

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та  
робототехніки

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

(тема)

КТ 1. Розроблення комп'ютерної моделі лабораторій кафедри КІТАР на основі  
методів віртуальної реальності. ПТ 1. Моделювання обладнання лабораторій

Виконав: здобувач 2 року навчання, гр. КІТПВм-23-1

Скляр М. В.

(прізвище, ініціали)

Спеціальність

174 Автоматизація, комп'ютерно-інтегровані технології та  
робототехніка

освітньої програми Комп'ютерно-інтегровані

технологічні процеси і виробництва

(код і повна назва напрямку)

Тип програми

освітньо-професійна

(повна назва освітньої програми)

Керівник

проф. Цимбал О.М.

(посада, прізвище, ініціали)

Допускається до захисту  
зав. кафедри

(підпис)

Невлюдов І.Ш.

(прізвище, ініц

2025 р.

Харківський національний університет радіоелектроніки

Факультет	<u>Автоматики і комп'ютеризованих технологій</u>
Кафедра	<u>Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки</u>
Рівень вищої освіти	<u>другий (магістерський)</u>
Спеціальність	<u>174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка</u>
Тип програми	<u>освітньо-професійна</u>
Освітня програма	<u>Комп'ютерно-інтегровані технологічні процеси і виробництва</u>

(код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Склярову Максиму Владиславовичу

(прізвище, ім'я, по батькові)

1. Тема роботи КТ 1. Розроблення комп'ютерної моделі лабораторій кафедри КІТАР на основі методів віртуальної реальності.

ПТ 1. Моделювання обладнання лабораторій

затверджена наказом по університету від \_\_\_\_\_ 22.11.2024 р. № 1231 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії \_\_\_\_\_ 31.01.2025 р.

3. Вихідні дані до роботи \_\_\_\_\_ план приміщення кафедри КІТАР, скани та фото предметів, об'єктів та приладів, інформація про їхній принцип роботи, фото та відео зовнішньої та внутрішньої частини приміщення, програмне забезпечення для моделювання, програмування та малювання, параметри реального та модельованого робота

4. Перелік питань, що потрібно опрацювати в роботі:

використання 3D-технологій для створення віртуальних лабораторій;

методи віртуальної реальності; бази даних; штучний інтелект;

аналіз літератури та успішної інтеграції схожих віртуальних середовищ;

аналіз засобів розробки та вибір технологій;

принципи моделювання, анімації, зберігання та обробки даних;

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (\*.ppt) – 16 с. формату А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Керівник (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Написання першого розділу	10.12.2024	виконано
2	Написання другого розділу	17.12.2024	виконано
3	Перший етап розробки	24.12.2024	виконано
4	Написання третього розділу	29.12.2024	виконано
5	Другий етап розробки	02.01.2025	виконано
6	Завершення розробки	06.01.2025	виконано
7	Оформлення пояснювальної записки	12.01.2025	виконано
8	Підготовка та оформлення презентації	14.01.2025	виконано

Дата видачі завдання

22.11.2024 р.

Здобувач

  
(підпис)

Скляров М.В.

( прізвище, ініціали)

Керівник роботи

(підпис)

проф. Цимбал О.М.

(посада, прізвище, ініціали)

Я, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

15 січня 2025 р.



Склярів М.В.

## РЕФЕРАТ

Пояснювальна записка: 88 с., 48 рис., 2 дод., 40 джерел.

### 3D-МОДЕЛЮВАННЯ, ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, БАЗА ДАНИХ, ШТУЧНИЙ ІНТЕЛЕКТ

Мета роботи – покращення методів створення віртуальної реальності на основі інтерактивного середовища з точки зору проведення лабораторних робіт у сфері освіти.

Об'єкт дослідження – технології віртуальної реальності.

Предмет дослідження – метод створення комп'ютерної моделі лабораторій на основі віртуальної реальності з використанням штучного інтелекту та баз даних.

Дана робота буде корисною для викладачів навчальних закладів, освітніх платформ та компаній, оскільки надає детальний огляд можливостей використання методів віртуальної реальності для вдосконалення лабораторних робіт у навчальному процесі, сприяючи підвищенню якості навчання і забезпеченню доступності для різних категорій студентів.

Крім того, дане дослідження стане корисним для компаній-розробників та розробників у навчальних закладах, оскільки його результати демонструють, як інтеграція технологій віртуальної реальності та штучного інтелекту може оптимізувати процес навчання, забезпечити автоматизацію аналізу результатів і підвищити ефективність освітніх програм у різних ситуаціях.

Пояснювальна записка виконана згідно з вимогами [1], [2].

Матеріали було висвітлено у збірнику «Цифрові інновації та сталий розвиток \ Digital innovations and sustainable development» (DI&SD2024) за темою [3] «Аналіз можливостей впровадження AI та 3D-технологій у різні галузі навчання».

## ABSTRACT

Explanatory note: 88 p., 48 fig., 2 appendix, 40 sources.

### 3D-MODELING, VIRTUAL ENVIRONMENT, DATABASE, ARTIFICIAL INTELLIGENCE

The purpose of the work is to improve methods for creating virtual reality based on an interactive environment from the point of view of conducting laboratory work in the field of education.

The object of research is virtual reality technologies.

The subject of research is a method for creating a computer model of laboratories based on virtual reality using artificial intelligence and databases.

This work will be useful for teachers of educational institutions, educational platforms and companies, as it provides a detailed overview of the possibilities of using virtual reality methods to improve laboratory work in the educational process, contributing to improving the quality of education and ensuring accessibility for different categories of students.

In addition, this study will be useful for development companies and developers in educational institutions, as its results demonstrate how the integration of virtual reality and artificial intelligence technologies can optimize the learning process, provide automation of results analysis, and increase the effectiveness of educational programs in various situations.

The explanatory note is prepared in accordance with the requirements of [1], [2].

The materials were published in the collection "Digital innovations and sustainable development" (DI&SD2024) on the topic [3] "Analysis of the possibilities of implementing AI and 3D technologies in various areas of education."

## ЗМІСТ

Перелік скорочень .....	8
Вступ.....	9
1 Аналіз технології віртуальної реальності .....	12
1.1 Аналіз базової інформації .....	12
1.2 Аналіз додаткових технологій.....	16
1.2.1 База даних.....	17
1.2.2 Штучний інтелект.....	19
1.3 Висновки до першого розділу .....	23
2 Моделювання структури та плану реалізації проєкту.....	25
2.1 Аналіз літератури та досліджень .....	25
2.2 Постановка задач .....	29
2.3 Актуальність.....	29
2.4 Завдання дослідження .....	30
2.5 Вимоги та вхідні дані.....	30
2.6 Аналіз отриманих результатів та розробка алгоритму дій.....	31
2.7 Висновки до другого розділу.....	34
3 Розробка програмної частини віртуального середовища.....	35
3.1 Аналіз засобів розробки та вибір технологій.....	35
3.2 Налаштування програмного середовища .....	41
3.3 Моделювання, ригінг та анімація.....	49
3.4 Реалізація баз даних .....	60
3.5 Реалізація систем зі застосуванням штучного інтелекту .....	68
3.6 Тестування програмної частини .....	71
3.7 Розрахункова частина.....	74
3.8 Охорона праці.....	78
3.9 Висновки до третього розділу .....	81
Висновки .....	82

Перелік джерел посилання .....	84
Додаток А Апробація результатів наукових досліджень.....	89
Додаток Б Демонстраційний матеріал .....	100

## ПЕРЕЛІК СКОРОЧЕНЬ

БЖД – безпека життєдіяльності;

БД – база даних;

КІТАР – комп’ютерно-інтегровані технології, автоматизації та робототехніки;

ПЗ – програмне забезпечення;

ТЗ – технічне завдання;

ШІ – штучний інтелект;

2D – 2-dimensional;

3D – 3-dimensional;

AI – Artificial Intelligence;

AR – Augmented Reality;

DB – Data Base;

IoT – Internet of Things;

MR – Mixed Reality;

UE – Unreal Engine;

VR – Virtual Reality.

## ВСТУП

Віртуальна реальність та штучний інтелект є ключовими технологіями, що визначають сучасний розвиток цифрового середовища. Віртуальна реальність забезпечує унікальний спосіб занурення в інтерактивний простір, створюючи ілюзію фізичної присутності в змодельованому середовищі.

Штучний інтелект, зокрема його алгоритми машинного навчання та обробки даних, надає цим технологіям можливість адаптації, прогнозування та автоматизації складних завдань.

Усі ці інновації знаходять своє ідеальне застосування у створенні віртуальних лабораторій для дистанційного навчання. Такі лабораторії дають змогу студентам та дослідникам проводити експерименти, моделювати процеси та отримувати практичні навички, навіть перебуваючи на значній відстані від фізичних лабораторій.

Використання віртуальної реальності дозволяє відтворити реалістичне середовище для навчання, а штучний інтелект забезпечує персоналізований підхід до викладання та автоматизацію оцінювання.

Крім того, штучний інтелект, у поєднанні з сучасними базами даних, створює потужний інструмент для зберігання, організації та обробки величезних обсягів інформації. Бази даних виступають фундаментом, який забезпечує не лише швидкий доступ до даних, а й їхню структуровану обробку, що є критично важливим для аналізу інформації, побудови прогнозів та обробки результатів роботи.

Завдяки таким системам можна автоматизувати перевірку завдань, проводити індивідуалізовані оцінювання та надавати рекомендації щодо подальшого навчання або збирати та аналізувати інформацію з віртуального простору, що у подальшому може повпливати на вдосконалення фізичних об'єктів чи програм навчання.

Це відкриває нові горизонти для освіти, розширюючи її можливості та роблячи її доступнішою, інтерактивнішою й ефективнішою навіть у найвіддаленіших куточках світу. У майбутньому така синергія технологій може перетворити освіту на повністю персоналізований процес, що враховує потреби, інтереси й можливості кожного студента. Проте, зростаючий потенціал такої взаємодії відкриває нові перспективи й у багатьох інших сферах. У медицині ці технології вже сьогодні сприяють революційним змінам: пацієнти можуть отримувати точні діагнози завдяки аналізу великих обсягів медичних даних, а лікарі мають можливість практикувати складні хірургічні втручання у віртуальному середовищі, яке точно імітує реальні умови. Штучний інтелект забезпечує аналіз найдрібніших деталей процедур, а база даних зберігає історію тренувань, дозволяючи лікарям відстежувати свій прогрес.

В індустрії розваг ці технології трансформують спосіб взаємодії з медіаконтентом. Віртуальна реальність дозволяє користувачам зануритися у вигадані світи, тоді як штучний інтелект динамічно адаптує сюжет та сценарії відповідно до вподобань гравця. Бази даних забезпечують зберігання та аналіз ігрових профілів, дозволяючи розробникам створювати контент, що відповідає інтересам кожного користувача.

Промисловість також отримує значні переваги. Така інтеграція дозволяє створювати моделі складних виробничих процесів, які можна тестувати у безпечному віртуальному середовищі перед впровадженням на практиці. Завдяки цьому компанії економлять ресурси, зменшують ризики помилок і вдосконалюють свої технології.

Однак із розширенням використання таких інновацій виникають і нові виклики. Забезпечення безпеки даних, конфіденційності користувачів та запобігання зловживанню технологіями стають пріоритетними завданнями.

У межах даної роботи буде здійснено розробку та впровадження підходів до використання технологій віртуальної реальності у проведенні лабораторних робіт для освітніх цілей. Особливу увагу буде приділено аналізу сучасних методів і технологій, пов'язаних із 3D-моделюванням, віртуальною реальністю,

штучним інтелектом та обробкою даних. Планується вивчення успішних прикладів застосування VR у різних сферах, дослідження можливостей створення інтерактивного навчального середовища та визначення ключових переваг і обмежень цих технологій. Також буде змодельовано структуру й план реалізації проєкту, орієнтованого на ефективне використання VR у навчальних процесах.

