вони мають обмежений термін і призначені винятково для навчальних цілей. Окрім фінансової сторони, важлива також технічна доступність: Blender працює на всіх основних платформах (Windows, macOS, Linux), тоді як 3ds Max офіційно підтримує лише Windows. Таким чином, Blender забезпечує більшу гнучкість в умовах, коли у студентів можуть бути різні операційні системи та обладнання. Також варто зазначити, що завдяки відкритості Blender спільнота створила численні безкоштовні плагіни та бібліотеки моделей/матеріалів, які студенти можуть використовувати у своїх проєктах, тоді як екосистеми 3ds Max/Maya більш закриті і часто орієнтовані на платні професійні додатки.

### **1.2.3 Висновок: обґрунтування вибору Unreal Engine та Blender**

Зважаючи на вищенаведене порівняння, Unreal Engine 5.5.4 та Blender було обрано в якості основи проєкту через їх відповідність потребам віртуального офісу. Unreal Engine забезпечує неперевершену якість графіки, реалістичне освітлення та потужні VR-можливості, що дозволяє створити інтерактивне середовище з ефектом повного занурення. Це особливо важливо для сценарію IT-компанії в енергетичній сфері, де, наприклад, можна віртуально презентувати технологічні процеси або проводити тренінги для персоналу. Попри дещо складніший процес освоєння, Unreal дає змогу реалізувати сценарії, недсяжні або складні у конкурентів –

зокрема, високоякісну візуалізацію офісних приміщень з фізично коректною взаємодією об'єктів та користувача (VR-тури, маніпуляція предметами, симуляція роботи обладнання тощо).

Blender, у свою чергу, обрано як інструмент створення 3D-контенту через його універсальність та доступність. Він надає студентській команді повний набір засобів для моделювання інтер'єру – від побудови геометрії офісних приміщень і меблів до призначення матеріалів і текстур. Типові завдання на кшталт моделювання офісного столу, стільця чи серверної шафи, налаштування освітлювальних приладів або створення декоративних елементів зручно виконувати саме у Blender завдяки інструментам точного моделювання та широкій базі готових ресурсів. Імпорт створених моделей в Unreal Engine відбувається без ускладнень (через формати FBX/OBJ або спеціальні плагіни); на практиці Blender давно використовується для виготовлення і підготовки ігрових Asset'ів, що потім інтегруються в рушії на кшталт Unreal. Зв'язка “Blender + Unreal” забезпечує ефективний наскрізний процес розробки: від моделювання до віртуальної реалізації.

Таким чином, комбінація Unreal Engine 5 і Blender дозволяє досягти балансу між найвищою якістю віртуальної реалізації та доступністю інструментів для освітнього проєкту. Студенти мають нагоду опанувати сучасні індустріальні технології без фінансових витрат на софт, отримавши при цьому максимально реалістичний і функціональний результат – віртуальний офіс, що відповідає вимогам сфери енергетики та наочно демонструє потенціал VR-технологій у навчальному процесі.

### **1.3 Висновок до Розділу 1**

У ході аналізу та постановки задачі було сформульовано основну мету проєкту — створити реалістичну модель офісу ІТ-компанії в енергетичній сфері, яку можна переглядати у віртуальній реальності. Це передбачає не лише побудову



тривимірного простору, а й додавання елементів взаємодії з об'єктами, підтримку VR-режиму та забезпечення максимального ефекту присутності.

Після детального порівняння існуючих технологій було обґрунтовано вибір двох основних інструментів: Unreal Engine як рушія для реалізації логіки, взаємодії та візуалізації віртуального простору, та Blender як середовища для створення 3D-моделей. Саме ця комбінація дозволяє ефективно реалізувати проєкт з урахуванням сучасних вимог до якості графіки, інтерактивності й доступності інструментів у навчальному процесі. Обрані засоби забезпечують гнучкість, потужний функціонал та можливість розвитку проєкту в майбутньому.

## **2 АНАЛІЗ ІНСТРУМЕНТІВ РОЗРОБКИ ТА РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

У цьому розділі буде описано інструменти та технології, які були використані під час створення віртуальної моделі офісу ІТ-компанії в енергетичній сфері. Буде пояснено, чому саме ці інструменти були обрані для реалізації проєкту, які переваги вони мають, а також які ще варіанти розглядалися, але не були використані. Крім того, розглянемо вибір платформи для розробки, мови програмування та інші технічні рішення, що допомогли реалізувати проєкт у середовищі віртуальної реальності.

### **2.1 Вибір платформи розробки та мови програмування**

Для реалізації проєкту віртуальної реальності (VR-проєкту) було обрано ігровий рушій Unreal Engine як основну платформу розробки, а в ролі мови програмування використано його вбудовану систему візуального скриптингу Blueprints. Unreal Engine зарекомендував себе як потужний інструмент для створення застосунків VR, пропонуючи широкий набір можливостей для роботи з 3D-графікою, фізикою та взаємодією з VR-пристроями. Одним з ключових чинників вибору цього рушія стала наявність системи Blueprints – візуальної мови програмування, яка дозволяє розробникам створювати ігрову логіку шляхом з'єднання логічних блоків (вузлів) без необхідності писати програмний код вручну.

У даному проєкті Blueprints стали головним засобом побудови логіки поведінки і взаємодії. Візуальне скриптування працює за подієво-орієнтованим принципом: розробник налаштовує події та реакції на них, визначаючи, як система має реагувати на дії користувача чи зміни стану середовища. Наприклад, за допомогою Blueprints реалізовано обробку колізій (зіткнень) та інших тригерів: коли контролер користувача у VR торкається певного віртуального об'єкта, спрацьовує подія OnOverlap, що запускає задану послідовність дій. Таким чином задається інтерактивна поведінка об'єктів – від підбору й переміщення предметів

до активації механізмів або відтворення аудіо-візуальних ефектів – і все це програмується шляхом з'єднання відповідних вузлів у Blueprint-графі. Система Blueprints надає доступ до більшості можливостей рушія (включно з фізикою, керуванням об'єктами, анімацією тощо) через наочний інтерфейс, що істотно спрощує процес розробки інтерактивних сценаріїв у VR.

Ще одним важливим аспектом використання Blueprints є побудова логіки роботи інтерфейсу користувача. Unreal Engine містить модуль UMG (Unreal Motion Graphics) для створення інтерфейсу, який тісно інтегрується з системою Blueprints. У межах проєкту базові елементи UI – зокрема, меню та підказки – створювалися у вигляді віджетів (Widget Blueprint-класів). Це дозволило реалізувати взаємодію користувача з меню через візуальний сценарій: наприклад, при наведенні VR-контролера на кнопку меню і натисканні кнопки дії генерується подія, що обробляється у відповідному Blueprint і виконує задану функцію (перехід до іншої сцени, зміна налаштувань тощо). Таким чином, елементи інтерфейсу, які потребують реагування на дії користувача, повністю реалізовані засобами візуального програмування. Це забезпечило швидке ітеративне тестування та вдосконалення дизайну UI, а також зручне налаштування інтерфейсу під особливості VR-середовища.

Вибір Unreal Engine був зумовлений і його гнучкістю в плані програмування: рушій підтримує не лише Blueprints, але й традиційну розробку на C++. Наявність C++ як альтернативи вважається перевагою платформи, оскільки у випадках, коли потрібна глибша кастомізація функціоналу або оптимізація продуктивності, розробник може реалізувати частину логіки саме мовою C++ з подальшим викликом цього коду в Blueprint-сценаріях. Такий підхід дозволяє поєднати переваги швидкого прототипування та наочності Blueprints із можливістю низькорівневого контролю, який надає C++. У результаті обрана платформа забезпечує ефективну реалізацію задуманої логіки VR-проєкту засобами візуального скриптингу, залишаючи можливість розширення функціональності проєкту за рахунок C++ у разі зростання складності задач.

## 2.2 Мова програмування C++

Мова C++ відіграє ключову роль у розробці на Unreal Engine 5, особливо в контексті VR-проектів. Весь рушій Unreal написаний на C++, і ця мова використовується для створення логіки гри поза межами можливостей візуального скриптингу Blueprints. Завдяки C++ розробники можуть розширювати функціонал Unreal Engine, створюючи власні класи акторів (Actor) та компоненти з унікальною поведінкою, а також реалізовувати високоефективні, оптимізовані елементи ігрового процесу. Хоча Blueprints надають зручний спосіб швидкого прототипування, вони залишаються лише верхнім шаром доступу до можливостей рушія і не охоплюють їх повністю. Тому для реалізації спеціалізованих або низькорівневих аспектів проекту (наприклад, функціональності, не передбаченої в Blueprints, або складної серверної логіки) звертаються до C++. Крім того, у критичних з точки зору продуктивності випадках (як-от інтенсивні обчислення, складна математика або керування великим числом об'єктів у сцені) доцільно використовувати C++, щоб отримати максимальну швидкодію додатку. Для VR-додатків, де потрібна стабільна висока частота кадрів і мінімальна затримка, можливість оптимізувати ключові ділянки коду на C++ є особливо важливою.

Основні переваги C++ як мови загального призначення:

Висока продуктивність. C++ компілюється у машинний код і виконується безпосередньо, тоді як багато скриптових мов працюють через проміжні інтерпретатори або віртуальні машини. Зокрема в Unreal Engine, нативний C++-код працює значно швидше за Blueprints, які виконуються у віртуальній машині. Це робить C++ більш придатним для ресурсоемних, performance-critical завдань, що вимагають максимальної швидкодії.

Контроль над пам'яттю. На відміну від мов із автоматичним прибиранням пам'яті (як C# у Unity чи скриптових мов), C++ надає програмісту прямий контроль над виділенням і вивільненням пам'яті. Розробник може самостійно керувати пам'яттю, застосовуючи спеціальні аллокатори або оптимізуючи структури даних,

що дозволяє зменшити накладні витрати та уникати непередбачуваних пауз на збір сміття. Практика показує, що «bare metal»-алокації в C++ здатні підвищити продуктивність програми, хоча і потребують глибшого розуміння та планування роботи з пам'яттю. Такий рівень контролю особливо цінний у великих проектах і VR-додатках, де важлива кожна мілісекунда кадру.

Низькорівневе програмування. C++ забезпечує глибокий рівень доступу до системних ресурсів і API. Це означає, що розробник може опрацьовувати низькорівневі деталі – від роботи з потоками і файловою системою до використання вказівників та спеціалізованих інструкцій процесора. У контексті Unreal Engine C++ відкриває повний доступ до внутрішнього API рушія та його функцій, надаючи можливість реалізувати розширені функціональні можливості й інтегрувати зовнішні системи або бібліотеки. По суті, C++ дає повний контроль над поведінкою гри: можна написати власні алгоритми обробки фізики, нетипові шейдери рендерингу або підтримку нового апаратного пристрою – і все це на рівні движка, чого неможливо досягти суто засобами Blueprints.

Приклади використання C++ в ігрових та VR-проектах на Unreal Engine:

Створення нових класів акторів та компонентів із кастомною логікою. Наприклад, розробник може визначити власний клас Pawn або Actor для VR-персонажа чи інтерактивного об'єкта, щоб реалізувати поведінку, якої немає в стандартних компонентах рушія (особливі методи взаємодії, унікальна механіка тощо). Такі C++ класи можуть надалі використовуватися у редакторі як базові для Blueprint-акторів, комбінуючи гнучкість C++ і простоту налаштування у Blueprint.

Реалізація складних обробників подій та ігрової логіки. C++ дає змогу безпосередньо працювати з подіями і циклами оновлення: розробники часто перевизначають методи життєвого циклу Actor (BeginPlay, Tick тощо) чи створюють власні події (Event Dispatcher) в C++, щоб точніше контролювати поведінку гри. Наприклад, реакцію на зіткнення (OnOverlap, OnHit) або ввід від VR-контролерів можна обробляти на C++, гарантуючи швидке реагування без накладних витрат, притаманних Blueprints.

Розширення системи фізики та обробки взаємодій. У VR-проектах часто виникає потреба змінити стандартну фізичну модель або додати нові типи взаємодії. Через C++ можна напряду працювати з фізичним рушієм (PhysX/Chaos) Unreal Engine: створювати власні Actor Component для спеціальної фізики, застосовувати сили, обмеження (constraints) або писати кастомні рішення для зіткнень. Це дозволяє, наприклад, реалізувати більш реалістичну симуляцію дотиків і гаптичного зворотного зв'язку у VR, ніж це доступно «з коробки».

Оптимізація продуктивності проєкту. C++ використовується для оптимізації критичних ділянок коду, щоб гра працювала плавно. Розробники можуть переносити трудомісткі обчислення з Blueprints до C++ (зменшуючи накладні витрати виконання), реалізовувати багатопотокове виконання задач, тонко налаштовувати використання пам'яті та ресурсів. У VR-додатках, де необхідно підтримувати високу частоту кадрів (90+ FPS), такі оптимізації на C++ є вирішальними для комфорту користувача.

Інтеграція обладнання та зовнішніх бібліотек. Повна інтеграція Unreal Engine з C++ дозволяє підключати до проєкту сторонні програмні рішення і пристрої. Наприклад, для підтримки нового VR-шолома чи контролера з нестандартним SDK розробник може використати C++ для роботи з їхнім API. Так само можна підключати до гри зовнішні бібліотеки (для мережових функцій, штучного інтелекту, аналітики тощо) і викликати їх функції безпосередньо з C++ коду. Така гнучкість у виборі інструментів значно розширює можливості проєктів на Unreal Engine у порівнянні з використанням тільки вбудованих засобів.