Отже, мета роботи – покращення методів створення віртуальної реальності на основі інтерактивного середовища з точки зору проведення лабораторних робіт у сфері освіти.

Об'єкт дослідження – технології віртуальної реальності.

Предмет дослідження – метод створення комп'ютерної моделі лабораторій на основі віртуальної реальності з використанням штучного інтелекту та баз даних.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести огляд літератури та досліджень, що стосуються використання таких технологій, як: 3D, віртуальна реальність, AI та методів зберігання та обробки даних.

- проаналізувати приклади успішного застосування методів віртуальної реальності у різних сферах діяльності та освітньому процесі;

- вивчити можливості інтеграції віртуальної реальності для створення безпечного і адаптивного середовища, яке підтримує взаємодію з лабораторним обладнанням і дозволяє студентам відпрацьовувати навички без фізичної присутності;

- визначити переваги та обмеження впровадження технологій 3D та віртуальної реальності для індивідуального та групового навчання, а також розглянути можливості вдосконалення цих технологій у навчальних середовищах за допомогою аналізу інформації з бази даних за допомогою штучного інтелекту (ШІ);

- провести моделювання структури та плану реалізації проєкту на основі проаналізованої інформації.

# 1 АНАЛІЗ ТЕХНОЛОГІЇ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

## 1.1 Аналіз базової інформації

Віртуальна реальність [4] – це технологія, яка дозволяє користувачам взаємодіяти з комп'ютерно змодельованим середовищем, що виглядає і відчувається, як реальний світ. Завдяки 3D-графіці та спеціальному програмному забезпеченню, VR створює ілюзію присутності, яка дозволяє людині «перенестися» в інше місце та отримати новий досвід, відмінний від звичайної роботи за комп'ютером. Це може бути симуляція реальних об'єктів, процесів або навіть абстрактних просторів, які використовуються в освіті, науці, індустрії розваг, підприємствах та інших сферах.

Окрім віртуальної реальності (VR), яка повністю переносить користувача в цифровий світ, важливими технологіями є також доповнена реальність (AR) і змішана реальність (MR), які вже активно застосовуються в освіті та інших сферах закордоном (рис. 1.1).



Рисунок 1.1 – Різновиди технологій віртуальної реальності [5]

Важливо зазначити, що для роботи з VR не обов'язково потрібні спеціальні шоломи чи контролери. У багатьох випадках достатньо звичайного персонального комп'ютера (ПК) або ноутбука, що дозволяє користувачам зануритися у VR-середовище через екран монітора. Це відкриває можливості для тих, хто не має доступу до дорогого обладнання, але все одно бажає отримати досвід навчання у віртуальному просторі. За допомогою клавіатури та миші або сенсорного екрану користувач може досліджувати середовище, взаємодіяти з об'єктами, вирішувати завдання та виконувати навчальні вправи.

Доповнена реальність (AR) дозволяє накладати цифрові об'єкти на реальне середовище [6]. Це означає, що через камеру смартфона, планшета чи інших пристроїв користувач може бачити віртуальні об'єкти, які інтегровані з фізичним світом. Наприклад, студент може використовувати AR-додаток, щоб досліджувати 3D-модель пристрою, яка віртуально «поміщена» у реальну аудиторію. AR допомагає доповнити навчальний процес візуалізацією об'єктів або процесів, які не завжди можна показати в реальності за різних причин. Це робить навчання більш інтерактивним і доступним, оскільки для AR потрібен лише стандартний пристрій з камерою та екраном.

Змішана реальність (MR) об'єднує елементи VR та AR, дозволяючи одночасно взаємодіяти як з віртуальними, так і з реальними об'єктами [7]. У змішаній реальності віртуальні об'єкти можуть не просто накладатися на реальність, а й реагувати на неї в реальному часі, що створює інтерактивне середовище, де фізичний і цифровий світи зливаються. MR вимагає більш просунутого обладнання, як, наприклад, окуляри Ray-Ban від Meta, що дозволяють, скажімо, інтегрувати цифрове лабораторне обладнання прямо в навчальне середовище. Це особливо корисно в інженерії, де студент може бачити віртуальні компоненти в реальному масштабі та здійснювати маніпуляції, які будуть точно інтегровані у фізичний простір.

Одним із перспективних напрямків використання VR є створення віртуальних лабораторій, де можна відтворювати навчальні експерименти та симуляції різних процесів. Віртуальна лабораторія дозволяє студентам

виконувати лабораторні роботи в безпечному, керованому середовищі, яке точно відображає реальні умови. Це може бути фізична лабораторія, що моделює закони механіки, теплопередачі та інші природні явища чи демонструє можливості технічного обладнання.

У такому середовищі студенти можуть отримати практичні навички, відпрацьовувати різні сценарії та одразу бачити результати своїх дій, що значно посилює ефективність навчання.

Віртуальні лабораторії надають можливість експериментувати з ресурсомісткими, складними або ризикованими процесами, які важко або дорого відтворити у реальних умовах. Технологія віртуальної реальності робить навчальний процес інтерактивним, зручним та доступним для різних груп користувачів, надаючи нові можливості для вдосконалення освітнього досвіду.

Технології віртуальної (VR), доповненої (AR) та змішаної реальностей (MR) можуть бути ще більш потужними й ефективними для навчання та досліджень завдяки інтеграції з Інтернетом речей (Internet of Things, IoT).

IoT – це мережа підключених пристроїв, сенсорів і об'єктів, які здатні збирати й обмінюватися даними в реальному часі [8]. Поєднання VR, AR, MR з IoT відкриває нові можливості для створення динамічних навчальних середовищ, де дані з реального світу активно впливають на поведінку та характеристики віртуальних об'єктів.

У VR-лабораторіях IoT може підключати реальні пристрої та сенсори, щоб відтворювати показники в реальному часі. Наприклад, у VR-середовищі для навчання інженерів може бути змодельована установка, дані якої оновлюються завдяки сенсорам на реальному обладнанні. Це дозволяє студентам взаємодіяти з реальними даними і бачити, як їхні дії віртуально впливають на систему, що моделює поведінку справжніх об'єктів. Такий підхід корисний і для моделювання критичних ситуацій, як-от збоїв у роботі обладнання, які важко і небезпечно імітувати в реальних умовах.

Доповнена реальність у поєднанні з IoT дозволяє «накладати» на реальний світ інформацію з підключених сенсорів і пристроїв. Наприклад, AR-

додаток може відображати показники температури, тиску, вологості та інших параметрів безпосередньо на екрані користувача в реальному часі. Це особливо корисно, коли потрібно отримувати оперативні дані про стан обладнання під час роботи. У лабораторіях AR може візуально демонструвати зміни параметрів віртуальних моделей об'єктів, допомагаючи студентам розуміти вплив змінних факторів на процес.

Змішана реальність разом з IoT створює середовище [9], де студенти можуть взаємодіяти з віртуальними та реальними об'єктами одночасно, з динамічним оновленням параметрів. Наприклад, у навчальних лабораторіях MR-додатки можуть відображати цифрові моделі машин або приладів, що реагують на дії користувача, а також на показники сенсорів з реальних пристроїв, інтегрованих у систему. Це дозволяє студентам проводити складні тести та симуляції, у яких вони одночасно бачать і аналізують як реальні дані, так і віртуальні сценарії. Таким чином, студенти отримують практичний досвід роботи з високотехнологічними пристроями й одночасно відпрацьовують навички контролю й аналізу.

Поєднання цих технологій з IoT робить навчальний процес набагато гнучкішим, дозволяючи студентам працювати з реальними даними й оволодівати практичними навичками в умовах, максимально наближених до реальних. Віртуальні середовища стають адаптивними: студенти бачать, як зміни у фізичних параметрах відображаються в симуляціях, і вчать розуміти реальні процеси через віртуальні дії.

Ця інтеграція також дозволяє створювати індивідуальні або групові симуляції з реальними сценаріями, що стимулюють розвиток критичного мислення, уміння працювати з великою кількістю інформації та приймати оперативні рішення на основі аналізу даних.

## 1.2 Аналіз додаткових технологій

Поєднання віртуальної реальності (VR), штучного інтелекту (AI) та баз даних (DB) створює потужну екосистему для розробки інтерактивних та адаптивних технологічних рішень та середовищ (рис. 1.2).

Кожна з цих технологій відіграє ключову роль у взаємодії: VR забезпечує інтерактивне середовище, що імітує реальний чи вигаданий простір; AI аналізує поведінку користувачів, пропонує персоналізовані рішення та оптимізує процеси у віртуальному середовищі; бази даних зберігають і організують величезні обсяги інформації, необхідної для роботи систем, які також можуть бути проаналізовані за допомогою AI.

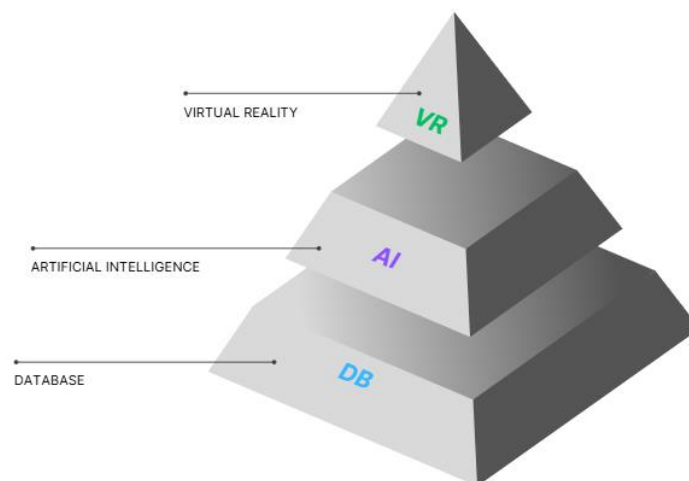


Рисунок 1.2 – Синергія трьох технологій

Проте інтеграція цих технологій має виклики, такі як висока потреба в обчислювальній потужності для обробки великих обсягів даних у реальному часі, забезпечення сумісності між компонентами системи та захист персональних даних. Незважаючи на це, майбутнє взаємодії VR, AI та DB виглядає багатообіцяючим. Воно передбачає створення автономних VR-середовищ, що адаптуються до користувача в реальному часі, інтеграцію з технологіями

доповненої реальності (AR) та вдосконалення механізмів персоналізації. Така синергія вже сьогодні змінює підходи до взаємодії людини з цифровими технологіями, формуючи нові стандарти у багатьох сферах життя.

Ще один перспективний напрям – розвиток інклюзивних технологій. Наприклад, у VR можна створювати середовища, які враховують особливі потреби людей з обмеженими можливостями. AI допомагатиме адаптувати віртуальний простір до фізичних чи когнітивних можливостей користувачів, а бази даних міститимуть інформацію для забезпечення зручного й безпечного користування системою. Таким чином, ці технології сприятимуть соціальній інтеграції та рівному доступу до сучасних можливостей.

### 1.2.1 База даних

Однією з найважливіших складових системи віртуальних лабораторій є база даних, яка дозволяє детально відстежувати прогрес студентів і надавати корисну аналітику для викладачів. Така система є не просто засобом фіксації результатів, а багатофункціональною платформою, що збирає та аналізує дані про діяльність студентів під час роботи у віртуальному середовищі [10]. Це забезпечує глибше розуміння того, як студенти навчаються, на чому зосереджують увагу, з якими об'єктами взаємодіють частіше та які аспекти викликають найбільші труднощі.

База даних надає можливість точно відстежувати, до яких об'єктів студент підходить частіше та з якими об'єктами він взаємодіє найбільше. Це дозволяє визначити, які матеріали чи експерименти викликають найбільшу зацікавленість або потребують додаткової уваги. Інформація про частоту взаємодій також допомагає ідентифікувати об'єкти чи процеси, які викликають труднощі у студентів, оскільки часте звертання до одного й того ж елемента може свідчити про складність його засвоєння або розуміння. Такий аналіз дозволяє викладачам коригувати навчальні матеріали, забезпечуючи більш глибоке розуміння або додаткові пояснення для складних тем.