Unreal Engine традиційно вважається одним із найпотужніших рушіїв у галузі саме завдяки повній інтеграції з C++. Рушій надає розробникам доступ до всього свого вихідного коду на C++, що забезпечує безпрецедентну свободу модифікації і розширення можливостей системи. Unreal Engine 5 написаний на C++ і підтримує всі ключові платформи (настільні ПК, консолі, мобільні пристрої, а також AR/VR). Важливо, що повний потенціал цього рушія розкривається лише за умови активного використання C++: саме ця мова дозволяє задіяти найглибші можливості

Unreal і реалізувати найвибагливіші ідеї розробників. Така тісна інтеграція є однією з причин, чому Unreal Engine часто обирають для великих проєктів – вона дозволяє гнучко адаптувати рушій під потреби конкретної гри або VR-додатку і досягти рівня якості, необхідного для AAA-продуктів.

Потенційні переваги використання C++ у навчальних, дослідницьких та індустріальних проєктах:

Навчальні проєкти. У освітньому середовищі використання C++ в рамках Unreal Engine сприяє глибшому розумінню принципів роботи ігрових технологій. Студенти отримують досвід низькорівневого програмування і керування ресурсами, що формує в них інженерне мислення. Попри відносно вищий поріг входу, оволодіння C++ дає майбутнім розробникам міцну базу для кар'єри в геймдеві: ця мова є однією з основних в індустрії, тож набуті навички безпосередньо відповідають потребам реальних проєктів. Навчальні VR-проєкти, де застосовується C++, дозволяють учасникам зрозуміти тонкощі оптимізації продуктивності та роботи апаратури, чого не завжди видно при використанні лише Blueprints.

Дослідницькі проєкти. У науково-дослідних та експериментальних VR-проєктах C++ відкриває можливість впровадження нетривіальних рішень. Дослідники можуть розробляти власні плагіни та модулі для Unreal Engine на C++, додаючи новий функціонал, якого не існує у стандартному рушії. Яскравим прикладом є плагіни на зразок UnrealNaptics, що реалізують розширені VR-взаємодії (гаптика, складне захоплення об'єктів) шляхом інтеграції спеціалізованих бібліотек зі швидкими алгоритмами фізичної взаємодії та підтримкою нестандартних пристроїв (наприклад, кібер-рукавичок). Використання C++ у таких проєктах дозволяє проводити експерименти на межі можливостей технології – дослідники не обмежені готовими інструментами, а можуть змінювати сам рушій або додавати нові функції для потреб своїх наукових експериментів. Це забезпечує більшу точність та реалізм симуляцій у VR-дослідженнях, що важливо для отримання достовірних результатів.

Індустріальні проекти. У комерційній розробці (ігровій індустрії та суміжних сферах) C++ є галузевим стандартом для створення високопродуктивних реалістичних додатків. Повна підтримка C++ в Unreal Engine – одна з причин, чому цей рушій став синонімом AAA-якості: він широко використовується для розробки провідних ігор з передовою графікою і складною логікою. Для великих команд розробників робота з текстовим C++ кодом є зручнішою в плані спільної розробки та контролю версій, ніж з суто візуальними скриптами. Крім того, C++ полегшує перенесення проектів між різними платформами (ПК, консолі, мобільні), що важливо для індустріальних продуктів з широкою аудиторією. З практичного погляду, можливість оптимізувати кожен аспект проекту на рівні коду C++ означає більш ефективне використання апаратних ресурсів та стабільність застосунку під виробничими навантаженнями. Таким чином, для індустріальних VR-рішень використання C++ забезпечує необхідну продуктивність, масштабованість і надійність, дозволяючи довести продукт до рівня, очікуваного споживачами та бізнесом.

### **2.3 Програмний пакет для створення тривимірної комп'ютерної графіки Blender**

Blender – це вільний, безкоштовний та з відкритим кодом програмний пакет для 3D-моделювання і комп'ютерної графіки. Він надає користувачам потужні інструменти для створення тривимірних сцен будь-якої складності. Blender використовується для розробки статичних зображень, 3D-анімації, візуальних ефектів (VFX) та навіть для монтажу відео. Завдяки широкому спектру можливостей цей пакет придатний для реалізації практично будь-яких проектів у сфері 3D: від навчальних або аматорських робіт до рекламних роликів і повнометражних фільмів – ним користуються як окремі ентузіасти, так і студії по всьому світу.

Основні функції Blender:



3D-модельовання. Програма оснащена широким набором засобів для полігонального та криволінійного модельовання, що дозволяють створювати і редагувати тривимірні моделі будь-якої форми. Користувач може побудувати модель «з нуля» або на основі готових примітивів, використовуючи інструменти для роботи з сітками, поверхнями та каркасами об'єктів.

Скульптинг (цифрове ліплення). Blender містить режим скульптингу, за допомогою якого можна «виліплювати» високодеталізовані форми, модифікуючи поверхню моделі немов глину. Ця функціональність дозволяє створювати складні органічні об'єкти (наприклад, персонажів) і вирізняється серед інших програм потужністю своїх засобів цифрового ліплення.

UV-розгортка. У Blender інтегровані інструменти для розгортання поверхні 3D-моделі в площину з метою підготовки до текстуровання. Користувач може визначати шви на моделі та розгортати сітку оптимальним чином, щоб забезпечити правильне накладання 2D-текстур на 3D-об'єкт. Ця функція є необхідною частиною підготовки моделі до подальшого текстуровання: Blender пропонує кілька методів автоматичного UV-розгортання для різних типів об'єктів, що дозволяє ефективно працювати як з простими, так і зі складними формами.

Текстуровання та матеріали. Blender надає широкі можливості призначення матеріалів і накладання текстур на об'єкти сцени. Є засоби як для традиційного UV-мапінгу (розміщення растрових текстур на поверхні моделі), так і для процедурного текстуровання за допомогою вузлових шейдерів. Вбудований режим Texture Paint дозволяє малювати безпосередньо по поверхні моделі, створюючи детальні карти текстур. Таким чином, усі етапи, пов'язані з підготовкою текстур і налаштуванням матеріалів, можуть виконуватися безпосередньо в середовищі Blender.

Анімація і рігінг. Пакет Blender містить повноцінний набір інструментів для створення анімації. Користувач може задавати ключові кадри та плавні переходи між ними (інтерполяцію), реалізуючи принцип покадрової анімації. Для оживлення персонажів Blender дозволяє створювати кістякову анімацію (рігінг) та

прив'язувати 3D-моделі до скелетів. Також підтримуються різноманітні фізичні симуляції (рух одягу, волосся, зіткнення об'єктів тощо) для реалістичного відтворення руху. Завдяки таким можливостям Blender придатний для створення як простих навчальних проєктів, так і складних анімаційних сцен у професійних виробництвах.

Рендеринг. Для візуалізації тривимірних сцен Blender оснащений двома власними рушіями рендерингу: Cycles (фотореалістичний рушій трасування променів) та Eevee (оптимізований рушій для рендерингу в режимі реального часу). Вони дозволяють отримати зображення високої якості прямо з Blender, без потреби в зовнішніх рендерерах. Крім того, у програмі є засоби композитингу та постобробки зображень, а також базовий інструментарій нелінійного відеомонтажу – це дає змогу виконати повний цикл виробництва графічного контенту в єдиному середовищі.

Blender успішно застосовується як в навчальних, так і в комерційних проєктах. В освітньому середовищі його популярність зумовлена доступністю (програма безкоштовна) та широкою спільнотою користувачів, які створили безліч навчальних матеріалів і доповнень. Відсутність ліцензійних витрат дозволяє навчальним закладам та студентам легально використовувати Blender на будь-якій кількості комп'ютерів. Для малого бізнесу та інді-розробників цей інструмент привабливий тим, що забезпечує професійний рівень можливостей без необхідності купувати дорогі програмні пакети. До того ж, Blender є крос-платформним застосунком (Windows, macOS, Linux) і має відносно невисокі вимоги до обладнання, що розширює сферу його використання на різних системах навіть з обмеженими ресурсами.

Ще однією важливою перевагою Blender є підтримка експорту моделей у численні формати, сумісні з зовнішніми рушіями та ігровими двигунами. Зокрема, результати, створені в Blender, можна експортувати у формати FBX, OBJ, DAE (Collada), glTF та багато інших. Це забезпечує сумісність з такими середовищами, як Unreal Engine та Unity, оскільки вони підтримують імпорт зазначених форматів.

Таким чином, Blender легко інтегрується в процес розробки ігор чи інтерактивних додатків: наприклад, художник може змодельовати і затекстурувати об'єкт у Blender, а потім перенести його до Unreal Engine для подальшого використання в ігровому просторі.

Порівняння з комерційними аналогами: У галузі 3D-графіки традиційно домінують платні продукти, такі як Autodesk 3ds Max та Autodesk Maya. Однак Blender пропонує ряд переваг, що вигідно вирізняють його на їхньому фоні. Найбільш очевидна перевага – відсутність ліцензійних витрат: Blender є повністю безкоштовним, тоді як професійні пакети Autodesk потребують дорогих передплат (близько \$230 на місяць за ліцензію). Це означає, що навіть невеликі студії чи незалежні митці можуть використовувати Blender легально без фінансових бар'єрів. Другий аспект – відкритий вихідний код: Blender розвивається спільнотою розробників і ентузіастів по всьому світу, що забезпечує швидкий випуск оновлень та виправлень, а також дає можливість розширювати функціонал програми через скрипти і доповнення (завдяки вбудованому API Python).

Крім того, Blender є універсальним програмним комплексом: він об'єднує у собі моделювання, скульптинг, анімацію, рендеринг, композитинг і монтаж – тоді як у багатьох платних рішеннях для досягнення такої ж комплексності потрібне придбання окремих модулів або плагінів. Варто зауважити, що за якістю результату Blender уже наближається до промислових стандартів: сучасні рушії Cycles та Eevee забезпечують реалістичне зображення, а інструменти анімації й симуляції постійно вдосконалюються. Хоча Maya та 3ds Max досі широко застосовуються у великих студіях (завдяки багаторічній присутності на ринку та вузькоспеціалізованим функціям), Blender стрімко набирає популярності як в освіті, так і у виробництві розважального контенту та активно використовується у розробці інді-ігор, рекламної 3D-графіки та навіть деяких кінофільмів. Отже, Blender виступає конкурентоспроможною альтернативою комерційним 3D-пакетам, пропонуючи користувачам потужний набір інструментів без жодних фінансових витрат.

## **2.4 Налаштування та генерація колізій об'єктів в Unreal Engine 5**

### **2.4.1 Ручне налаштування колізій**

Unreal Engine 5 надає гнучку систему налаштування колізій, що дозволяє визначити, як об'єкти взаємодіють під час зіткнень. Основні елементи цієї системи – пресети колізій (Collision Presets), параметр Collision Enabled, тип об'єкта (Object Type) та реакції на колізії (Collision Responses), які включають режими Block (блокувати), Overlap (перетинати) та Ignore (ігнорувати). Взаємодія об'єктів визначається їх налаштуваннями з обох сторін: для зіткнення враховується і тип каналу об'єкта, і налаштування його реакції на канал іншого об'єкта. Якщо один об'єкт налаштований блокувати, а інший – лише перетинати, результатом буде перетин (Overlap) замість блокування; тільки коли обидва об'єкти бажають блокувати, зіткнення повністю блокується.

### **2.4.2 Канали та реакції колізій**

Канали колізій визначають категорії, за якими відбувається фільтрація зіткнень. Існують два типи каналів:

Object Channels (канали об'єктів) – типи, що призначаються самим об'єктам. Кожен об'єкт має свій тип (Object Type), що описує його природу (наприклад, WorldStatic – нерухомий об'єкт оточення, WorldDynamic – рухомий об'єкт, Pawn – персонаж або VR-кабіна, PhysicsBody – фізичне тіло, тощо). У проекті за замовчуванням є кілька вбудованих типів (WorldStatic, WorldDynamic, Pawn, PhysicsBody, Vehicle, Destructible тощо) і можна додавати власні (до 32 каналів). Тип об'єкта визначає його поведінку за типовим профілем колізії та спрощує глобальне налаштування – наприклад, всі об'єкти типу Pawn можуть мати схожі реакції.

Trace Channels (канали трасування) – умовні канали, що використовуються при трасуванні (кастомні промені, лінійні перевірки). Коли ми виконуємо, скажімо,

Line Trace by Channel, ми вказуємо, по якому каналу перевіряти (наприклад, Visibility чи Camera – ці канали є стандартними). Об'єкти налаштовують, чи блокують або пропускають такі трасування.

Collision Responses (реакції колізії) визначають, як об'єкт реагує на інші канали при зіткненні. Для кожного каналу (і об'єктного, і трасуючого) можна встановити одну з трьох реакцій:

- Block (Блокувати) – об'єкт фізично блокує інший об'єкт/трасу на цьому каналі. Якщо відбувається зіткнення двох тіл і обидва налаштовані на Block відносно типу один одного, виникає Hit Event (подія удару) і рух блокується.
- Overlap (Перетинати) – об'єкт не перешкоджає проникненню іншого, але генерує подію Overlap Event при перетині. Перетин спрацьовує, якщо інший об'єкт налаштований як Overlap або Block по відношенню до даного типу. Тобто, хоча фізично вони не зупиняються, подія перетину відбувається для логіки гри (наприклад, входження в зону тригера).
- Ignore (Ігнорувати) – об'єкт повністю ігнорує взаємодію по заданому каналу. Якщо один з двох об'єктів встановлений на Ignore щодо іншого, зіткнення не відбувається взагалі (ні подій, ні фізичної реакції).