Важливим аспектом є вимірювання часу, проведеного студентом у різних зонах віртуальної лабораторії. База даних може фіксувати, скільки часу студент проводить, працюючи з окремими інструментами або виконуючи певні експерименти. Це дозволяє виявити, на які аспекти завдань студент витрачає найбільше часу, що може свідчити про складність певних дій або потребу у додатковому практичному занятті. За допомогою цієї інформації викладачі можуть приймати обґрунтовані рішення, наприклад, надавати студентам індивідуальну підтримку або виділяти додатковий час на певні етапи лабораторних робіт.

База даних також фіксує, де саме студент допускає помилки, які дії призводять до помилкових результатів, і як часто ці помилки повторюються. Завдяки такому аналізу система може не тільки виявляти загальні проблеми в розумінні матеріалу, але й надавати індивідуальні рекомендації для кожного студента.

Наприклад, якщо студент повторно робить ту саму помилку під час виконання певної задачі, це може свідчити про необхідність глибшого вивчення даної теми або повторного пояснення. Викладачі можуть переглядати історію помилок студента, щоб краще розуміти його прогрес і забезпечити більш цілеспрямовану допомогу.

Для ефективного збору інформації у віртуальному середовищі використовуються камери та датчики, які автоматично фіксують дії та позиції студентів під час роботи. Ці технології дозволяють збирати великий обсяг даних про поведінку студента у віртуальному середовищі: наприклад, як він взаємодіє з інструментами, які об'єкти обирає для дослідження та які рухи виконує. Така інформація є цінною для подальшої обробки та аналізу, оскільки дозволяє отримати точну картину поведінки та взаємодії студентів з приладами і виявляти закономірності у їхніх діях.

Зібрана база даних відкриває широкі можливості для покращення навчального процесу. На основі аналізу даних можна адаптувати навчальні

матеріали, додаючи нові завдання або модифікуючи існуючі, щоб забезпечити глибше розуміння матеріалу.

Також база даних дозволяє виявляти студентів, яким необхідна додаткова допомога, і забезпечувати їм підтримку у вигляді додаткових пояснень або індивідуальних занять. Це дозволяє прийняти рішення щодо змісту та структури навчання, адаптуючи курс під потреби кожного студента.

### 1.2.2 Штучний інтелект

Застосування штучного інтелекту (ШІ) у процесі навчання є одним з найбільш перспективних напрямків у розвитку освітніх технологій [11]. Віртуальні лабораторії, створені на основі Unreal Engine 5, можуть збирати велику кількість даних про студентів під час навчання: як вони взаємодіють із середовищем, де найчастіше допускають помилки, які завдання виконують швидше, а з якими виникають труднощі. Такий підхід дає можливість не лише фіксувати діяльність кожного користувача, а й проводити аналіз цих даних для оптимізації навчального процесу.

Однією з ключових цілей аналізу даних є можливість адаптувати процес навчання під індивідуальні потреби кожного студента. Завдяки аналізу результатів ШІ може автоматично визначати, на яких етапах студент стикається з труднощами, та пропонувати відповідні підказки чи додаткові навчальні матеріали. Наприклад, якщо система бачить, що студент часто помиляється під час взаємодії з певним обладнанням, вона може надавати інструкції або відеопояснення, спрямовані на пояснення роботи саме з цим обладнанням. Це дозволяє кожному студенту отримувати підтримку в залежності від його потреб та прогресу, що підвищує ефективність навчання.

Для реалізації розглянутих технологій використовується Unreal Engine 5, а одним із інструментів для інтеграції штучного інтелекту у даному додатку є Behavior Trees (дерева поведінки) (рис. 1.3).

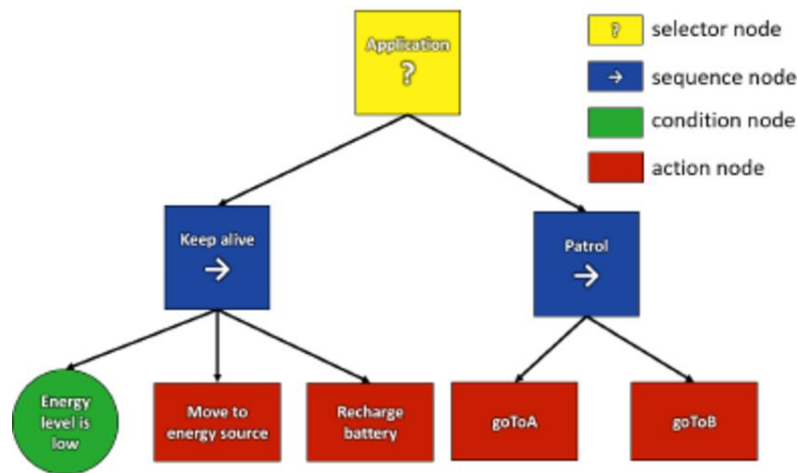


Рисунок 1.3 – Приклад дерева поведінки [12]

Behavior Trees дозволяють створювати складні сценарії поведінки для віртуальних персонажів, приладів і симуляцій. У контексті навчання та лабораторних занять Behavior Trees можуть використовуватися для моделювання інтелектуальної взаємодії між студентом і віртуальним середовищем. Наприклад, поведінкові дерева можуть контролювати віртуального асистента, який надає поради студенту, вказує на помилки та навіть ставить уточнюючі запитання, щоб стимулювати глибше розуміння матеріалу або для відтворення логіки обладнання.

Створення Behavior Trees для освітніх симуляцій включає багатоетапну розробку. Спершу визначаються основні дії, які може виконувати віртуальний об'єкт. Далі ці дії структуруються у дерево поведінки, де враховуються різні сценарії та відповіді на дії студента. Кожна гілка Behavior Tree відповідає за конкретний сценарій, наприклад, коли студент робить правильний або помилковий вибір, віртуальний об'єкт реагує відповідно, надаючи зворотний зв'язок.

Також треба виділити, що Unreal Engine має систему Unreal Engine Machine Learnig – це вбудована система Unreal для розробки та інтеграції моделей машинного навчання [13]. Ви можете створювати та навчати моделі ШІ за допомогою зовнішніх фреймворків (наприклад, у Python), а потім імпортувати

їх в Unreal для виконання на стороні сервера або клієнта. Це дозволяє використовувати ШІ для аналізу даних у реальному часі, що може бути корисним для персоналізованого контенту.

Для більш детального аналізу результатів також використовуються алгоритми машинного навчання, що дозволяють системі не лише відстежувати поведінку студентів, але й прогнозувати можливі труднощі на основі вже зібраних даних. Якщо студенту знадобилося більше часу для виконання певної справи, система може оцінити цей результат і, можливо, запропонувати додаткові ресурси для кращого розуміння матеріалу. Такі моделі можуть не лише адаптувати процес навчання, але й допомагати викладачам у формуванні програми навчання.

Завдяки аналізу даних ШІ може забезпечити своєрідний інтерактивний зворотний зв'язок для студентів. Кожен студент може отримувати індивідуальні поради та рекомендації. Система може генерувати звіти про результати виконаних лабораторних робіт, показувати статистику прогресу студента, а також давати поради щодо поліпшення навичок. Крім того, така система може автоматично виділяти зони, де студент досяг успіху, і зони, де є потреба у вдосконаленні.

Ще одна важлива перевага ШІ – це можливість прогнозування результатів навчання. За допомогою аналізу даних система може не лише відслідковувати поточний рівень підготовки студента, але й робити прогнози на майбутнє. Наприклад, ШІ може оцінити ймовірність успішного складання певного модуля або лабораторної роботи. Відповідно до таких прогнозів система може рекомендувати індивідуальну траєкторію навчання, пропонуючи студенту додаткові завдання або тематичні тести для закріплення знань. Це дозволяє мінімізувати ризик невдачі в процесі навчання та значно підвищити ймовірність успішного виконання всіх завдань курсу.

ШІ також спрощує процес оцінювання студентів. Наприклад, алгоритми можуть автоматично відстежувати всі помилки, які допускає студент, і надавати йому можливість їх виправити. Це дозволяє системі проводити більш точне

оцінювання, враховуючи не лише кількість правильно виконаних завдань, але й прогрес, який студент зробив у процесі навчання. Таким чином, автоматизовані оцінки є об'єктивними й детальними, дозволяючи викладачам зосередитися на індивідуальній підтримці студентів, а не на технічному процесі перевірки.

Штучний інтелект також здатний виявляти загальні тенденції в навчанні всієї групи студентів. Якщо система бачить, що більшість студентів допускає схожі помилки або витрачає занадто багато часу на певний модуль, це може свідчити про складність конкретної теми чи завдання. В такому випадку ШІ може автоматично попередити викладача, що певна частина матеріалу потребує додаткових пояснень або модифікації. Такий підхід дозволяє покращити загальну якість навчання та забезпечити ефективне засвоєння знань не лише окремими студентами, а й цілими групами.

Behavior Trees можуть використовуватися не лише для управління поведінкою віртуальних об'єктів, а й для інтеграції з іншими компонентами ШІ, такими як обробка природної мови або розпізнавання об'єктів у просторі. Це дозволяє створювати більш комплексні сценарії, де віртуальний асистент чи прилад може взаємодіяти з іншими різноманітними об'єктами, реагуючи на сигнали чи голосові команди студента або аналізуючи зміни у віртуальному середовищі. Розширені можливості Behavior Trees дозволяють автоматично оновлювати сценарії поведінки ШІ, підлаштовуючи їх під індивідуальний стиль навчання кожного студента.

### 1.3 Висновки до першого розділу

У першому розділі було проведено всебічний аналіз технологій віртуальної реальності (VR), доповненої реальності (AR) та змішаної реальності (MR). У ході дослідження встановлено, що ці технології створюють нові можливості для побудови інтерактивних, адаптивних і безпечних навчальних середовищ. Особливу увагу приділено аналізу основних переваг використання VR у навчанні, таких як гнучкість проведення лабораторних занять, зниження

витрат на матеріали та обладнання, а також можливість проведення експериментів у віртуальному просторі, який максимально наближений до реальних умов.

Досліджено також синергію VR із такими технологіями, як штучний інтелект (ШІ) та бази даних (БД). Встановлено, що використання ШІ в поєднанні з VR забезпечує персоналізацію навчального процесу, автоматизацію аналізу результатів і адаптацію освітнього контенту під індивідуальні потреби студентів. Базы даних, у свою чергу, виконують ключову роль у зборі та організації великих обсягів інформації, пов'язаної з прогресом студентів, їхньою взаємодією з системою та ефективністю виконання завдань.

Особливу увагу приділено важливості створення віртуальних лабораторій. Такі середовища дозволяють студентам проводити експерименти, симулювати складні процеси та відпрацьовувати практичні навички без необхідності фізичної присутності в лабораторії. Завдяки використанню VR і IoT технологій можливо інтегрувати реальні дані з сенсорів у віртуальне середовище, що дозволяє відстежувати параметри обладнання в режимі реального часу та проводити симуляції, що максимально відповідають реальним умовам.

Значну увагу було приділено дослідженню ролі штучного інтелекту в процесі навчання. Виявлено, що ШІ здатен аналізувати помилки студентів, генерувати персоналізовані рекомендації, прогнозувати можливі труднощі в навчанні та адаптувати навчальні завдання залежно від рівня підготовки кожного студента. Крім того, інтеграція ШІ у віртуальні лабораторії дозволяє створювати сценарії взаємодії між студентами та віртуальними об'єктами, забезпечуючи додатковий рівень інтерактивності та ефективності навчання.

Таким чином, результати проведеного аналізу підтверджують доцільність використання віртуальної реальності, штучного інтелекту та баз даних у навчальному процесі.