Налаштування реакцій виконується через матрицю каналів у деталях об'єкта. Для Object Responses вказується реакція на кожен тип об'єкта (WorldStatic, WorldDynamic, Pawn тощо), а для Trace Responses – на трасувальні канали (Visibility, Camera та інші). Важливо, що результат зіткнення визначається парою об'єктів: наприклад, якщо гравець (Pawn) наближається до куща (WorldStatic), і Pawn налаштований блокувати WorldStatic, а кущ налаштований перетинати Pawn, то в сумі вони лише перетнуться (Overlap), дозволивши пройти крізь кущ. Якщо ж обидва об'єкти мають реакцію Block один на одного (наприклад, гравець і стіна), то станеться фізичне блокування руху і подія Hit. Завдяки цьому, можна досягти складних сценаріїв фільтрації колізій – наприклад, зробити “примарного” гравця,

змінивши його реакцію на Pawn з Block на Ignore (в результаті він проходитиме крізь інших гравців).

Окрім ручного налаштування, Unreal Engine може автоматично визначати форму колізії для об'єкта. Існують два підходи: використання простого чи складного контуру колізії та генерація опуклих оболонок через інструмент Auto Convex Collision.

### 2.4.3 Прості vs. складні колізії

Кожен статичний меш або геометричний об'єкт в UE5 може мати дві репрезентації колізії: simple (проста) і complex (складна). Складна колізія (Complex Collision) – це використання безпосередньо геометрії рендеру (трикутників меша) для зіткнень. Іншими словами, двигун бере кожен полігон моделі як площину для перевірки зіткнення – це дає найточніше співпадіння форми, корисне для, скажімо, трасування куль – можна “вистрілити” і точно влучити саме в видиму деталь моделі. Проста колізія (Simple Collision) – це спрощена модель зіткнення, що складається з базових примітивів: сфер, коробок, капсул та опуклих оболонок (convex hulls). Прості колізійні форми приблизно обгортають об'єкт, згладжуючи дрібні деталі.

При кожній перевірці колізії можна вказати, яку геометрію використовувати – просту чи складну. За замовчуванням, рух гравця (наприклад, капсула Pawn при русі) використовує simple-колізії, щоб не “чіплятися” за дрібні виступи моделі. Натомість лінійне трасування променем на видимість або постріл зброї часто налаштовують на complex-колізії, щоб влучати точно по деталях об'єкта. В редакторі є спеціальний режим перегляду колізій (Player Collision), що показує, з чим реально взаємодіє гравець у світі – він візуалізує прості колізійні форми, якими підмінені складні моделі.

Візуалізація простих колізійних форм в рівні (праворуч, блакитні примітиви) порівняно з реальною геометрією сцени (ліворуч). Примітиви спрощують контури об'єктів, щоб уникнути зачеплення за дрібні деталі при русі персонажа. У налаштуваннях статичного меша можна вибрати політику колізії: використовувати просту, складну або обидві репрезентації.

UE5 дозволяє обирати політику зіткнень меша через параметр Collision Complexity у редакторі статичного меша. Опції такі:

- Project Default – за замовчуванням проєкту: прості запити колізії (наприклад, рух) використовують просту форму, а складні запити (трасування по видимій геометрії) – складну. Це стандартна поведінка.
- Simple and Complex – генерувати і прості, і складні колізії. Двигун сам вибиратиме відповідну форму залежно від типу запиту (аналогічно до Project Default, фактично).
- Use Simple Collision As Complex – примусово ігнорувати складну геометрію навіть для складних запитів: якщо запит вимагає complex-колізію, все одно використати прості форми. Цей режим економить пам'ять (не зберігається міш як коллайдер) та може прискорити перевірки, якщо проста форма суттєво простіша за модель.
- Use Complex Collision As Simple – примусово використовувати полігональну геометрію як колізію навіть для простих запитів. Тобто фактично використовувати лише complex-колізію для всіх випадків. Цей режим дозволяє, наприклад, точно симулювати фізику об'єкта за його реальною формою. Важливо: об'єкти з встановленим Use Complex As Simple не можуть симулювати фізику (їх не можна увімкнути як PhysicsBody).

#### 2.4.4 Інструмент Auto Convex Collision

Для об'єктів складної форми, особливо увігнутих (concave), вручну створювати набір простих колізійних примітивів може бути трудомістко. Unreal Engine пропонує інструмент Auto Convex Collision для автоматичної генерації опуклих оболонок (convex hulls), що приблизно повторюють форму моделі. Цей інструмент доступний в редакторі статичних мешів (меню Collision > Auto Convex Collision). При активації відкривається панель налаштувань Convex Decomposition, де можна задати параметри генерації оболонок:

- Hull Count – максимальна кількість опуклих частин, на які буде розбито колізію. Чим більше hull-частин, тим точніше можна наблизитися до форми об'єкта, особливо якщо об'єкт має кілька окремих виступів або порожнин.
- Max Hull Verts – максимальна кількість вершин (полігональної складності) для однієї опуклої оболонки. Обмеження на вершини дозволяє контролювати детальність кожного примітива: менше значення спрощує форму кожної оболонки, більше – дозволяє складніші контури.
- Hull Precision – кількість вокселів для вокселізації моделі при генерації колізії. Інструмент розбиває об'єм об'єкта на тривимірну сітку (вокселізація) і на її основі будує опуклі оболонки. Вищий Precision (більше вокселів) дає точніше наближення форми, але збільшує час розрахунку.

Коли ви налаштували ці параметри (наприклад, типовими є ~20–32 hull count, ~16 вершин, точність десятки тисяч вокселів), потрібно натиснути Apply – UE5 виконає алгоритм декомпозиції у фоновому режимі та побудує набір опуклих колізійних мешів. Існуючі старі коллайдери при цьому замінюються новими. Результат можна переглянути, увімкнувши відображення Simple Collision для меша – зелені дроти або непрозорий шар покажуть згенеровані оболонки.

Алгоритм, що використовується в Auto Convex Collision, базується на принципах наближеної опуклої декомпозиції. Починаючи з Unreal Engine 4, в



двигун інтегровано бібліотеку V-HACD (Volumetric Hierarchical Approximate Convex Decomposition) для побудови колізій. У UE5 використовується оновлена версія V-HACD, яка дає більш точний результат автоматичної генерації колізій. Принцип її роботи: модель апроксимується сукупністю опуклих частин, які в сумі максимально заповнюють вихідний об'єм. Спочатку об'єкт вокселізується з заданою точністю, потім алгоритм ітеративно кластеризує ці вокселі в опуклі групи, на кожному з яких будується опуклий корпус. Параметр Hull Count обмежує кількість груп (тобто деталей), а Precision визначає розмір вокселів, що впливає на мінімальний помітний дрібний елемент форми. Декомпозиція триває, поки не буде досягнуто потрібної кількості оболонок або поки похибка (різниця об'єму між сумою hull-ів та оригіналом) не стане нижче певного допуску. Опукла оболонка кожної групи – це найменший опуклий полігональний поліедр, що містить усі точки (вокселі) цієї групи; таким чином увігнуті частини об'єкта автоматично розбиваються на кілька опуклих сегментів. При високих значеннях Hull Count і Precision наближення буде дуже точним, хоча і складним (багато примітивів), а при низьких – об'єкт отримає грубу колізійну форму. Розробник може експериментувати з налаштуваннями, щоб отримати баланс між точністю колізії та продуктивністю.

#### **2.4.5 Типові проблеми з колізіями у VR-проектах**

Неправильні налаштування колізій особливо помітні у VR-проектах, де користувач фізично взаємодіє з віртуальним світом. Ось кілька поширених проблем, що виникають через некоректні колізії:

Проходження крізь об'єкти: VR-користувач або контролери можуть прослизати крізь стіни чи предмети. Наприклад, гравець може просунути голову або руку за межі стіни, якщо колізія для цих об'єктів неправильно налаштована. Також, якщо капсула Pawn відключена або об'єкти оточення ігнорують Pawn, гравець просто проходить крізь них без опору.

Неправильне підбирання предметів: Взаємодія при хватанні об'єктів може не спрацьовувати або давати артефакти. Наприклад, при спробі взяти предмет рукою в VR подія `Overlap` може не генеруватися, якщо в об'єкта чи руки невірні виставлені канали (вони можуть блокуватися замість перетину). Інший випадок – об'єкт виривається з рук або телепортується, якщо під час хватання його колізія конфліктує з колізією гравця чи оточення (може відбуватися відштовхування або “вибух” фізики).

Провалювання крізь поверхні: Предмети або навіть сам гравець можуть провалитися крізь підлогу чи інші геометрії. Це трапляється, коли колізійна форма підлоги надто проста або має щілини (наприклад, дуже тонка колізія, через яку на великій швидкості проскакують об'єкти), або коли рухомий об'єкт не отримує вчасно обробку колізії (проблема тунелювання на великій швидкості). У VR, де гравець може кинути об'єкт швидко, фізичне тіло інколи проходить крізь статичну підлогу, якщо та не має достатньо точної колізії чи не ввімкнено спеціальні заходи проти тунелювання.

#### **2.4.6 Вирішення проблем у VR через налаштування колізій**

Правильне налаштування колізій – як вручну, так і за допомогою автоматичних інструментів – допомагає усунути описані проблеми у віртуальному середовищі:

Запобігання проходженню крізь об'єкти: Переконайтеся, що для статичних об'єктів оточення використано профіль, який блокує `Pawn` (наприклад, `BlockAll` або `BlockAllDynamic` замість `OverlapAll` чи тим більше `IgnoreOnlyPawn`). Наприклад, профіль `IgnoreOnlyPawn` відносить об'єкт до `WorldDynamic` і ставить `Ignore` для `Pawn`, через що гравець зможе пройти крізь нього. У VR-шаблоні `Pawn` має капсулу зі стандартним типом `Pawn`, тож стіни й предмети повинні блокувати цей тип. Якщо гравець “фантомний”, перевірте, щоб `Collision Enabled` у `Pawn` не було вимкнено і щоб об'єкти не ігнорували `Pawn`-канал. Також, для запобігання

просовування голови крізь стіну, можна додатково відстежувати положення камери і застосовувати механізми на кшталт відсування або затемнення екрану, коли голова проходить крізь колізію, але базовий рівень – це коректна колізійна капсула персонажа та блокування її оточенням.

Коректне підбирання об'єктів: Для об'єктів, які гравець може хапати, варто налаштувати колізію таким чином, щоб руки/контролери генерували події перетину, а не блокувалися. Зазвичай у VR рука реалізована як контролер зі сферичним колайдером, який налаштований на `Overlap` з дрібними предметами. Предмет, що піднімається, повинен мати ввімкнені `Overlap Events` і реакцію `Overlap` на канал руки (наприклад, віднести руку до типу `Pawn` або спеціального каналу `Hand`, а в об'єкта виставити `Overlap` для нього). Це гарантує, що спрацює тригер захоплення. Коли предмет взято, часто відключають фізику об'єкта (`Simulate Physics = false`) і прив'язують його до контролера рук, щоб він рухався разом із рукою без затримок. При цьому можна тимчасово змінити колізійний профіль об'єкта на спеціальний (наприклад, `Pickup`), в якому об'єкт ігнорує колізії з гравцем, щоб не відштовхуватися від капсули `Pawn`, але при цьому може залишатися активним для інших взаємодій. Після відпускання об'єкта фізику знову вмикають, а профіль повертають на звичайний (`PhysicsActor` тощо), щоб він взаємодівав зі світом. Такий підхід запобігає конфліктам – об'єкт не “вибухне” в руках і не підкине гравця, бо на час хватання він не відштовхується від колізій гравця. Важливо також, щоб при симуляції фізики підбраний об'єкт мав просту (`convex`) колізію. Якщо об'єкт залишався з `Use Complex As Simple`, фізика на ньому не працюватиме. Тому для усіх фізичних взаємодій (кидання, падіння) переконайтесь, що у об'єкта є коректні `simple`-колайдери (можна призначити вручну примітиви або використати `Auto Convex`).

Уникнення провалювання крізь підлогу та об'єкти: Для статичних поверхонь (підлога, стіни) рекомендується використовувати достатньо точну колізію. Якщо підлога – це тонка площина або складна модель, розгляньте встановлення для неї `Collision Complexity = Use Complex Collision As Simple` (щоб двигун

використовував сам меш підлоги для колізії). Оскільки підлога статична, це безпечно і гарантує, що навіть нерівності будуть враховані. Однак рухомі об'єкти, які падають на підлогу, при великій швидкості можуть “проскакувати” крізь будь-яку колізію через ефект тунелювання, коли об'єкт переміщується за один фізичний крок далі, ніж товщина підлоги. Для вирішення: по-перше, зробіть колізійні форми підлоги товстішими (наприклад, екструдуйте площину в коробку або додайте невидиму колізійну коробку під підлогою). По-друге, увімкніть Continuous Collision Detection (CCD) для швидкісних об'єктів – в налаштуваннях компоненту є прапорець CCD, який змушує фізичний рушій ретельніше відслідковувати колізії при швидкому русі, запобігаючи проникненню. По-третє, можна зменшити крок симуляції фізики (увімкнути субкроки), щоб зіткнення розраховувалися частіше. На практиці, у великій VR-сцені ручне додавання колізій може бути трудомістким, але інструмент Auto Convex Collision допомагає отримати досить точні колізії для нестандартних форм – застосуйте його до моделей оточення, з якими гравець може взаємодіяти, щоб уникнути невидимих щілин у колізії. Правильна колізія запобіжить провалюванню: об'єкти будуть стабільно лежати на підлозі, а гравець не провалиться крізь геометрію навіть при швидких рухах віртуальних предметів.