## 2 МОДЕЛЮВАННЯ СТРУКТУРИ ТА ПЛАНУ РЕАЛІЗАЦІЇ ПРОЄКТУ

### 2.1 Аналіз літератури та досліджень

Перед тим, як перейти до моделювання структури та плану реалізації, було проведено аналіз можливостей впровадження AI та 3D-технологій у різні галузі навчання, що показав, що ці технології однозначно можуть підвищити ефективність навчального процесу, зробивши його більш інтерактивним і персоналізованим. Зокрема, використання штучного інтелекту дозволяє адаптувати навчальний контент до індивідуальних потреб студентів, що сприяє більш глибокому засвоєнню матеріалу.

Крім того, можна оцінити вплив розглянутих технологій на освіту та інші галузі, оскільки закордоном активно впроваджуються інноваційні рішення, які значно змінюють традиційні підходи до навчання. У країнах Європи та Північної Америки спостерігається зростаючий інтерес до використання штучного інтелекту для автоматизації навчальних процесів, що дозволяє створювати персоналізовані навчальні траєкторії, адаптуючи матеріал під потреби кожного студента.

Також треба відзначити, що і 3D-технології використовуються в освіті для створення віртуальних лабораторій та симуляцій, які дозволяють студентам проводити експерименти у безпечному середовищі, що значно підвищує їхнє розуміння теорії та практики. Інтеграція цих технологій сприяє розвитку критичного мислення та креативності учнів, оскільки вони навчаються вирішувати реальні проблеми за допомогою інноваційних підходів.

В українському просторі наявні лише примітивні моделі 2D віртуальних лабораторій чи спрощені установки, але у ході аналізу можна помітити тенденцію до зростання зацікавленості та попиту на дані технології.

Наприклад, вже було реалізовано такі розробки як: хімічна лабораторія Virtual Lab [14] у Криворізькому державному педагогічному університеті, наявно багато просторів для ознайомлення з фізичними явищами у школах та вищих навчальних закладів. За партнерством з університетом штату Колорадо у Боулдері ми маємо доступ до PhET (рис. 2.1), який надає безкоштовні, інтерактивні науково-математичні симуляції, засновані на наукових дослідженнях.

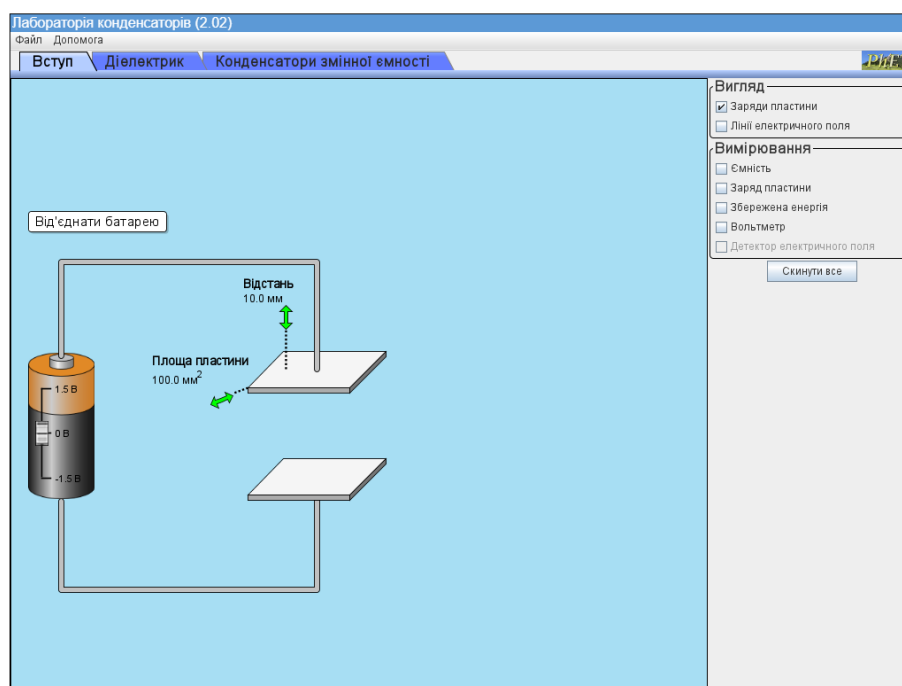


Рисунок 2.1 – Приклад спрощеної двовимірної лабораторної установки Phet [15]

Розробники ретельно перевіряють та оцінюють кожне моделювання, щоб забезпечити ефективність навчання. Ці тести включають опитування студентів і спостереження за використанням моделювання в класах. Симуляції написані на HTML5 (з деякими застарілими симуляціями на Java або Flash), і їх можна запускати онлайн або завантажити на свій комп'ютер. Усі симуляції мають відкритий код (дивіться наш вихідний код). Численні спонсори підтримують

проект PhET, завдяки чому ці ресурси є безкоштовними для всіх студентів і викладачів.

Також МОН України повідомляло про відкритий доступ до віртуальної наукової лабораторія Labster для українських закладів освіти [16]. Треба зауважити, що усі матеріали платформи створюють та перевіряють науковці з відповідних галузей.

Як стверджують автори проекту, більшість користувачів лабораторії – заклади вищої освіти, однак матеріали Labster будуть цікавими, зокрема, і для старшокласників під час дистанційного чи змішаного навчання.

Крім того, педагоги можуть додавати до курсів на платформі власні тестові завдання та переглядати аналітику. У МОН зазначили, що школи, заклади професійної (професійно-технічної) освіти, фахової передвищої та вищої освіти вже отримали інструкції з реєстрації.

Отже, результати показують, що освітні програми потрібно оновлювати, щоб відповідати сучасним вимогам цифрового світу. Це дає можливість покращити навчальні методи і підвищити якість освіти. На даний момент наявні відносно прості віртуальні простори, проте вже є можливість їх вдосконалити та перевести на більш високий рівень з різноманітними можливостями за допомогою нових технологій.

Також варто проаналізувати приклади впровадження додаткових технологій. У результаті було виявлено, що у проектах на основі електронних медичних записів, таких як IBM Watson Health, AI аналізує дані пацієнтів для підтримки прийняття рішень лікарями, що забезпечує персоналізоване лікування. Також яскравим прикладом є використання поєднання баз даних і AI в e-commerce. Великі онлайн-платформи, як Amazon, використовують потужні бази даних для зберігання інформації про покупців, їхні вподобання та історію покупок. На основі цих даних штучний інтелект застосовує алгоритми машинного навчання для прогнозування потреб користувачів, рекомендуючи їм товари, що можуть їх зацікавити. Це дозволяє значно покращити персоналізацію послуг і збільшити прибуток завдяки цільовій рекламі та рекомендаціям.

У сфері інтернету речей (IoT) поєднання баз даних та AI дозволяє ефективно керувати та аналізувати дані, що надходять від численних сенсорів. У таких проєктах, як Smart Cities, штучний інтелект використовує дані з баз для моніторингу та оптимізації міської інфраструктури. AI аналізує інформацію з різних джерел – від систем управління транспортом до моніторингу рівня забруднення повітря, і на основі отриманих даних може прогнозувати зміни або автоматично коригувати систему управління містом для покращення якості життя.

Проєкт «Osso VR» є прикладом поєднання VR, AI та DB для тренування хірургів (рис. 2.2). Ця платформа використовує віртуальну реальність для симуляції медичних процедур, що дозволяє студентам та лікарям практикувати операції в безпечному середовищі без реальних пацієнтів. Платформа дозволяє створювати детальні 3D-моделі органів та тканин, які віртуально реконструюються для кожного користувача. Використання бази даних є ключовим аспектом цього проєкту, оскільки вона зберігає дані про історію тренувань кожного користувача, такі як точність виконання операцій, час, витрачений на кожен крок, і складність процедур.



Рисунок 2.2 – Процедурний тренінг із зануренням Osso VR [17]

Штучний інтелект у цій системі використовується для аналізу ефективності виконаних процедур. Наприклад, AI оцінює техніку хірурга, виявляє помилки або відхилення від нормальних стандартів та надає рекомендації для покращення результатів. Віртуальні тренування за допомогою AI можуть адаптуватися до рівня кожного студента, поступово ускладнюючи завдання в залежності від його прогресу. Таким чином, система не тільки дає студентам можливість тренуватися в реалістичних умовах, але й використовує бази даних для зберігання та аналізу їхнього навчального процесу, що дозволяє точно оцінювати прогрес та надавати персоналізовані рекомендації.

## 2.2 Постановка задач

На основі проведеного аналізу доречно створити план реалізації майбутнього проєкту за поставленим технічним завданням (ТЗ).

Постановка задач проєкту створення віртуальної лабораторії зі застосуванням штучного інтелекту та баз даних передбачає розробку концепції віртуальної лабораторії, визначення основних функцій та можливостей, які вона повинна мати. Необхідно інтегрувати штучний інтелект для моделювання експериментів, аналізу даних та надання рекомендацій користувачам, а також розробити структуру бази даних, яка буде зберігати інформацію про експерименти, результати та користувачів. Забезпечення користувацького інтерфейсу є важливим етапом, оскільки потрібно проєктувати інтуїтивно зрозумілий інтерфейс для користувачів лабораторії, а також визначити методи тестування системи на ефективність і точність.

## 2.3 Актуальність

Актуальність проєкту ґрунтується на тенденції до зростання ролі онлайн-освіти в умовах пандемії та глобалізації, потребою в доступності лабораторних експериментів для студентів та дослідників незалежно від їхнього географічного

розташування, інноваційністю використання штучного інтелекту для автоматизації і вдосконалення навчальних процесів, а також економією ресурсів завдяки зменшенню витрат на матеріали та обладнання, оскільки експерименти проводитимуться віртуально.

#### 2.4 Завдання дослідження

Завдання проєкту полягає у створенні інтерактивної віртуальної лабораторії, яка дозволить користувачам виконувати наукові експерименти за допомогою віртуального обладнання, проводити аналіз та отримувати допомогу зі застосуванням штучного інтелекту, а призначенням є надання студентам та дослідникам зручного та доступного інструменту для навчання та проведення досліджень у різних наукових дисциплінах.

#### 2.5 Вимоги та вхідні дані

Вимоги та вхідні дані для реалізації даного проєкту включають технічні вимоги, такі як платформа (у даному випадку десктопний додаток, проте можна використовувати у подальшому веб-додаток або мобільний додаток), технології (мови програмування, бази даних, фреймворки, у даному випадку це C++ та DataAsset на Unreal Engine 5) та серверні та клієнтські вимоги, а також функціональні вимоги, зокрема можливість виконання різних типів експериментів, інтерактивний інтерфейс для введення даних та отримання результатів, систему моніторингу та аналізу даних.

Вхідні дані, які будуть використані в проєкті, включають типи експериментів, що будуть включені, формати даних та параметри для моделей штучного інтелекту (для даного проєкту використовується Behaviour Tree in Unreal Engine 5).

Для створення та вдосконалення даного проєкту використовується власна попередня розробка віртуального простору, візуально аналогічна інтер'єру кафедри КІТАР.

## 2.6 Аналіз отриманих результатів та розробка алгоритму дій

Даний пункт передбачає використання статистичних методів для оцінки ефективності експериментів та зворотній зв'язок від користувачів для покращення системи, а план дій включає створення прототипу віртуальної лабораторії, тестування функціональності та юзабіліті, впровадження змін на основі тестування та аналізу, а також запуск фінальної версії проєкту та його популяризацію серед цільової аудиторії.

Загалом алгоритм можна поділити на наступні ключові етапи (рис. 2.3).

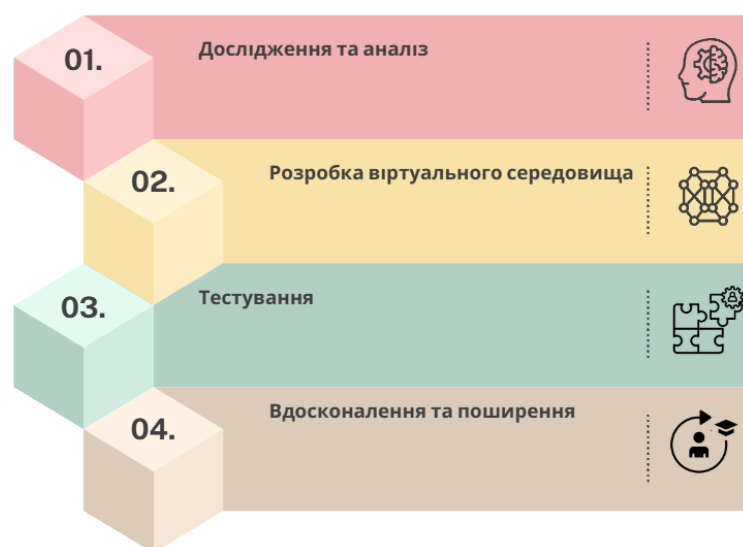


Рисунок 2.3 – Ключові етапи реалізації проєкту

Перший етап – дослідження та аналіз потреб для визначення необхідних елементів віртуального середовища. Для цього можна провести опитування серед викладачів, студентів і дослідників, щоб зрозуміти їхні потреби та вимоги до віртуальних лабораторій. Визначити, які типи експериментів найбільше

цікавлять користувачів, а також їхні очікування щодо функціоналу та інтерфейсу.

На основі зібраних даних створити концепцію віртуальної лабораторії, що включає її основні функції, типи експериментів, що можуть бути виконані, і методи взаємодії з користувачем. Визначити роль штучного інтелекту в автоматизації процесів, таких як аналіз даних, рекомендації та адаптація навчальних планів.

Другий етап – розробити середовище віртуальної лабораторії в Unreal Engine 5, яке включатиме фронтальну частину (3D-моделі), бекенд (логіка на Blueprint чи C++) та базу даних.

Створити структуру бази даних для зберігання інформації про експерименти, результати, користувачів і взаємодії з системою. Розпочати програмування віртуальної лабораторії. Впровадити модулі штучного інтелекту, які зможуть аналізувати дані експериментів, надавати рекомендації користувачам на основі отриманих результатів та автоматизувати рутинні процеси, такі як оцінка успішності експериментів.

Третій етап – провести всебічне тестування віртуальної лабораторії, включаючи функціональне тестування та навантажувальне тестування. Важливо отримати зворотний зв'язок від користувачів, щоб виявити можливі проблеми та недоліки. На основі результатів тестування внести зміни та вдосконалення в систему.

Четвертий етап – після завершення тестування та вдосконалення віртуальної лабораторії провести її поширення та запуск у навчальних закладах. Підготувати навчальні матеріали для викладачів та студентів, а також створити інструкцію з користування системою.

Після запуску забезпечити моніторинг роботи системи, щоб виявляти та вирішувати можливі проблеми, а також збирати дані про використання віртуальної лабораторії. Регулярно оновлювати систему на основі відгуків користувачів і нових технологій. Проводити регулярну оцінку ефективності віртуальної лабораторії в контексті навчального процесу. Вимірювати

успішність студентів, задоволеність користувачів і досягнення навчальних цілей. На основі отриманих даних розробляти плани для подальшого вдосконалення та розширення функціоналу лабораторії.

Отже, можна побачити, що процес створення віртуальної лабораторії складається з кількох етапів, кожен з яких має свою мету та важливі функції для успішної реалізації проєкту. Переваги цих етапів включають систематичний підхід до розробки, що дозволяє врахувати всі потреби користувачів та забезпечити високу якість кінцевого продукту.

Однак є й деякі недоліки, проте вони можуть бути мінімізовані за допомогою ретельного планування та адаптації процесу. Наприклад, дослідження потреб користувачів може зайняти багато часу, оскільки не всі учасники здатні чітко сформулювати свої вимоги або мають різні уявлення про те, як має функціонувати віртуальна лабораторія. Це може створити труднощі на ранніх етапах розробки. Проте, проведення глибоких інтерв'ю, опитувань та обговорень із потенційними користувачами допоможе зібрати достатньо інформації для створення максимально зручної і функціональної системи.

Щодо технічних складнощів, пов'язаних із розробкою віртуального середовища, то вони можуть вимагати значних ресурсів та часу. Використання таких платформ, як Unreal Engine 5, потребує кваліфікованих фахівців, що може бути обтяжливим для деяких команд. Проте, за умови залучення досвідчених розробників і розподілу завдань, цей процес можна здійснити ефективно.

Навантажувальне тестування може виявити непередбачувані проблеми, які потребують додаткових зусиль для їх усунення, особливо в умовах великих обсягів даних або складних експериментів. Але регулярне оновлення та покращення системи з урахуванням зворотного зв'язку від користувачів дозволить вчасно реагувати на ці проблеми та забезпечити безперебійну роботу лабораторії.

Таким чином, хоча є певні недоліки, їх можна успішно подолати за допомогою комплексного підходу, ефективного управління процесом розробки та постійної оптимізації системи.

## 2.7 Висновки до другого розділу

У другому розділі було здійснено моделювання структури та плану реалізації проєкту, спрямованого на створення інтерактивної віртуальної лабораторії з використанням технологій віртуальної реальності, штучного інтелекту та баз даних.

Аналіз літератури та досліджень дозволив встановити, що інтеграція сучасних технологій у навчальний процес значно підвищує ефективність освіти, створюючи умови для персоналізованого навчання. Було вивчено приклади успішного впровадження віртуальних лабораторій у різних освітніх і наукових сферах, що підтвердило їхню актуальність і затребуваність.

Під час постановки задач було визначено ключові технічні та функціональні вимоги до віртуальної лабораторії. Зокрема, проєкт передбачає створення віртуального середовища, що імітує реальні лабораторні умови, інтеграцію штучного інтелекту для аналізу результатів і надання рекомендацій, а також бази даних для зберігання інформації про результати експериментів і діяльність користувачів.

Розроблено детальний план реалізації проєкту, який складається з чотирьох основних етапів: аналіз потреб і вимог користувачів, створення віртуального середовища з інтеграцією ШІ та баз даних, тестування та вдосконалення системи на основі зворотного зв'язку, а також впровадження та популяризація проєкту серед цільової аудиторії.

Отже, результати другого розділу закладають основу для практичної реалізації проєкту.

## 3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ ВІРТУАЛЬНОГО СЕРЕДОВИЩА

### 3.1 Аналіз засобів розробки та вибір технологій

У проєкті основою для розробки віртуальної лабораторії є рушій Unreal Engine 5 (UE5), який є надійною та сучасною платформою для створення інтерактивних 3D-середовищ, симуляцій та ігор.

Unreal Engine 5 став значним кроком вперед у порівнянні з попередніми версіями, забезпечуючи надзвичайно високий рівень графічної деталізації, реалістичні світлові ефекти та ефективне використання ресурсів. Однією з ключових переваг є підтримка сучасних технологій, таких як віртуальна реальність (VR), штучний інтелект (AI) та власні бази даних.

Unreal Engine 5 [18] надає розробникам безліч можливостей для інтеграції віртуальної реальності. Зокрема, підтримка VR у UE5 дозволяє створювати високоякісні віртуальні середовища, що забезпечують максимальну присутність і реалістичність для користувачів. Завдяки вбудованим інструментам і оптимізованим алгоритмам, Unreal Engine 5 дозволяє знижувати затримки візуалізації та забезпечувати плавність роботи навіть при великих обсягах даних, що критично важливо для VR-додатків. Це дає змогу розробляти віртуальні лабораторії та тренажери, які надають користувачам можливість проводити експерименти, взаємодіяти з 3D-об'єктами та відчувати себе частиною реалістичного середовища.

Щодо інтеграції з штучним інтелектом (AI), Unreal Engine 5 має вбудовані можливості для застосування алгоритмів машинного навчання [19], що дозволяє створювати адаптивні системи для аналізу даних та оптимізації взаємодії з користувачем. AI може бути використаний для автоматизації процесів у віртуальній лабораторії, наприклад, для аналізу результатів експериментів, надання рекомендацій користувачам або адаптації навчальних планів залежно

від успіхів учасників. Unreal Engine 5 також підтримує інтеграцію з популярними бібліотеками машинного навчання, що дає змогу розширити функціонал системи і покращити досвід користувачів. Крім того на платформі наявні додаткові плагіни для роботи з AI та розумними об'єктами у середовищі.

Бази даних є невід'ємною частиною будь-якої великої системи, і Unreal Engine 5 пропонує безліч варіантів для їх інтеграції. Платформа дозволяє зберігати та управляти великими обсягами даних, зокрема інформацією про експерименти, результати та поведінку користувачів. В Unreal Engine 5 можна налаштувати підключення до зовнішніх баз даних [20], використовувати внутрішні системи для збереження інформації або навіть реалізувати механізми зберігання даних безпосередньо у віртуальному середовищі. Це дозволяє зберігати та обробляти інформацію про виконані експерименти, взаємодії з об'єктами в лабораторії та інші параметри, які можуть бути потім використані для аналітики та адаптації системи.

Потенціал Unreal Engine 5 величезний, і його можливості інтеграції VR, AI та баз даних відкривають нові горизонти для створення високотехнологічних, адаптивних і ефективних систем. Власні функції UE5, такі як інтеграція з зовнішніми бібліотеками і підтримка широкого спектра платформ. Unreal Engine 5 надає потужний інструментарій для створення віртуальних лабораторій, навчальних симуляторів та систем, що комбінують віртуальну реальність, штучний інтелект і великі бази даних для наукових і навчальних цілей.

Варто зазначити, що UE5 є безкоштовним для розробників, що робить його дуже привабливим у використанні для даного проєкту.

Також Unreal Engine 5 має чудову інтеграцію з Visual Studio [21]. Visual Studio служить основним середовищем для написання коду на мові C++, яке є основною мовою програмування для UE5. Ця інтеграція дає змогу розробникам ефективно працювати з кодом, тестувати, дебажити та здійснювати інші важливі операції в процесі розробки.

Користувачі UE5 мають можливість безпосередньо підключати проєкт до Visual Studio, що дає доступ до широких можливостей, таких як автодоповнення

коду, рефакторинг, покрокове дебагування і інші функції, що значно підвищують продуктивність розробки. При цьому C++ є однією з основних мов, що підтримується Unreal Engine, дозволяючи розробникам створювати високоефективні та оптимізовані системи для різних аспектів гри або додатка.

У UE5, використання C++ дозволяє мати повний контроль над механізмами та логікою. Наприклад, можна створювати власні класи для управління ігровою логікою, фізикою, обробкою введення та іншими критичними аспектами роботи застосунку. Розробники можуть створювати власні класи, успадковуючи стандартні класи UE5, такі як AActor або UObject, що дозволяє значно розширювати функціональність та можливості ігрових механік. Завдяки цьому, UE5 дозволяє реалізовувати складні і унікальні ігрові концепції, такі як штучний інтелект, складні фізичні симуляції або взаємодія з великими даними.

C++ дає змогу програмістам максимально оптимізувати виконання програми, знижуючи затримки та покращуючи ефективність обробки великих обсягів даних, що особливо важливо для VR-додатків та великих відкритих світів, що є характерними для багатьох проектів на UE5.

В UE5, Visual Studio забезпечує автоматичне підключення до середовища розробки, де програмісти можуть додавати власні нові класи, компоненти і функції до проекту без необхідності постійно перемикатися між середовищами.

Важливою частиною цієї інтеграції є також можливість використання Blueprints [22] – візуального інтерфейсу для програмування, який дозволяє швидко створювати ігрові механіки без написання коду. Однак, для більш складних і кастомізованих рішень, C++ дає більшу гнучкість і можливість повністю контролювати всі аспекти розробки. UE5 дозволяє комбінувати обидва підходи, використовуючи C++ для створення базової логіки, а Blueprints – для більш простих або швидких розробок.