Таким чином, поєднання ретельного ручного налаштування (через профілі та канали колізій) і автоматичних засобів (генерація convex-оболонки для складних об'єктів) дозволяє створити надійне колізійне середовище у VR. Це забезпечить реалістичні взаємодії – гравець відчуватиме коректні фізичні межі об'єктів, зможе впевнено хапати предмети, і не буде “провалюватися” або проходити крізь елементи віртуального світу. При підготовці VR-проектів варто закладати додатковий час на тестування й налагодження колізій, оскільки від цього безпосередньо залежить іммерсивність та комфорт користувача. Правильно налаштовані колізії значно підвищують якість взаємодії у віртуальній реальності, роблячи її більш правдоподібною та приємною для користувача.

## **2.5 Висновок до Розділу 2**

У цьому розділі було проаналізовано основні інструменти, використані під час реалізації проєкту, зокрема платформу розробки Unreal Engine 5, мову програмування C++ та програмний пакет для 3D-моделювання Blender. Вибір Unreal Engine був зумовлений його сучасними технологіями для побудови віртуального середовища, підтримкою VR та зручним візуальним програмуванням через Blueprints. C++ розглядався як потужна низькорівнева мова з широкими можливостями оптимізації та розширення функціоналу рушія. Blender, у свою чергу, став основним інструментом для створення 3D-моделей завдяки безкоштовному доступу, широкому функціоналу і сумісності з Unreal Engine. З огляду на цілі проєкту, обрані інструменти забезпечили оптимальний баланс між функціональністю, доступністю й можливістю подальшого розвитку інтерактивного VR-середовища.

## **3 ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Процес розробки віртуальної моделі офісу IT-компанії в енергетичній сфері складався з кількох важливих етапів, які забезпечили успішну реалізацію проєкту. У цьому розділі будуть детально описані кожен з цих етапів, включаючи планування структури віртуального середовища, вибір технологій, створення 3D-моделей офісу, інтеграцію віртуальної реальності, а також тестування і оптимізацію проєкту.

### **3.1 Проектування архітектури віртуального офісу**

Проектування архітектури віртуального офісу є одним із найважливіших етапів розробки, оскільки від цього залежить зручність використання віртуального середовища, взаємодія користувача з об'єктами та загальний досвід перебування в офісі. В даному випадку, проектування архітектури включало в себе кілька важливих етапів, таких як планування простору, визначення зон та їх функцій, створення логічних зв'язків між елементами, а також врахування специфічних вимог до взаємодії з користувачем в умовах віртуальної реальності.

Першим етапом проектування була постановка завдання щодо того, як саме має виглядати офіс, яку атмосферу потрібно створити та які функціональні зони мають бути присутніми. Це дозволило визначити необхідні об'єкти, такі як робочі місця, зона відпочинку, а також елементи, які забезпечують інтерактивність — наприклад, предмети, з якими можна взаємодіяти (підбирати, переміщати, перевертати тощо).

Після визначення загальної концепції, була спроектована структура офісу в тривимірному просторі. Важливою частиною цього етапу було також продумування взаємодії між різними зонами офісу та розміщення об'єктів таким чином, щоб користувач відчував себе комфортно в процесі переміщення. Розподіл простору був виконаний з урахуванням логіки переміщення віртуального

персонажа (користувача) в межах офісу, де кожна зона повинна була бути доступною і взаємопов'язаною з іншими через природні шляхи. На рисунку 3.1 та рисунку 3.2 можна побачити порівняння реального офісу та моделі з проєкту.



Рисунок 3.1 – Реальне фото офісу



Рисунок 3.2 – Скріншот з проєкту

Далі було виконано проектування «меню» або інтерфейсу для користувача. Це дозволило передбачити, як користувач буде взаємодіяти з віртуальними елементами в офісі. Зокрема, для полегшення навігації у віртуальному середовищі були створені елементи управління, такі як кнопки для виклику меню, інформаційні панелі та інші інструменти, які дозволяють управляти функціями

офісу, змінювати вигляд об'єктів або взаємодіяти з різними предметами. Усе це було необхідно для того, щоб користувач міг комфортно орієнтуватися у віртуальному середовищі та здійснювати необхідні дії без зайвих труднощів.

Також, на етапі проектування архітектури офісу, було передбачено подальше використання фізики для симуляції реальних взаємодій з об'єктами. Це дозволило забезпечити коректну поведінку об'єктів при взаємодії з ними, що має велике значення для реалізму віртуального середовища. Наприклад, якщо користувач підбирає предмет з столу, то він має плавно піднімати його, а не «проходити» через нього, що додає реалістичності і полегшує сприйняття віртуальної реальності.

У процесі проектування також було враховано важливість оптимізації простору з точки зору продуктивності. Оскільки для роботи з віртуальною реальністю потрібні високі показники графічної продуктивності, важливо було створити таку архітектуру, яка б дозволяла ефективно відображати всі елементи без значних втрат у продуктивності при переході між різними частинами офісу або при великій кількості одночасно активних об'єктів.

Таким чином, проектування архітектури віртуального офісу включало комплексний підхід до створення інтер'єру, планування простору, забезпечення інтерактивності та врахування специфічних вимог для роботи з віртуальною реальністю, що в сукупності дозволило створити комфортне та ефективне середовище для користувача.

### **3.2 Створення 3D-моделей та текстуровання**

Створення 3D-моделей та текстуровання є важливою частиною розробки віртуального офісу, оскільки саме ці елементи формують візуальне середовище, з яким взаємодіє користувач у VR. Для побудови моделей було використано програмне забезпечення Blender, а також велику кількість безкоштовних ассетів, доступних в інтернеті. Це дало змогу значно скоротити час на моделювання та зосередитись на інтеграції та оптимізації проєкту.



Основна частина 3D-моделей була завантажена з відкритих онлайн-бібліотек, таких як Sketchfab, Quixel Megascans, Poly Haven та інші джерела, які пропонують якісні безкоштовні моделі для комерційного та некомерційного використання. Серед об'єктів, які були завантажені, можна виділити:

- офісні меблі: столи, стільці, дивани, полиці;
- техніка: монітори;
- декор: лампи, картини, вазони, годинники;

Вибір на користь готових моделей був зроблений для того, щоб скоротити час розробки та приділити більше уваги інтерактивності й реалізації фізики віртуального середовища. Всі завантажені моделі були ретельно перевірені на якість і оптимізацію для використання в Unreal Engine. За потреби вони редагувалися у Blender для покращення текстур, оптимізації полігонів або налаштування UV-розгортки.

Деякі об'єкти, яких не вдалося знайти у вільному доступі або які потребували унікального дизайну, були змодельовані самостійно в Blender. Після завершення процесу моделювання ці об'єкти були підготовлені для експорту в Unreal Engine. Для цього використовувався формат GLB, який забезпечує правильне перенесення геометрії, текстур та матеріалів. В Unreal Engine моделі проходили додаткову перевірку на коректність відображення та інтегрувалися в загальне віртуальне середовище офісу.

Процес текстурювання включав нанесення матеріалів на об'єкти, щоб надати їм більш реалістичний вигляд. Для цього використовувалися як стандартні текстури з Blender, так і текстури, знайдені в інтернеті. Деякі моделі потребували додаткового налаштування UV-розгортки, щоб текстури правильно відображалися на поверхнях без спотворень.

Таким чином, поєднання використання готових моделей з інтернету та самостійного моделювання у Blender дозволило швидко і якісно наповнити віртуальний офіс реалістичними об'єктами, готовими для взаємодії у VR-

середовищі. Це допомогло скоротити час розробки та зосередитись на важливих аспектах інтерактивності та оптимізації.

### **3.3 Розробка інтерактивних елементів**

Розробка інтерактивних елементів у віртуальному офісі передбачала реалізацію можливості користувача взаємодіяти з об'єктами у VR-середовищі. Основні інтерактивні елементи включали підбір предметів, їх переміщення, а також коректне відображення фізичних взаємодій. Для цього були налаштовані колізії об'єктів, а також створені механізми, що дозволяють користувачу природно взаємодіяти з оточенням, використовуючи VR-контролери. Це забезпечило більш реалістичний досвід під час перебування у віртуальному офісі.

#### **3.3.1 Взаємодія з об'єктами (підбір предметів)**

Взаємодія з об'єктами (підбір предметів) у віртуальному офісі є однією з ключових функцій, що дозволяє користувачеві більш глибоко зануритися у VR-середовище. Механізм підбору об'єктів реалізований за допомогою інтеграції VR-контролерів з рушієм Unreal Engine. Це дозволяє користувачу вільно маніпулювати різними предметами в офісі: піднімати їх, переміщати та розміщувати в інших частинах простору.

Коли гравець намагається схопити об'єкт за допомогою контролера, система перевіряє тип захоплення (вільне, фіксоване або з прив'язкою), після чого вимикає фізику об'єкта (щоб він більше не реагував на гравітацію чи зіткнення) і "прикріплює" його до контролера — це створює ефект, ніби об'єкт знаходиться у руці користувача. Далі відбувається орієнтація об'єкта: його положення та обертання вирівнюється відносно положення віртуальної руки, щоб він виглядав природно під час взаємодії. Також перевіряється, чи вдалося захопити об'єкт, і якщо так — активуються додаткові події: передається інформація про контролер,

викликається відповідна подія у Blueprint'i, і програвається тактильний (гаптичний) зворотний зв'язок, щоб користувач фізично відчув момент захоплення.

Коли користувач відпускає тригер на контролері або виконує іншу дію, що має призвести до відпускання, система перевіряє, чи справді об'єкт ще тримається (через змінну стану) і яка саме логіка має бути застосована залежно від типу захоплення (вільне, прив'язане або користувацьке). У разі успішного відпускання викликається подія, яку можна додатково налаштувати для окремих об'єктів (наприклад, програвання анімації чи звуку відпускання). Далі виконується перевірка, чи потрібно вмикати фізику об'єкта після відпускання (тобто дозволити йому знову підкорятися законам гравітації та взаємодії з іншими об'єктами). Якщо так — увімкнюється симуляція фізики, після чого об'єкт "від'єднується" від віртуальної руки користувача. Таким чином забезпечується плавний перехід об'єкта з керованого стану (коли він закріплений за контролером) до вільного стану у VR-сцені, з можливістю реалістичного падіння або відскоку, залежно від фізичних параметрів.

У результаті ця система створює природну і реалістичну поведінку предметів під час взаємодії з ними у віртуальній реальності.

Для забезпечення коректної взаємодії з об'єктами були налаштовані колізії та фізичні властивості кожного з предметів. Це включає визначення їхньої ваги та реакції на дотики. Під час підбору об'єкта контролер користувача "закріплюється" за модель, і фізика руху відповідає за те, щоб об'єкт рухався в просторі відповідно до положення руки користувача.

Крім того, для запобігання випадковим помилкам, було додано механізм перевірки взаємодії, що визначає, чи знаходиться об'єкт у зоні досяжності. Це запобігає ситуаціям, коли предмет може "випасти" з рук або пройти крізь інші об'єкти.

Таким чином, підбір предметів у віртуальному офісі реалізований з максимальним врахуванням фізичної коректності та інтерактивності, що забезпечує більш реалістичний досвід для користувача.

### 3.3.2 Тестування взаємодії користувача з віртуальним середовищем

Тестування взаємодії користувача з віртуальним середовищем є важливим етапом розробки, що дозволяє перевірити коректність роботи інтерактивних елементів та забезпечити комфортне використання VR-середовища. Метою тестування було виявлення можливих помилок у взаємодії з об'єктами, перевірка фізичної коректності рухів, а також оцінка плавності та стабільності роботи додатку.

Під час тестування було проведено кілька видів перевірок:

1. Функціональне тестування — перевірка основних взаємодій користувача з об'єктами: підбір предметів, їх переміщення та взаємодія з поверхнями. Тестувалося, чи об'єкти коректно піднімаються, не "провалюються" крізь підлогу та чи правильно реагують на дотики.
2. Тестування фізики — оцінка поведінки об'єктів під час маніпуляцій у VR. Було перевірено, чи дотримуються вони законів фізики під час підбору, обертання та кидання. Окрему увагу приділено зіткненням з іншими предметами.
3. Відстеження колізій — аналіз того, як об'єкти реагують на дотик користувача та інші предмети у середовищі. Виявлялися помилки, коли предмети могли "проходити" крізь стіни або інші об'єкти без коректної взаємодії.
4. Тестування VR-контролерів — перевірка роботи взаємодії з використанням VR-контролерів. Оцінювалася точність захоплення об'єктів, відповідність рухів у віртуальному середовищі до фізичних дій користувача.
5. Зручність навігації — оцінка того, наскільки легко користувач може орієнтуватися у віртуальному просторі, підбирати предмети та переміщатися між об'єктами. Це включало тестування "VRpaw'n" для перевірки плавності пересування та зручності користування меню.

Після кожного тестування проводився аналіз виявлених помилок та вносилися відповідні корективи для покращення взаємодії. Особлива увага

приділялася усуненню затримок під час підбору предметів, коректній обробці зіткнень та покращенню навігації у віртуальному офісі.

Завдяки комплексному тестуванню вдалося забезпечити стабільну та плавну взаємодію користувача з віртуальним середовищем, що робить перебування у VR-компанії більш реалістичним та комфортним.