Visual Studio [23] є стандартним інструментом для розробки з UE5 і, на відміну від інших редакторів, має повну інтеграцію з екосистемою Unreal Engine. Оскільки Unreal Engine 5 активно підтримує C++ як основну мову

програмування, Visual Studio є найкращим вибором для великих, складних проєктів, де важлива максимальна продуктивність і контроль над кодом.

VS Code [24], з іншого боку, є більш легким і швидким текстовим редактором, який надає розробникам базові можливості для програмування на C++ в UE5. Однак, на відміну від Visual Studio, VS Code потребує додаткових плагінів та налаштувань для повноцінної роботи з UE5.

Rider від JetBrains [25] – це ще один редактор, який підтримує роботу з UE5, і він набирає популярності серед розробників, особливо серед тих, хто вже звик до середовищ JetBrains, таких як IntelliJ IDEA або PyCharm. Проте, на відміну від Visual Studio, Rider має деякі обмеження в контексті підтримки специфічних можливостей UE5, таких як специфічні плагіни або деякі інструменти для роботи з графікою.

Таким чином, взаємодія Unreal Engine 5 з Visual Studio та використання C++ класів дозволяє розробникам мати великий контроль над проєктом та є найкращим вибором.

Тепер розглянемо етап моделювання в Blender [26], який є одним із ключових етапів у створенні 3D-моделей для Unreal Engine 5. Blender – це безкоштовне і відкрито доступне програмне забезпечення, яке надає інструменти для моделювання, текстурування, рендерингу та анімації. На етапі моделювання розробники створюють 3D-об'єкти та сцени, які пізніше будуть інтегровані в UE5 для створення віртуальних світів або інших об'єктів для віртуальних лабораторій, ігор чи інших додатків. В Blender можна працювати як з полігональним моделюванням, так і з процедурними методами, що дозволяє отримувати точні та високоякісні моделі з мінімумом часу.

Однією з основних переваг Blender є його доступність і багатофункціональність. Крім того, Blender підтримує інтеграцію з UE5, що дозволяє зручно експортувати моделі та текстури без необхідності в додаткових налаштуваннях. У Blender є величезна спільнота користувачів, яка активно ділиться знаннями та ресурсами, що значно полегшує навчання та підвищує ефективність роботи. З точки зору інтерфейсу, Blender має дуже гнучкі

інструменти для швидкої обробки моделей і зручні методи для організації проектів, що дає йому велику перевагу перед іншими програмами, які можуть мати більш обмежені можливості чи вимагати платної ліцензії.

Крім того, етап рігу та анімації в Autodesk Maya [27] є не менш важливим для даного проекту для створення складних рухів та деформацій об'єктів у 3D-просторі. Maya – це професійне програмне забезпечення, яке використовується для моделювання, рігу та анімації, і широко застосовується в індустрії кіно, анімації та ігор. На етапі рігу (створення скелетної структури для персонажів або об'єктів) в Maya створюються кістки, зв'язки і контролери для анімації, що дозволяє анімувати персонажів і об'єктів з максимальною реалістичністю. Maya підтримує інструменти для автоматизації рігу, що значно полегшує процес створення складних анімацій, таких як симуляція руху, а також складні взаємодії між об'єктами.

Переваги Maya перед іншими програмами полягають в її надійності і високому рівні підтримки анімації. Вона пропонує розширені функції для роботи з анімацією, включаючи Time Slider для точного налаштування анімацій, можливість роботи з Motion Capture, а також інструменти для ретаргетування анімацій і роботи з інверсною кінематики. Крім того, Maya має розвинуту систему плагінів і скриптів, що дозволяє налаштовувати програму під конкретні потреби проекту. Вона активно підтримується у великих студіях і є стандартом у багатьох індустріях, що робить її оптимальним вибором для професійних аніматорів та риггерів.

Треба зауважити, що Maya має пробну версію на 30 днів та безкоштовну версію для студентів, а Blender є повністю безкоштовним і відкритим програмним забезпеченням без обмежень на використання.

Було обрано це програмне забезпечення, оскільки воно повністю відповідає всім необхідним вимогам, а також даний вибір обумовлений досвідом успішного використання в попередніх проектах.

Наступник кроком є вибір бази даних, SQLite [28] – це легка, вбудована реляційна база даних, яка не потребує окремого сервера для роботи. Вона

зберігає всі дані в одному файлі, що робить її ідеальним рішенням для застосунків, де важлива простота в налаштуваннях і мінімальне споживання ресурсів. В контексті Unreal Engine 5 (UE5) SQLite може бути використана для зберігання локальних даних, таких як налаштування користувачів, результати експериментів, дані про прогрес чи інші типи інформації, які потрібно швидко зберігати і обробляти.

UE5 підтримує інтеграцію з SQLite через різні плагіни або пряме підключення через C++. У даному проєкті буде проведено саме пряме підключення.

Це дозволяє розробникам використовувати SQLite для зберігання даних у внутрішніх проєктах, де потрібно ефективно зберігати та запитувати інформацію, не використовуючи зовнішні сервери або більш складні системи управління базами даних. SQLite може бути корисною для малих і середніх проєктів, які не потребують масштабованих рішень, таких як сервери SQL.

Переваги SQLite включають її простоту, малий розмір та ефективність. Вона не потребує налаштування серверної інфраструктури, що робить її ідеальним вибором для мобільних додатків, невеликих ігор або віртуальних лабораторій, де база даних повинна бути вбудована без складних зовнішніх залежностей. Вона також швидко працює з невеликими обсягами даних, що є великим плюсом для продуктивності при роботі з локальними даними.

Аналоги SQLite [29] включають MySQL, PostgreSQL та MongoDB, які можуть бути більш підходящими для великих проєктів, що потребують високої масштабованості, обробки великих обсягів даних або комплексної роботи з віддаленими базами даних. Однак ці системи зазвичай потребують налаштування сервера і більше ресурсів для обробки запитів, що робить їх менш оптимальними для невеликих локальних застосунків.

Доречим буде використання DB Browser [30] як доповнення до SQLite. DB Browser – це безкоштовна, з відкритим вихідним кодом програма, яка надає зручний графічний інтерфейс для роботи з базами даних SQLite. Вона дозволяє

користувачам створювати, переглядати та редагувати SQLite-бази даних, а також виконувати SQL-запити без необхідності використовувати командний рядок.

Програма підтримує всі основні функції роботи з базами даних SQLite, включаючи імпорт і експорт даних у різні формати, такі як CSV або SQL, що робить її зручною для роботи з даними з інших джерел.

Крім того, DB Browser for SQLite дозволяє користувачам зберігати бази даних у вигляді файлів, що забезпечує зручність у використанні та зберіганні даних на локальному рівні.

Однією з основних переваг DB Browser for SQLite є його простота та легкість у використанні. Інтерфейс програми дає змогу швидко виконувати стандартні операції, не вимагаючи від користувачів великих технічних знань або досвіду. Вона також дозволяє створювати і виконувати SQL-запити в інтерактивному режимі, що значно спрощує процес розробки та тестування.

Оскільки дана робота є комплексною, то доречно використовувати GitHub [31] для хостингу та спільної розробки програмного забезпечення, що використовує систему контролю версій Git. Вона дозволяє кільком розробникам працювати над одним проектом, забезпечуючи зручний інтерфейс для керування кодом, відстеження змін і спільного внесення змін. На GitHub можна створювати репозиторії для зберігання коду, створювати гілки для окремих функцій, здійснювати злиття та переглядати історію змін.

GitHub підтримує pull-запити, які дозволяють розробникам пропонувати зміни до основної гілки коду, а також надає інструменти для рев'ю коду, обговорення та виправлення помилок. Це робить платформу ідеальною для колективної роботи, управління проектами та збереження високої якості коду в команді.

### 3.2 Налаштування програмного середовища

Оскільки ця робота є комплексною і передбачає спільну участь кількох розробників, доцільно інтегрувати процес із платформою GitHub (рис. 3.1) для

ефективної командної розробки та управління проектом. GitHub забезпечує централізоване сховище для коду, зручні інструменти для відстеження змін та координації між членами команди. Хоча основні аспекти процесу описані у другій частині, варто коротко розглянути ключові етапи створення та налаштування підключення.

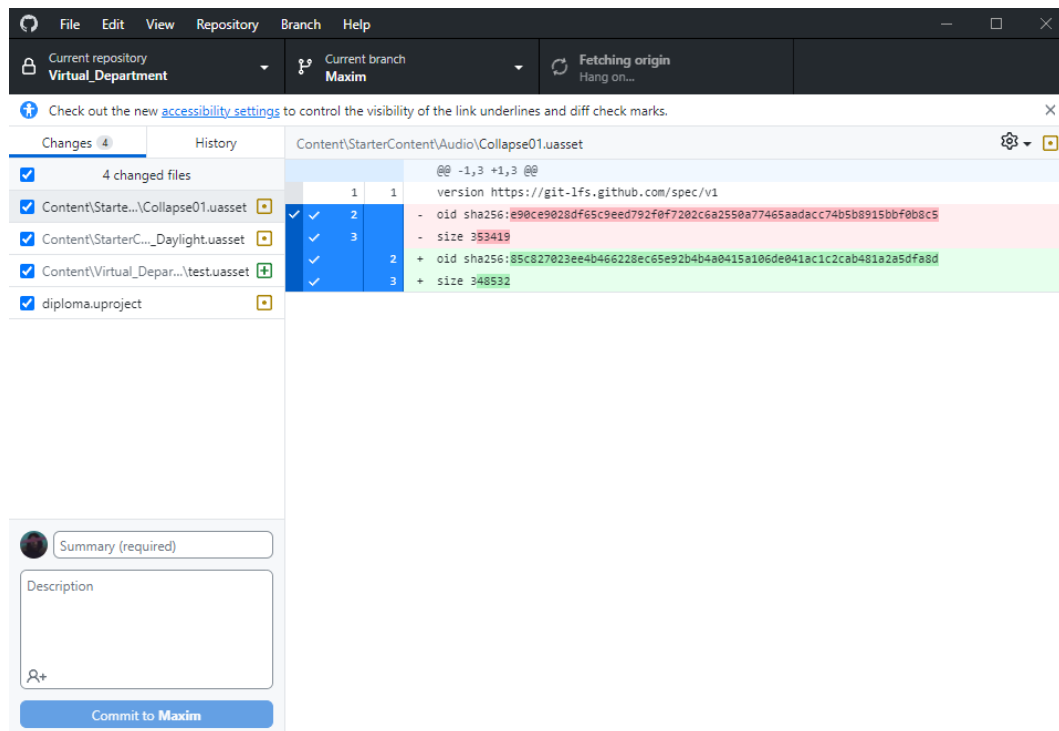


Рисунок 3.1 – Інтерфейс GitHub Desktop з підключеним репозиторієм

Першим кроком є реєстрація нового облікового запису на платформі GitHub та завантаження десктопного застосунку GitHub Desktop, який спрощує роботу з репозиторіями. Для роботи з файлами великого обсягу, такими як 3D-моделі або текстури, необхідно встановити Git LFS [32] – додаток, що дозволяє ефективно керувати великими файлами. Крім того, для інтеграції Git з Unreal Engine 5 потрібно встановити Git SCM [33], який забезпечує зв'язок між локальними файлами проекту та віддаленим репозиторієм.

Далі один із членів команди створює новий репозиторій на GitHub, де буде зберігатися весь код і супутні файли проекту. Репозиторій можна налаштувати

як приватний (для обмеженого доступу) або публічний (для відкритого доступу). Після створення репозиторію інші учасники отримують запрошення до співпраці, яке надсилається через GitHub, і підключаються до репозиторію. Для цього вони клонують репозиторій на свій локальний комп'ютер, завантажуючи копію всіх наявних файлів.

Даний підхід дозволяє всім учасникам команди одночасно працювати над проектом, завантажувати свої зміни до спільного сховища та зберігати історію змін.

Наступним кроком у процесі є завантаження та встановлення програми Blender (рис. 3.2). Вона готова до роботи одразу після встановлення і не потребує складних налаштувань. Однак для зручності роботи користувач може налаштувати гарячі клавіші відповідно до своїх уподобань або змінити візуальну частину інтерфейсу для кращої продуктивності.

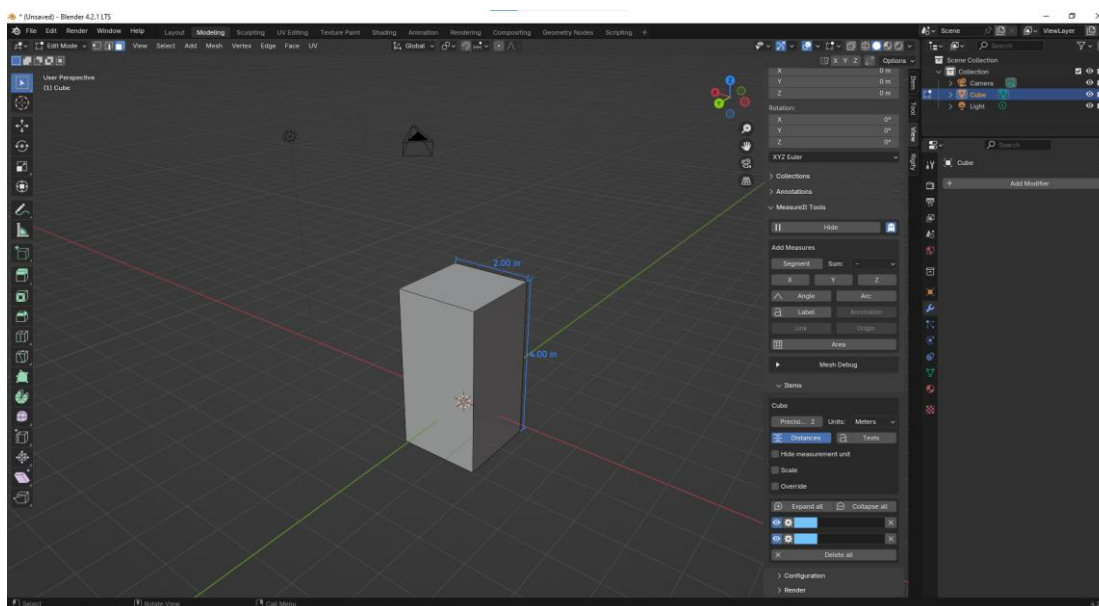


Рисунок 3.2 – Інтерфейс Blender з додатком MeasureIT

Також необхідно увімкнути підтримку експорту файлів у форматі .fbx у налаштуваннях аддонів. Це дозволить підготувати моделі для подальшого

імпорту в Autodesk Maya та Unreal Engine 5, забезпечуючи сумісність і зручність у роботі з тривимірними об'єктами на всіх етапах розробки.

Особливу увагу варто приділити точності моделювання, особливо якщо воно потребує роботи з реальними розмірами. У таких випадках рекомендується встановити додатковий плагін під назвою MeasureIT Tools [34]. Цей інструмент дозволяє точно вимірювати і позначати розміри ребер полігону, що робить його незамінним для проєктів, де важливі точність і масштабність. MeasureIT Tools значно спрощує процес створення моделей, адаптованих до реальних параметрів, і забезпечує високу якість результату.

Наступним кроком є встановлення програми Autodesk Maya (рис. 3.3) для створення анімацій і ригінгу об'єктів у віртуальному середовищі.

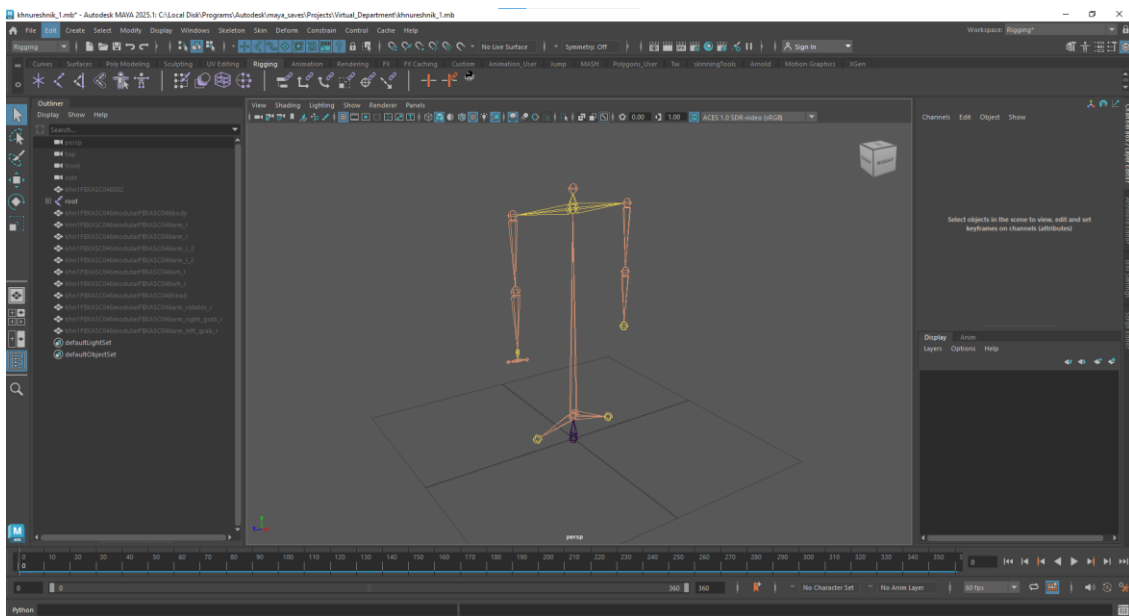


Рисунок 3.3 – Інтерфейс Autodesk Maya

Autodesk має багатофункціональне програмне забезпечення, яке надає широкий спектр інструментів для роботи з тривимірною графікою. Попри високу функціональність і комплексність Maya, у цьому проєкті вона не потребує спеціальних налаштувань. Вбудованого інструментарію програми буде цілком достатньо для виконання поставлених завдань. Maya забезпечує необхідні

можливості для створення якісних анімацій та точної обробки ригів, що робить її ідеальним вибором для таких завдань.

Далі слід встановити рушій Unreal Engine 5, який буде використовуватися для основної розробки. Важливим етапом є створення проєкту з підтримкою класів C++, оскільки у майбутньому виникне необхідність додавання власних елементів коду до проєкту.

На цьому етапі також варто одразу активувати необхідні плагіни для роботи з базою даних, зокрема для зчитування текстових файлів і забезпечення зручного перезавантаження проєкту після компіляції коду у Visual Studio. Також для майбутньої взаємодії з SQLite та OpenCV є потреба у плагінах OpenCV, SQLite, SQLite Support (рис. 3.4).

Для створення власних плагінів за допомогою C++ треба перейти у розділ Edit – Plugins та натиснути кнопку Add, де вибрати Blueprint Library (рис. 3.5).

Далі можна буде редагувати створений файл у середовищі Visual Studio.

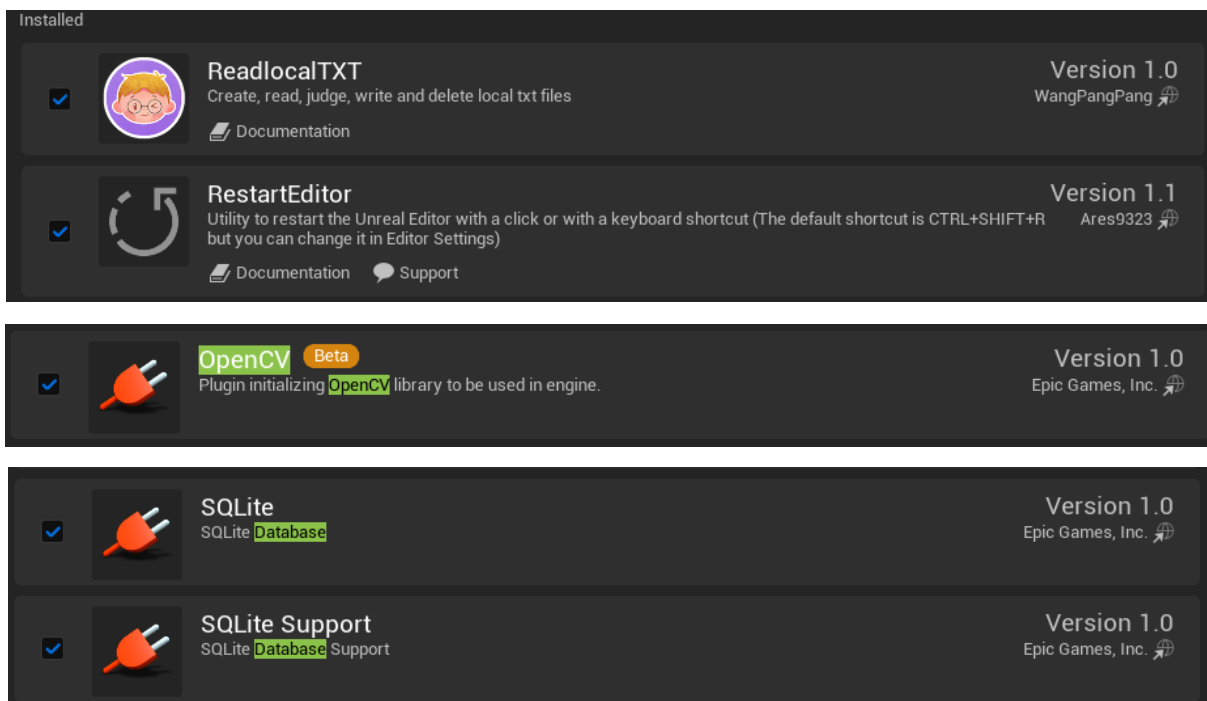


Рисунок 3.4 – Необхідні плагіни для подальшої роботи з UE5

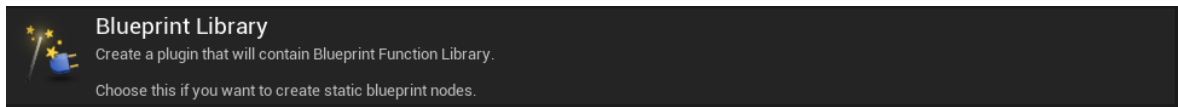


Рисунок 3.5 – Елемент для створення власних блоків коду UE5

Наступним етапом є встановлення додатку Visual Studio. На цьому етапі треба відразу вказати папку з розташуванням бібліотеки OpenCV та файлами SQLite у системі комп'ютера у розділі EDIT SYSTEM ENVIRONMENT VARIABLES (рис. 3.6).