### **3.3.3 Зміна колізій об'єктів через некоректну роботу**

Зміна колізій об'єктів через некоректну роботу була проведена після виявлення помилок під час тестування. Деякі об'єкти некоректно взаємодіяли зі середовищем: проходили крізь стіни або не відображали зіткнення. Для виправлення були змінені налаштування колізій у Unreal Engine, додано додаткові перевірки фізичних властивостей об'єктів та оптимізовано їхні параметри, щоб забезпечити реалістичну взаємодію у VR.

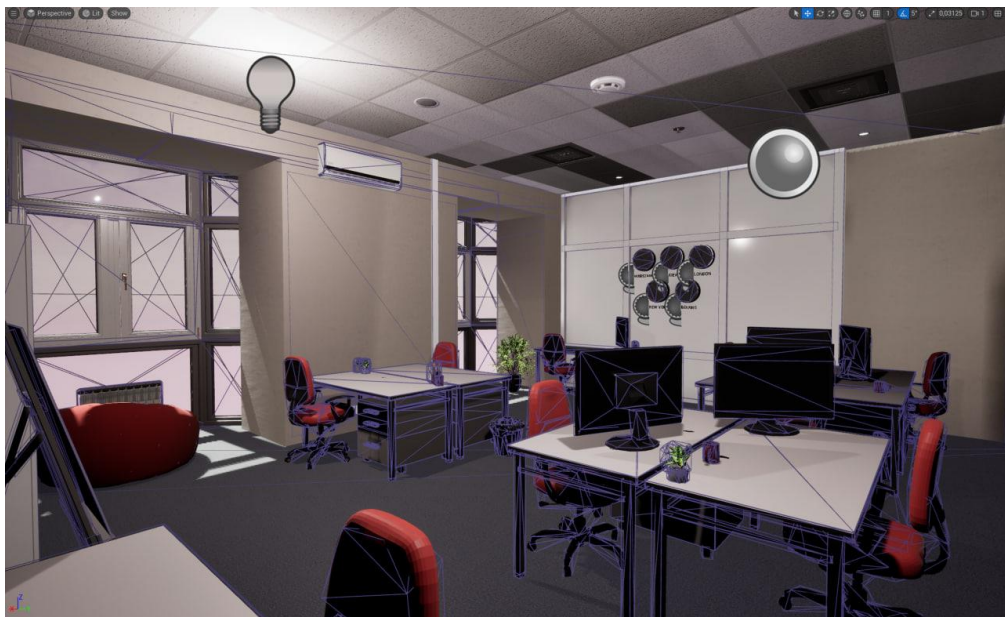


Рис 3.3 – Відображення колізії об'єктів

### **3.3.4 Створення VRrawn (модель користувача, переміщення та меню)**

Створення VRrawn є ключовим елементом віртуального офісу, оскільки саме ця модель представляє користувача у віртуальному середовищі та забезпечує його повноцінну взаємодію з об'єктами. VRrawn було реалізовано на базі Unreal Engine із використанням компонентів для підтримки VR-контролерів, що дозволяють користувачу вільно переміщатися офісом, підбирати предмети та використовувати меню управління.

Основними компонентами VRrawn є:

- Головний контролер (Headset Display) — відповідає за відображення зору користувача та його орієнтацію у просторі;
- Ручні контролери (Hand Controllers) — дозволяють маніпулювати об'єктами, підбирати предмети та взаємодіяти з інтерфейсом;
- Система переміщення (Teleportation System) — реалізує механіку переміщення у просторі без втрати орієнтації.

Окрім базових можливостей, у VRrawn було додано інтерактивне меню, яке відкривається за допомогою VR-контролерів. У цьому меню користувач може:

- перезавантажити рівень;
- перезавантажити орієнтацію у просторі;
- вийти з програми.

Модель VRrawn дозволяє повністю зануритися у віртуальну реальність, забезпечуючи плавне переміщення, зручне управління та можливість взаємодії з об'єктами віртуального офісу.

## **3.4 Висновок до Розділу 3**

У цьому розділі було детально описано етапи розробки програмного забезпечення для віртуального офісу ІТ-компанії в енергетиці. Було реалізовано архітектуру віртуального простору, створено 3D-моделі, виконано текстуровання та інтеграцію у VR-середовище. Особливу увагу приділено побудові логіки

взаємодії користувача з об'єктами середовища, включно з можливістю підбирати предмети, тестуванням таких дій та налаштуванням коректної роботи колізій. Окремо розглянуто створення VRpawp – об'єкта, який відповідає за управління переміщенням користувача у віртуальному просторі та роботу меню. Усі ці кроки дозволили створити функціональний та інтерактивний VR-досвід, що відповідає поставленим завданням і забезпечує зручну взаємодію з віртуальним офісом.

## **4 РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ**

Віртуальний офіс передбачає два основні варіанти використання: у режимі VR за допомогою шолома віртуальної реальності та в режимі класичного 3D-проглядання на екрані монітора. Обидва режими забезпечують користувачу доступ до інтерактивних елементів, взаємодію з об'єктами та можливість переміщення по офісному простору.

### **4.1 Використання в режимі VR**

Використання у VR-режимі є основним і найцікавішим способом взаємодії з віртуальним офісом, який дозволяє користувачу повністю зануритися у віртуальне середовище та відчувати ефект присутності. Для цього необхідно мати підключений VR-шолом, сумісний з Unreal Engine, наприклад, Meta Quest, HTC Vive або Valve Index. Після запуску додатку користувач потрапляє у віртуальний простір офісу, де може вільно пересуватися та взаємодіяти з об'єктами.

Після підключення шолома віртуальної реальності та запуску програми, користувач автоматично переходить у віртуальне середовище. В цей момент VR-шолом починає відстежувати положення голови користувача та контролерів, що дозволяє бачити віртуальний офіс під різними кутами в залежності від рухів голови, а також здійснювати маніпуляції з об'єктами за допомогою рук.

Для навігації у віртуальному офісі реалізована технологія телепортації, що є зручним способом переміщення у VR-середовищі. Користувач направляє контролер на бажану точку переміщення, натискає кнопку — і одразу опиняється у вказаному місці. Це дозволяє легко обходити меблі, стіни та інші об'єкти, не відчуючи дискомфорту. Такий метод забезпечує плавність рухів і запобігає можливим випадкам захитування або втрати орієнтації, що іноді трапляється при класичному способі пересування у VR. На рисунку 4.1 можна побачити приклад інтерфейсу користувача під час переміщення.



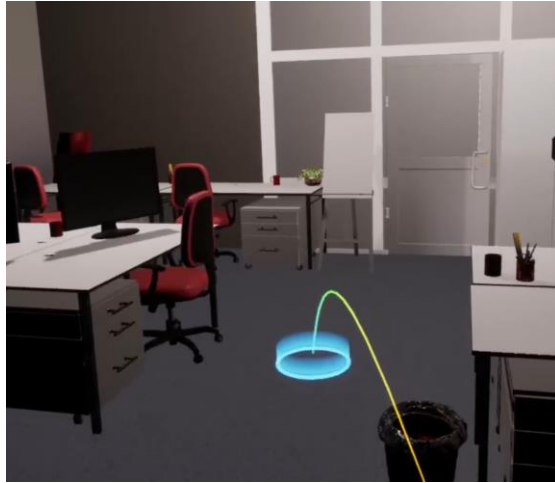


Рис 4.1 – Відображення переміщення користувача

Однією з ключових особливостей віртуального офісу є можливість підбирати та переміщувати предмети. За допомогою контролерів користувач може:

- піднімати об'єкти з поверхонь;
- переносити їх у інші місця;
- кидати або складати на полиці;

Кожен об'єкт у віртуальному офісі має налаштовану фізичну модель і колізії, що забезпечує його правильну поведінку у просторі. Наприклад, якщо користувач кидає предмет, він буде падати на підлогу відповідно до законів фізики.

На рисунку 4.2 зображено приклад підбирання предметів.

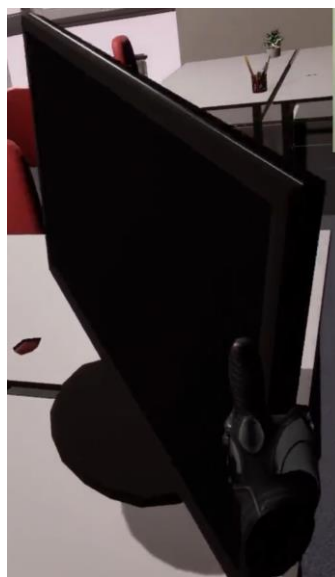


Рис 4.2 – Підбирання предметів

Віртуальний офіс оснащений VR-меню, яке дозволяє користувачу здійснювати різні дії без необхідності виходу з VR-режиму. Меню відкривається натисканням кнопки на контролері та дозволяє:

- перезавантажити рівень;
- перезавантажити орієнтацію у просторі;
- вийти з програми.

Меню побудоване у вигляді інтерактивної панелі, що з'являється перед користувачем у зручному для доступу положенні. Усі елементи управління зрозумілі та мають візуальний зворотній зв'язок (підсвічування при наведенні), що спрощує навігацію у VR. На рисунку 4.3 зображено меню користувача.



Рис 4.3 – Відображення меню користувача

## **4.2 Класичний режим 3D-переглядання**

У класичному режимі 3D-переглядання користувач може взаємодіяти з віртуальним офісом через екран комп'ютера. Це забезпечує огляд моделі, навігацію по кімнатах, зміну ракурсів камери та взаємодію з окремими об'єктами. Даний

режим корисний для демонстрацій, попереднього ознайомлення з віртуальним простором або тестування моделей без підключення VR-шолома.

### **4.3 Висновок до Розділу 4**

У цьому розділі було розглянуто особливості взаємодії користувача з розробленою системою у двох основних режимах: віртуальної реальності (VR) та класичного 3D-перегляду. Режим VR забезпечує глибоке занурення у віртуальне середовище та природну взаємодію з об'єктами за допомогою контролерів, що дозволяє ефективно оцінити простір і логіку роботи офісу. Класичний режим 3D-перегляду забезпечує швидкий доступ до вмісту на звичайному моніторі без додаткового обладнання, що робить систему універсальною та зручною для ознайомлення широкого кола користувачів. Обидва режими доповнюють один одного та дозволяють адаптувати продукт до різних сценаріїв використання.

## ВИСНОВКИ

У ході виконання проєкту була розроблена віртуальна модель офісу ІТ-компанії в енергетичній сфері з використанням сучасних технологій віртуальної реальності. Реалізація проєкту включала створення архітектури віртуального середовища, моделювання 3D-об'єктів у Blender, їх інтеграцію в Unreal Engine 5.5.4, а також налаштування фізичних взаємодій та інтерактивних елементів.

Розроблений віртуальний офіс підтримує два режими роботи: VR-режим для повного занурення у віртуальний простір та режим 3D-проглядавання на моніторі для зручної навігації. У VR-режимі користувач має можливість взаємодіяти з об'єктами, переміщувати їх, використовувати інтерактивне меню, а також вільно переміщатися по віртуальному середовищу за допомогою технології телепортації.

Проведене тестування підтвердило коректну роботу інтерактивних елементів, фізичної моделі об'єктів та забезпечення плавної взаємодії користувача з віртуальним середовищем. Була також виконана оптимізація колізій для підвищення реалістичності фізичних властивостей об'єктів.

Результатом роботи стала реалістична віртуальна модель офісу, яка може бути використана для навчання, презентацій, віртуальних турів та симуляцій робочих процесів. Впровадження таких технологій відкриває нові можливості для дистанційної роботи, тренувань та оптимізації бізнес-процесів у сфері енергетики та ІТ.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unreal Engine Documentation. [Електронний ресурс]. URL: <https://docs.unrealengine.com> (Дата звернення: 01.05.2025)
2. Blender Documentation [Електронний ресурс]. URL: <https://docs.blender.org> (Дата звернення: 24.04.2025)
3. Epic Games Marketplace [Електронний ресурс]. URL: <https://www.unrealengine.com/marketplace> (Дата звернення: 30.04.2025)
4. Sketchfab [Електронний ресурс]. URL: <https://sketchfab.com> (Дата звернення: 30.04.2025)
5. Quixel Megascans [Електронний ресурс]. URL: <https://quixel.com/megascans> (Дата звернення: 30.04.2025)
6. Collision Response Reference [Електронний ресурс]. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/collision-response-reference-in-unreal-engine> (Дата звернення: 30.04.2025)
7. Simple versus Complex Collision [Електронний ресурс]. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/simple-versus-complex-collision-in-unreal-engine> (Дата звернення: 30.04.2025)
8. Comparative Review: Blender vs. Maya vs. 3ds Max [Електронний ресурс]. URL: <https://irendering.net/comparative-review-blender-vs-maya-vs-3ds-max/#:~:text=> (Дата звернення: 25.04.2025)
9. Importing & Exporting Files [Електронний ресурс]. URL: [https://docs.blender.org/manual/en/2.93/files/import\\_export.html#:~:text=Sometimes%20you%20may%20want%20to,used%20to%20import%20and%20export](https://docs.blender.org/manual/en/2.93/files/import_export.html#:~:text=Sometimes%20you%20may%20want%20to,used%20to%20import%20and%20export) (Дата звернення: 25.04.2025)
10. Вступ – Introduction [Електронний ресурс]. URL: [https://docs.blender.org/manual/uk/2.82/getting\\_started/about/introduction.html#:~:text=,%D1%82%D0%B8%D0%BF%D1%96%D0%B2%20%D1%81%D0%B8%D0%](https://docs.blender.org/manual/uk/2.82/getting_started/about/introduction.html#:~:text=,%D1%82%D0%B8%D0%BF%D1%96%D0%B2%20%D1%81%D0%B8%D0%)

[BC%D1%83%D0%BB%D1%8F%D1%86%D1%96%D0%B9%20%E2%80%93%20%2037](https://program-ace.com/blog/unreal-engine-blueprints-vs-c/#:~:text=In%202025%2C%20the%20software%20development,notch%20apps) (Дата звернення: 25.04.2025)

11. Unreal Engine Blueprints vs. C++ [Електронний ресурс]. URL: <https://program-ace.com/blog/unreal-engine-blueprints-vs-c/#:~:text=In%202025%2C%20the%20software%20development,notch%20apps> (Дата звернення: 30.04.2025)

12. Unity vs Unreal Engine: Pros and Cons [Електронний ресурс]. URL: <https://kevrugames.com/blog/unity-vs-unreal-engine-pros-and-cons/#:~:text=The%20main%20feature%20of%20Unreal,by%20knowing%20and%20using%20C> (Дата звернення: 30.04.2025)

13. Learning Unreal: Creating Custom Actors and Components in C++ [Електронний ресурс]. URL: <https://medium.com/@lemapp09/learning-unreal-creating-custom-actors-and-components-in-c-dd87e04c70f1> (Дата звернення: 30.04.2025)

# ДОДАТОК А

Віртуальний світ ІТ-компанії в енергетиці

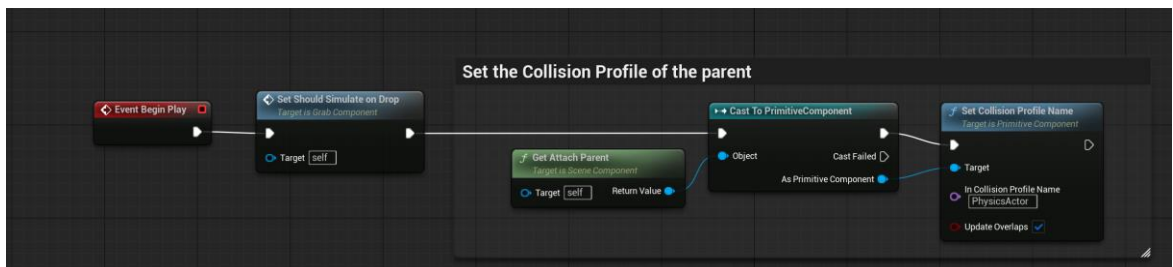
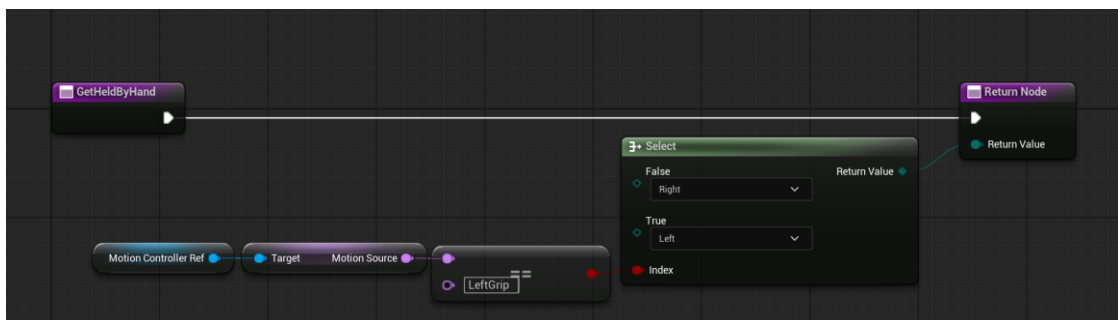
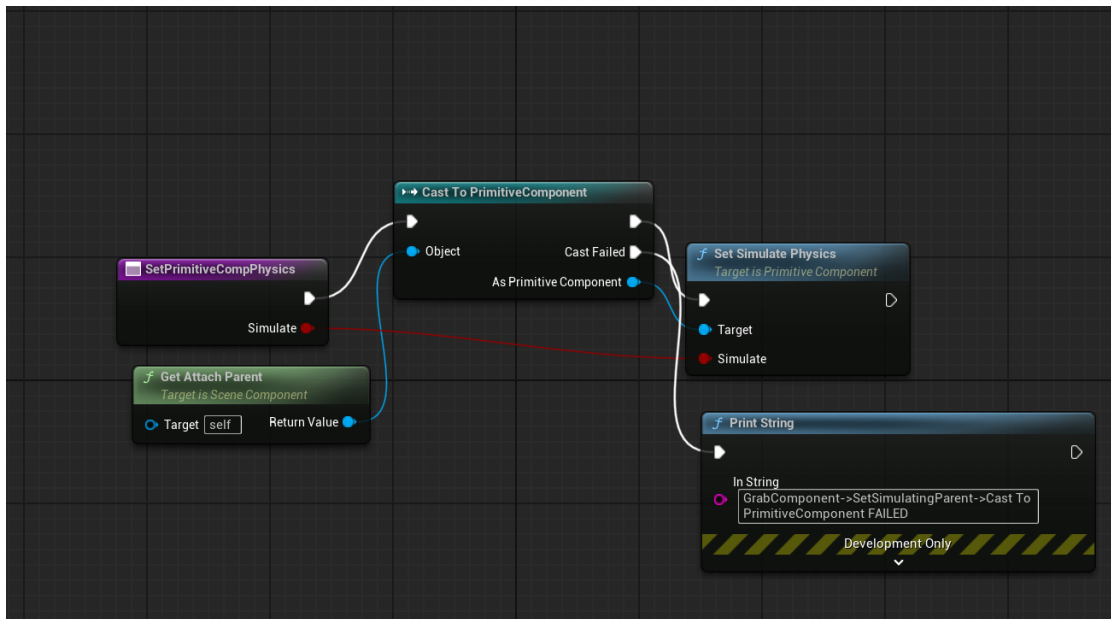
Елемент GrabComponent

Аркушів 2

Київ – 2025







**GRAPHS**

- EventGraph

**FUNCTIONS (2 OVERRIDABLE)** Override

- TryGrab
- TryRelease
- SetShouldSimulateOnDrop
- SetPrimitiveCompPhysics
- GetHeldByHand

**MACROS**

- AttachParentToMotionController

**VARIABLES**

- Components**
  - bIsHeld: Boolean
  - PrimaryGrabRelativeRotation: Rotator
  - bSimulateOnDrop: Boolean
  - GrabType: Grab Type
  - OnGrabHapticEffect: Haptic Feedback

**EVENT DISPATCHERS**

- OnGrabbed
- OnDropped

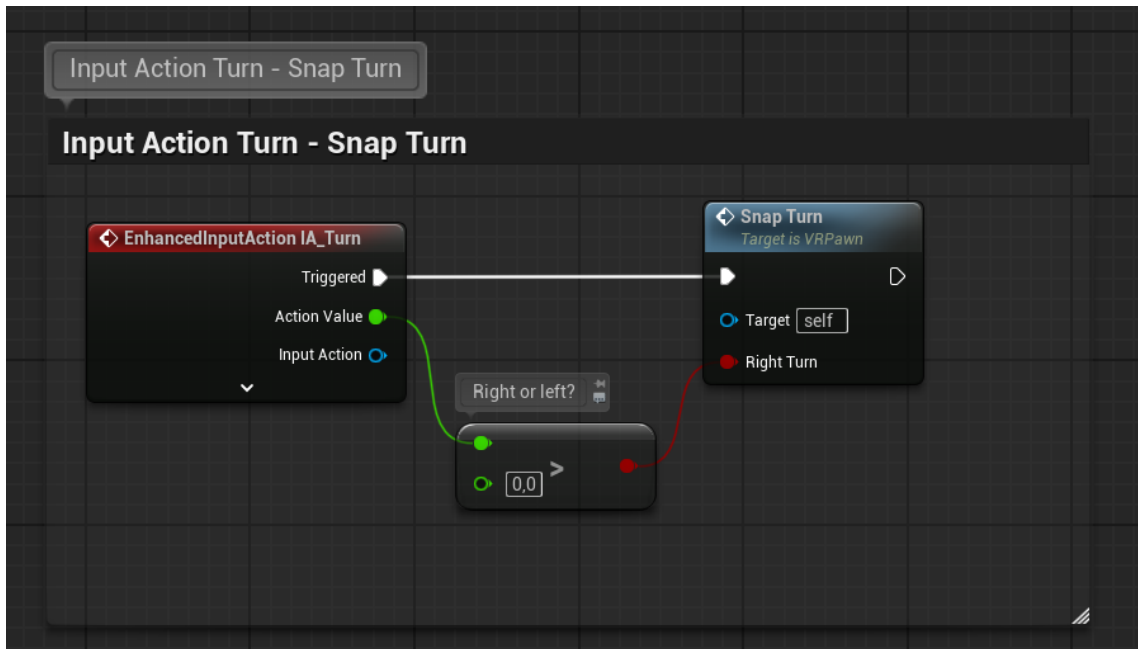
## ДОДАТОК Б

Віртуальний світ ІТ-компанії в енергетиці

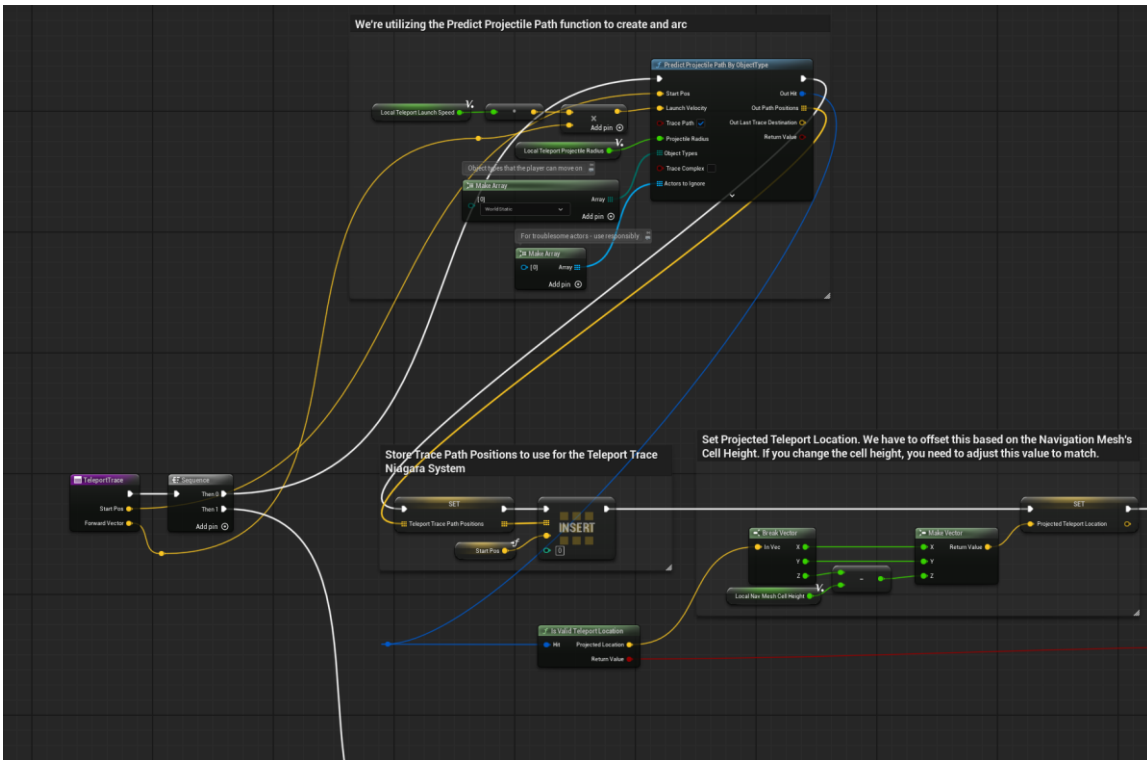
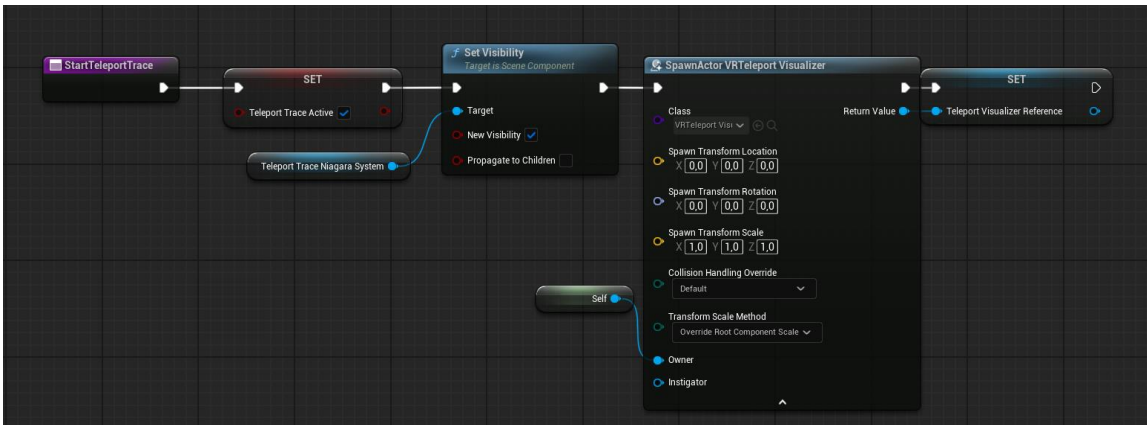
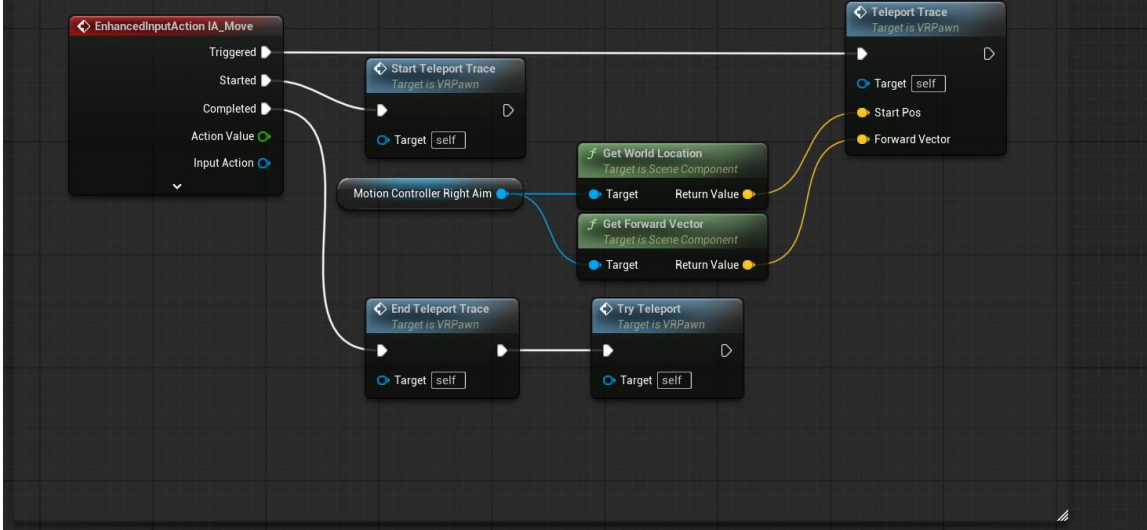
Елемент VRPawn

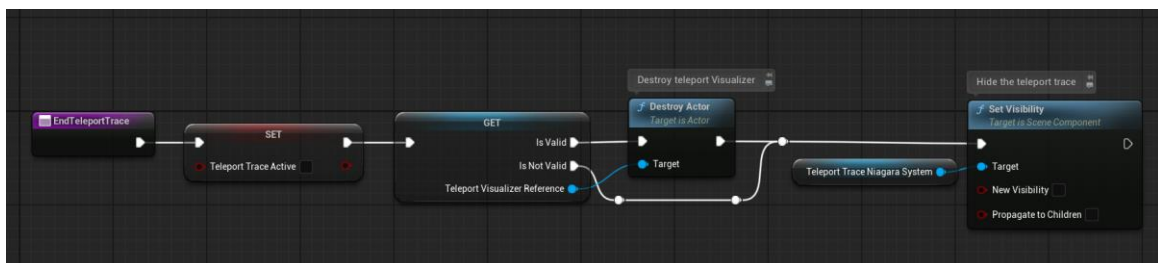
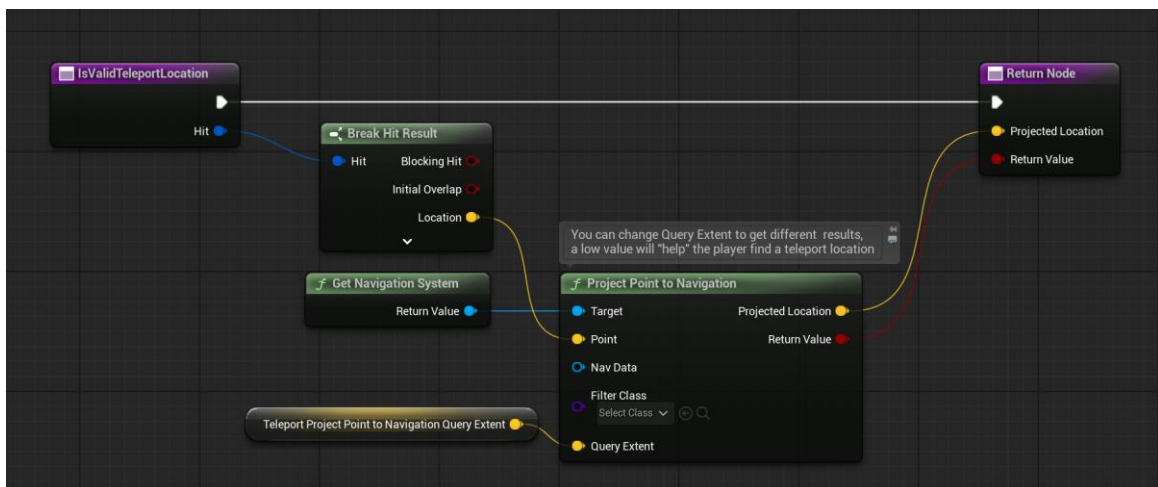
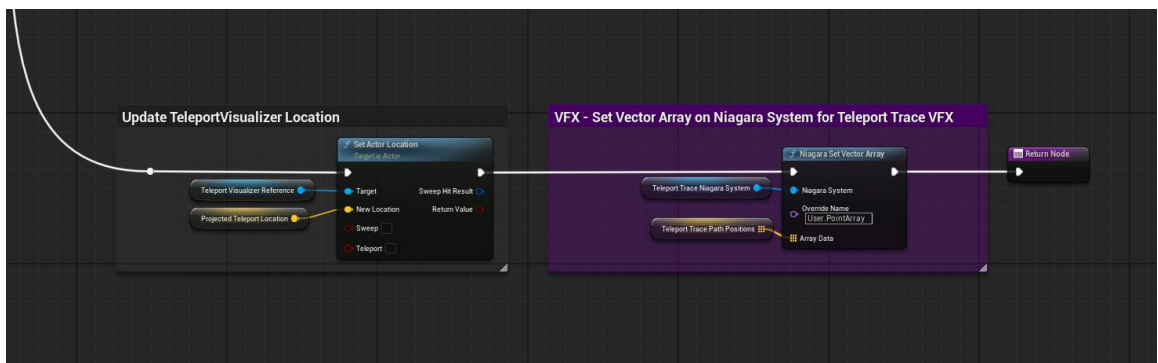
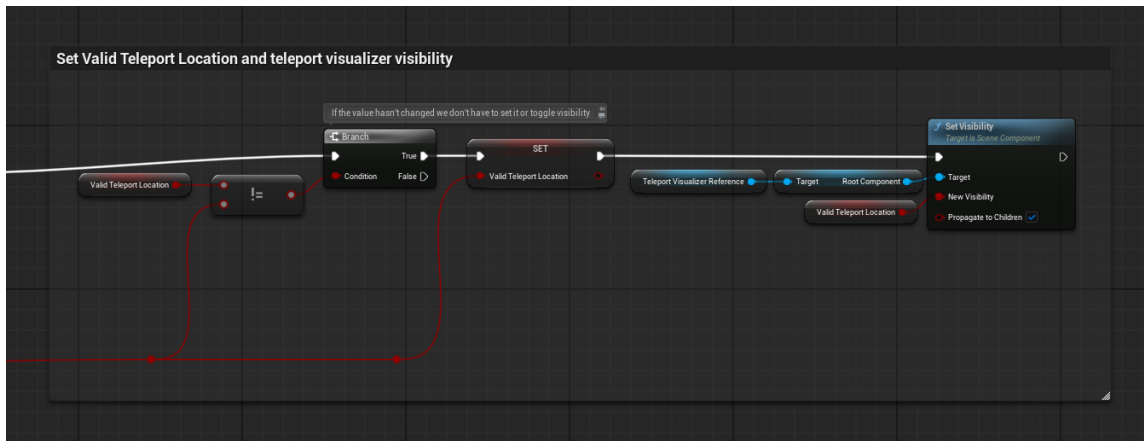
Аркушів 5

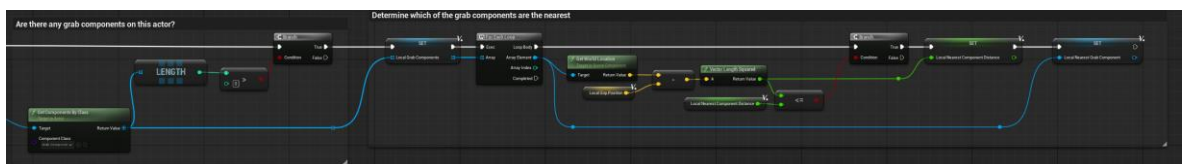
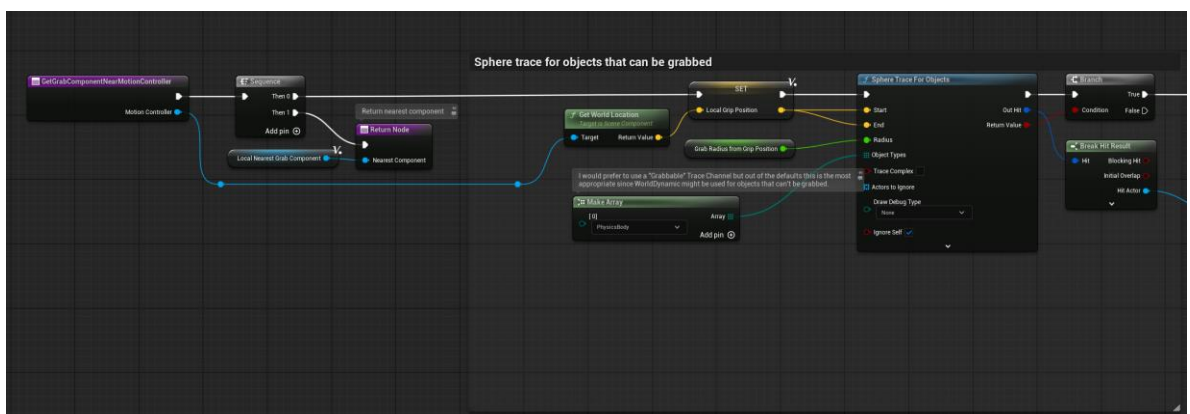
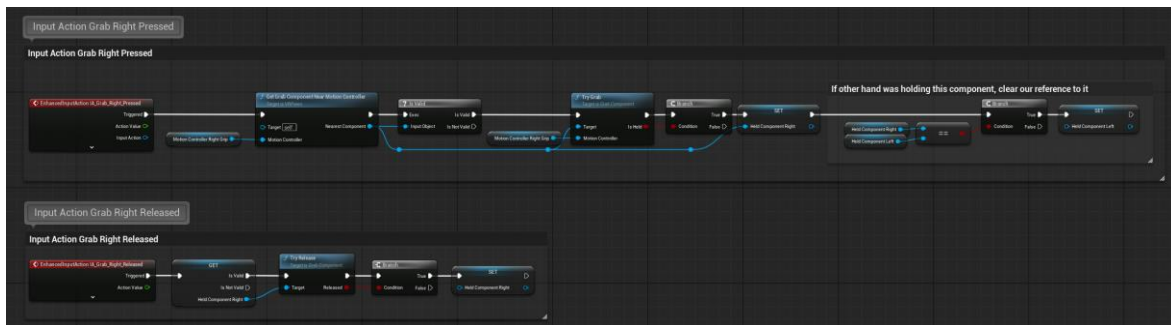
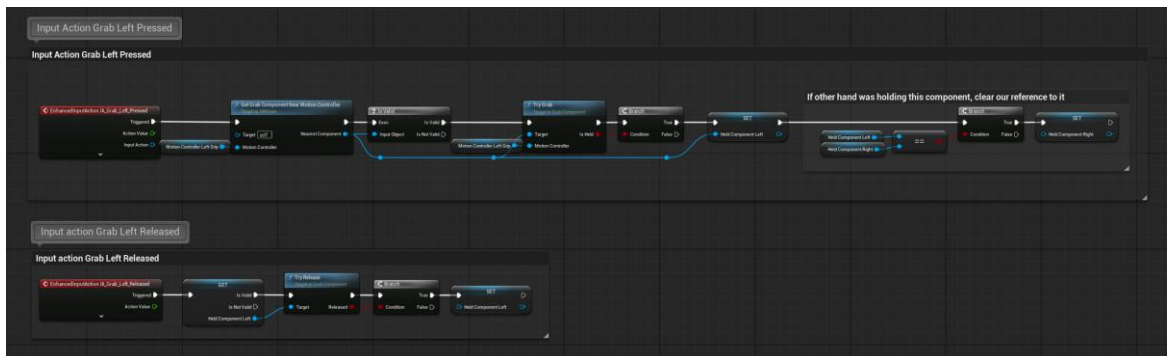
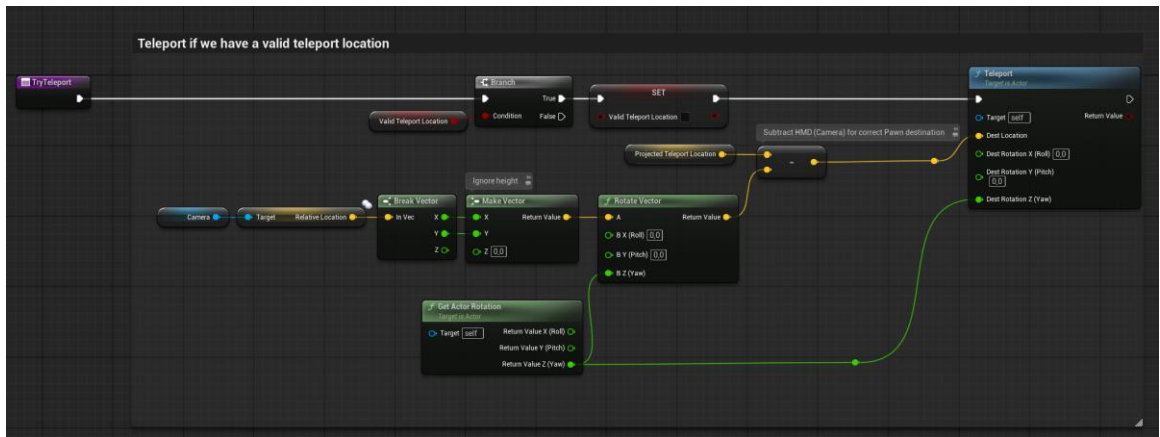
Київ – 2025

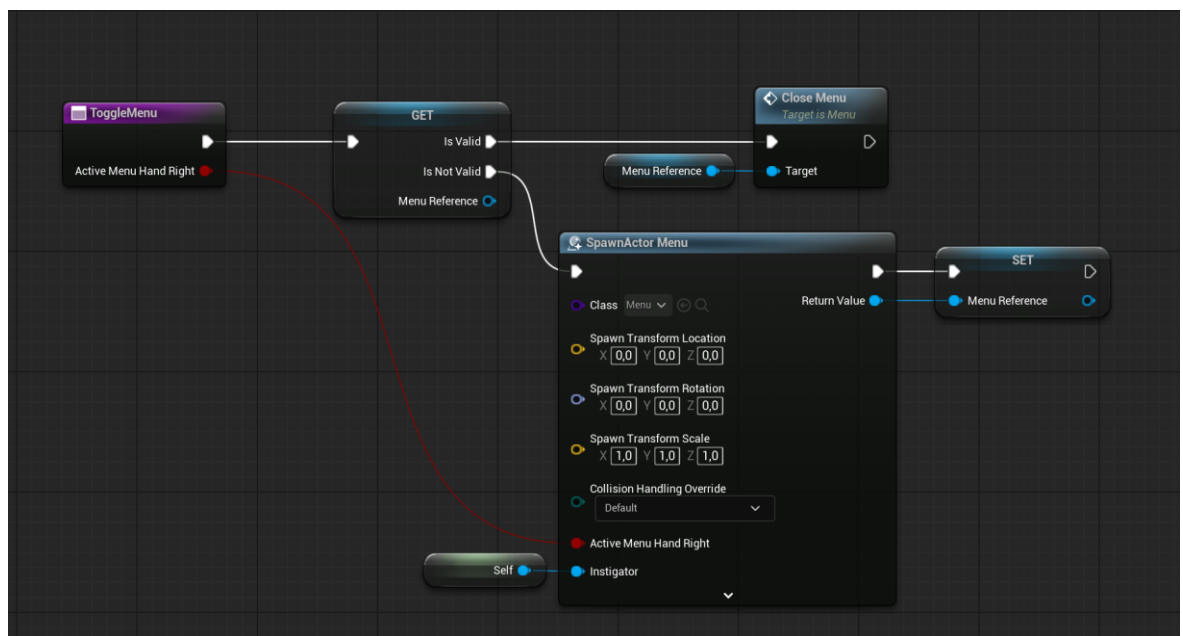
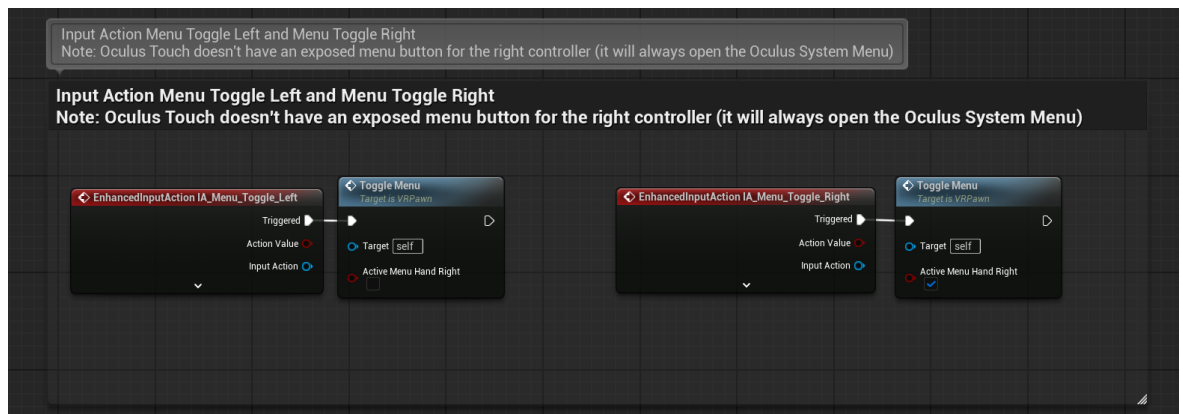


### Input Action Move - Teleport









## **ДОДАТОК В**

Віртуальний світ ІТ-компанії в енергетиці

Презентація

Аркушів 5

Київ – 2025





Навчально-науковий інститут атомної та теплової енергетики  
Кафедра інженерії програмного забезпечення в енергетиці

## Віртуальний світ IT-компанії в енергетиці

виконав: студент групи ТВ-11 Каленський Олександр Миколайович

керівник: доцент Залевська Ольга Валеріївна, кандидат технічних наук

2025

### Розгляд аналогу

Основні недоліки аналогу:

- Низький рівень деталізації 3D моделей, що дає лише уяву про актуальний стан робочого середовища.
- Відсутність будь-якої інтерактивності, що напряму впливає на зосередженість користувача на поданій інформації.
- Відсутність можливості перегляду у віртуальному середовищі



### Актуальність та мета

**Метою** є створення інтерактивної 3D-моделі офісу IT компанії в галузі енергетики з можливістю перегляду та взаємодії користувача з предметами віртуальної реальності для підвищення зацікавленості аудиторії енергетичною галуззю.

**Об'єктом дослідження** є моделювання офісної інфраструктури IT-компанії у VR-просторі.

**Предметом дослідження** є алгоритми та методи побудови VR-просторів, 3D-моделей та налаштування інтерактивної взаємодії користувача з віртуальним середовищем.



## Постановка задач

- Розробити програмне забезпечення, де буде реалізовано реалістичну 3D-модель офісу з робочими зонами
- Реалізувати можливість адаптованого до навколишнього середовища переміщення користувача у віртуальному просторі
- Розробити та адаптувати 3D-моделі та реалізувати взаємодію користувача з ними у віртуальному просторі
- Адаптувати систему колізій для обмеження фізичного проникнення через стіни та меблі віртуального середовища



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

4/14

## Засоби розробки

Ігровий двигун Unreal Engine 5.5.4

Програмний пакет для створення комп'ютерної графіки Blender

Плагін для Blender, що надає доступ до великої бібліотеки 3D-моделей [BlenderKit](#)

Інструмент для імпорту високоякісних Megascans-ресурсів в Unreal Engine [QuixelBridge](#)



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

5/14

## Реалізація переміщення користувача



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

6/14

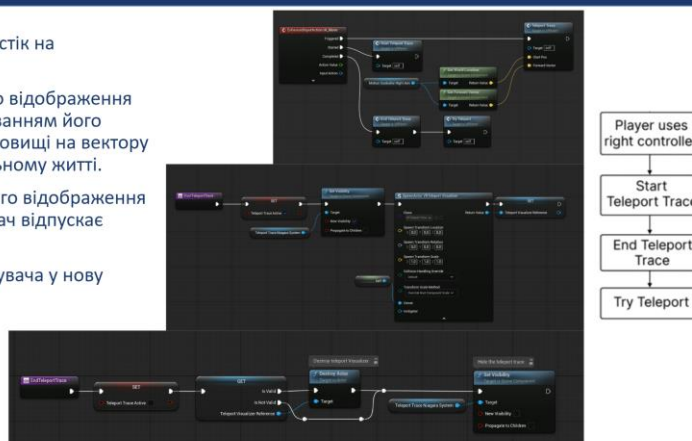
## Реалізація переміщення користувача

Крок 1. Користувач відхилив стік на контроллері в правій руці.

Крок 2. Побудова візуального відображення переміщення гравця з урахуванням його позиції у віртуальному середовищі на вектору правого контроллера в реальному житті.

Крок 3. Вимкнення візуального відображення переміщення, коли користувач відпускає раніше використаний стік.

Крок 4. Переміщення користувача у нову позицію, якщо можливо.

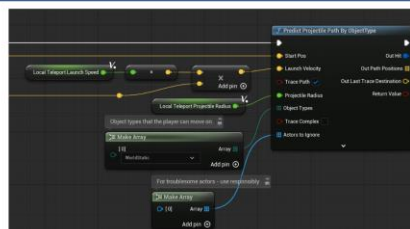


Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

7/14

## Реалізація переміщення користувача

Побудова візуалізації переміщення користувача відбувається за допомогою вбудованої функції Predict Projectile Path By ObjectType, що прораховує дорогу об'єкта по якому він має рухатися.



Переміщення відбувається у випадку, коли було обрано коректну позицію та з урахуванням напрямлення камери користувача до переміщення та координат нової позиції. Телепортація реалізована вбудованою функцією Teleport.



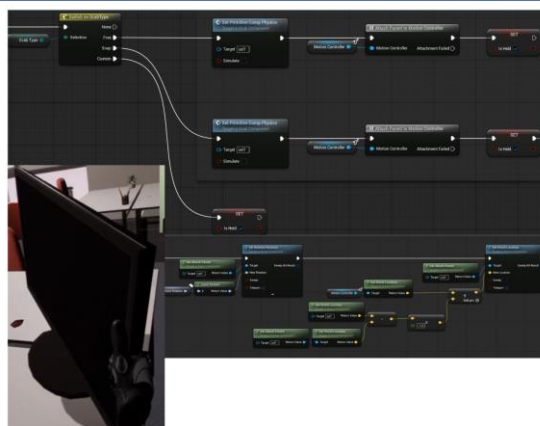
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

8/14

## Реалізація підбору предметів

Крок 1. Користувач натиснув кнопку Grab на контроллері в момент, коли рука аватара у віртуальному середовищі знаходилась біля моделі з елементом GrabComponent.

Крок 2. Об'єкт отримує значення IsHeld. Передаються дані для переміщення предмету до відповідної руки з урахуванням актуальної позиції.

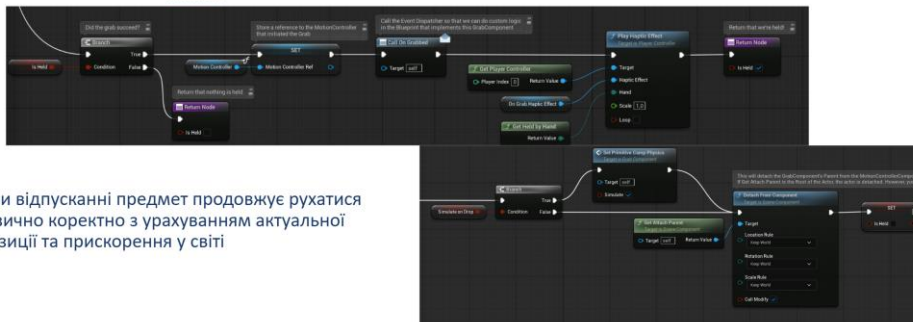


Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

9/14

## Реалізація підбору предметів

Крок 3. Якщо об'єкт має значення IsHeld, то переміщується до руки віртуального аватара. Користувач отримує гаптич віддачу від відповідного контроллера.



При відпусканні предмет продовжує рухатися фізично коректно з урахуванням актуальної позиції та прискорення у світі

## Меню користувача

Користувач має наступні функції:

- Відновлення орієнтації камери
- Перезавантаження рівня
- Вихід з програми



## Колізія об'єктів

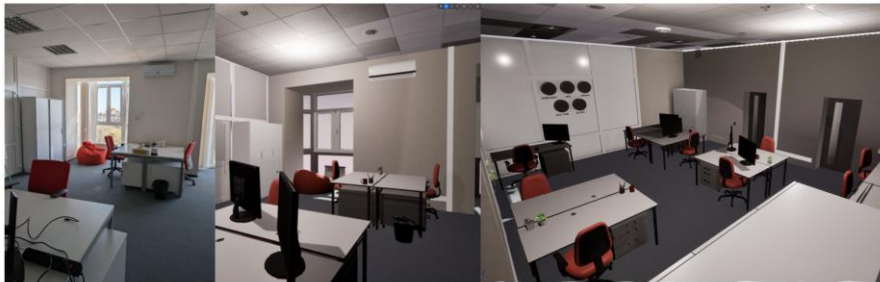
При тестуванні додатку було виявлено проблеми з колізіями усіх 3D-об'єктів, оскільки вони були створені двигуном автоматично.

Коректна колізія об'єктів необхідна для коректної фізичної взаємодії об'єктів у віртуальному середовищі.





## Режим класичного 3D-перегляду



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

13/14

## Висновки

- Обрано та обґрунтовано використання інструментів Unreal Engine 5 і Blender
- Розроблено 3D-модель офісного простору з інтерактивними об'єктами
- Реалізовано логіку переміщення користувача та взаємодії з об'єктами у VR-середовищі
- Проведено тестування і налагодження колізій та навігації
- Створено два режими перегляду: у VR-шоломі та у звичайному 3D-режимі



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

14/14



Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

National Technical University of Ukraine  
"Igor Sikorsky Kyiv Polytechnic Institute"