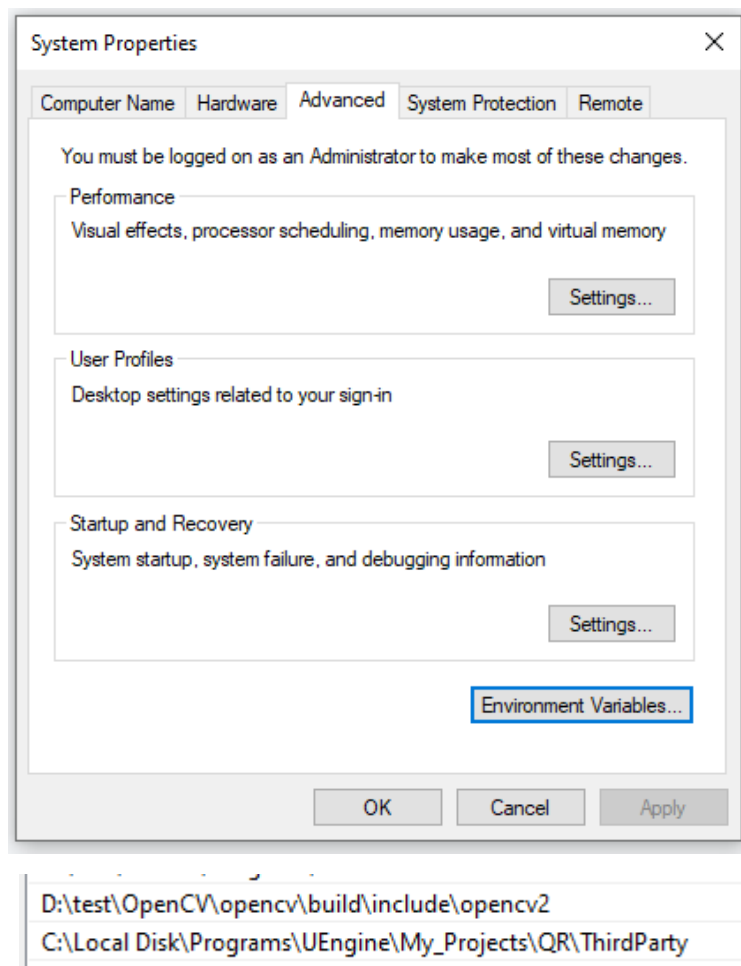


Рисунок 3.6 – Розділ System Properties для редагування змінних та шляхи розташування бібліотек

Під час встановлення Visual Studio (рис. 3.7) слід обрати додаткові компоненти для інсталяції (рис. 3.8), такі як .NET desktop development, Desktop development with C++, Game Development with C++, IDE support for Unreal Engine 5, а також Unreal Engine Test Adapter. Подальших дій не потрібно, оскільки основна частина коду та налаштувань буде автоматично згенерована Unreal Engine 5 після створення проєкту.

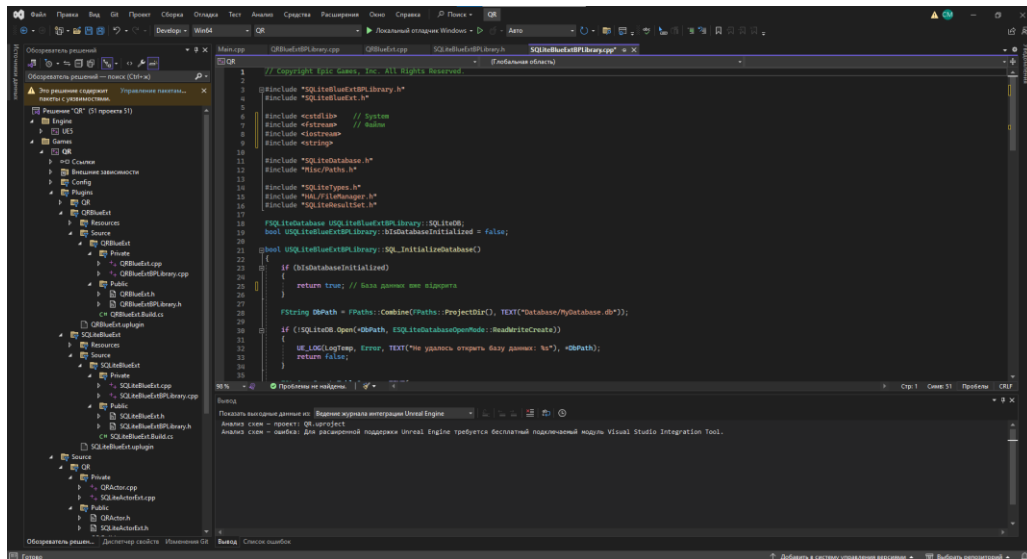


Рисунок 3.7 – Інтерфейс Visual Studio

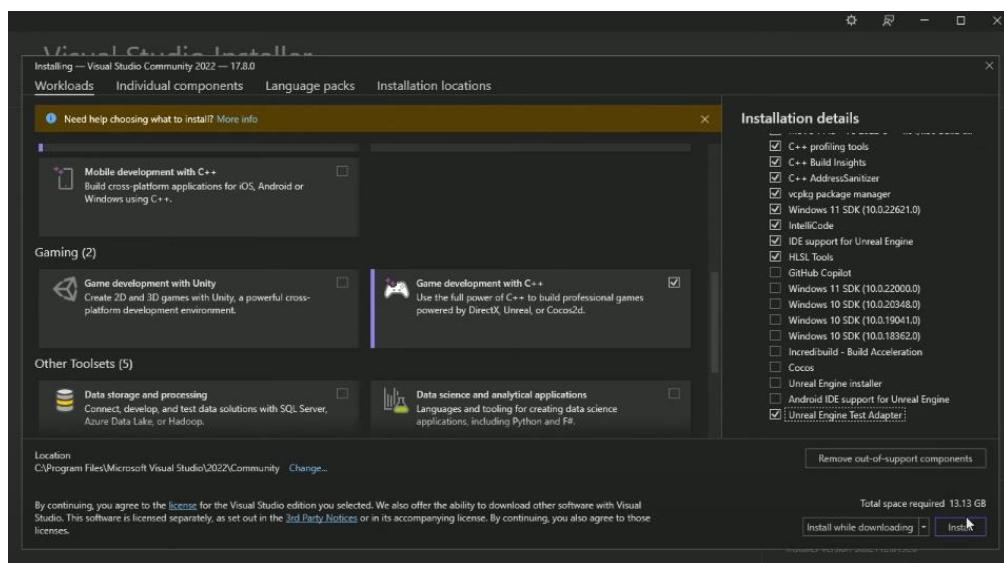


Рисунок 3.8 – Інсталятор Visual Studio

Для підключення бази даних SQLite до Unreal Engine 5 необхідно встановити раніше зазначені плагіни та вказати шлях до папки з файлами SQLite.dll та іншими супутніми файлами. Після встановлення плагінів і перезапуску проєкту слід відредагувати головний файл проєкту з розширенням .uproject, додавши відповідні рядки коду (рис. 3.9) за допомогою будь-якого текстового редактора. Після цього з'являється можливість використовувати бібліотеку SQLite у файлі .sln та взаємодіяти з нею через Visual Studio C++.

Для спрощення роботи з базою даних і спостереження за змінами у ній рекомендується встановити програму DB Browser for SQLite (рис. 3.10). Вона дозволяє відкривати файли з розширенням .db, переглядати дані, редагувати їх і легко керувати структурою бази, забезпечуючи зручний інтерфейс для роботи.

```
{
    "Name": "SQLiteCore",
    "Enabled": true
},
{
    "Name": "SQLiteSupport",
    "Enabled": true
},
{
    "Name": "DatabaseSupport",
    "Enabled": true
},
```

Рисунок 3.9 – Підключення SQLite до UE5

На цьому етапі налаштування програмного середовища завершується, і всі необхідні інструменти готові для подальшої роботи. Було встановлено та налаштовано основні програми, такі як Blender для моделювання, Autodesk Maya для ригу та анімацій, а також Unreal Engine 5 як основний рушій для розробки. Крім того, забезпечено інтеграцію бази даних SQLite та налаштовано середовище розробки у Visual Studio, зокрема з усіма потрібними компонентами для роботи з UE5.

Таке підготовлене середовище дозволяє ефективно виконувати комплексні завдання, пов'язані зі створенням віртуального середовища.

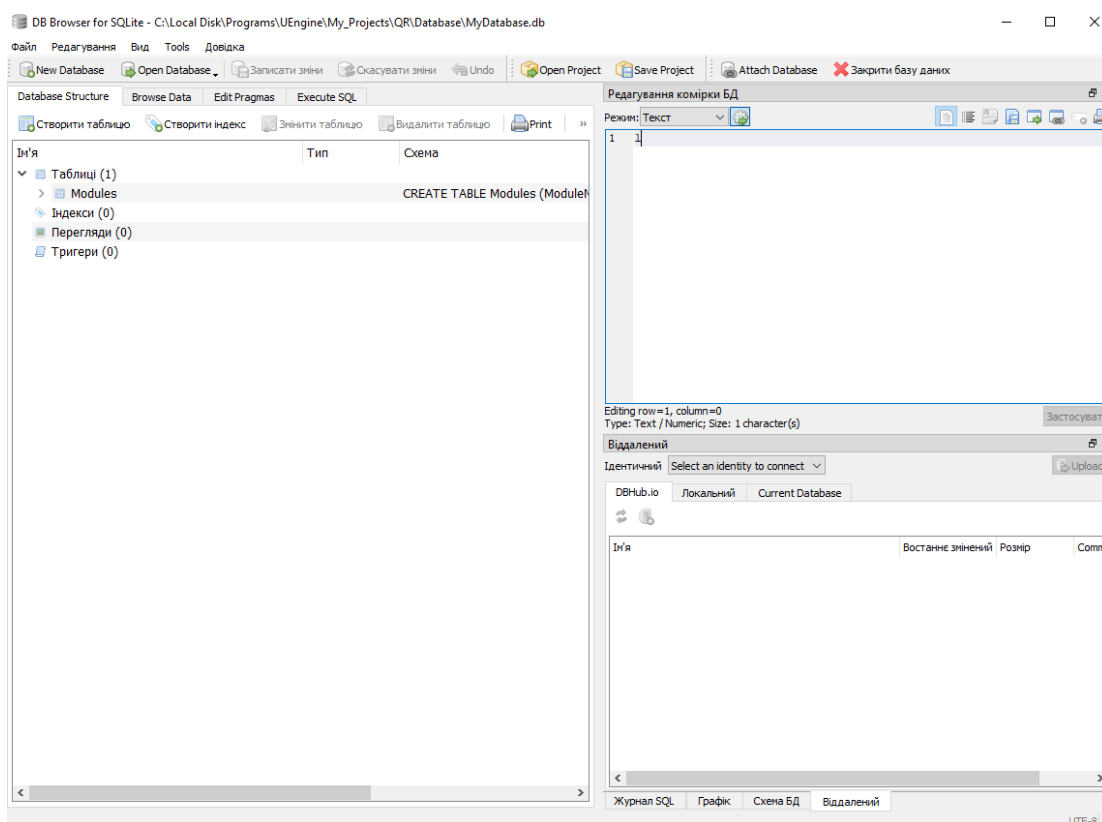


Рисунок 3.10 – Інтерфейс DB Browser

### 3.3 Моделювання, ригінг та анімація

Етап моделювання є ключовим у процесі створення віртуального середовища, адже саме на цьому етапі відбувається формування предметів, об'єктів, персонажів та інших фізичних елементів, що наповнюють віртуальний простір. Цей процес дозволяє відтворити об'єкти з високою точністю, створюючи їхні цифрові аналоги, які будуть використовуватися для подальшої взаємодії у віртуальному середовищі.

Розглянемо цей процес на прикладі моделювання головної установки – робота, призначеної для дослідження методів віртуальної реальності, її програмування та порівняння з реальним прототипом.

На першому етапі важливо визначити або розрахувати точні розміри моделі (рис. 3.11). Якщо існують креслення об'єкта, вони слугують базовою основою для роботи. Це дозволяє забезпечити відповідність розмірів і пропорцій моделі реальному прототипу, що є критично важливим для точності подальших досліджень.

Після визначення розмірів розпочинається полігональне моделювання. Цей процес передбачає побудову форм об'єкта, де кожен елемент механізму створюється як окрема частина. Це необхідно для подальшого етапу створення рігу та анімації, що дозволяє забезпечити коректну взаємодію між частинами механізму у віртуальному середовищі. Завдяки цьому підходу можна відтворити не лише зовнішній вигляд, але й функціональні особливості об'єкта.

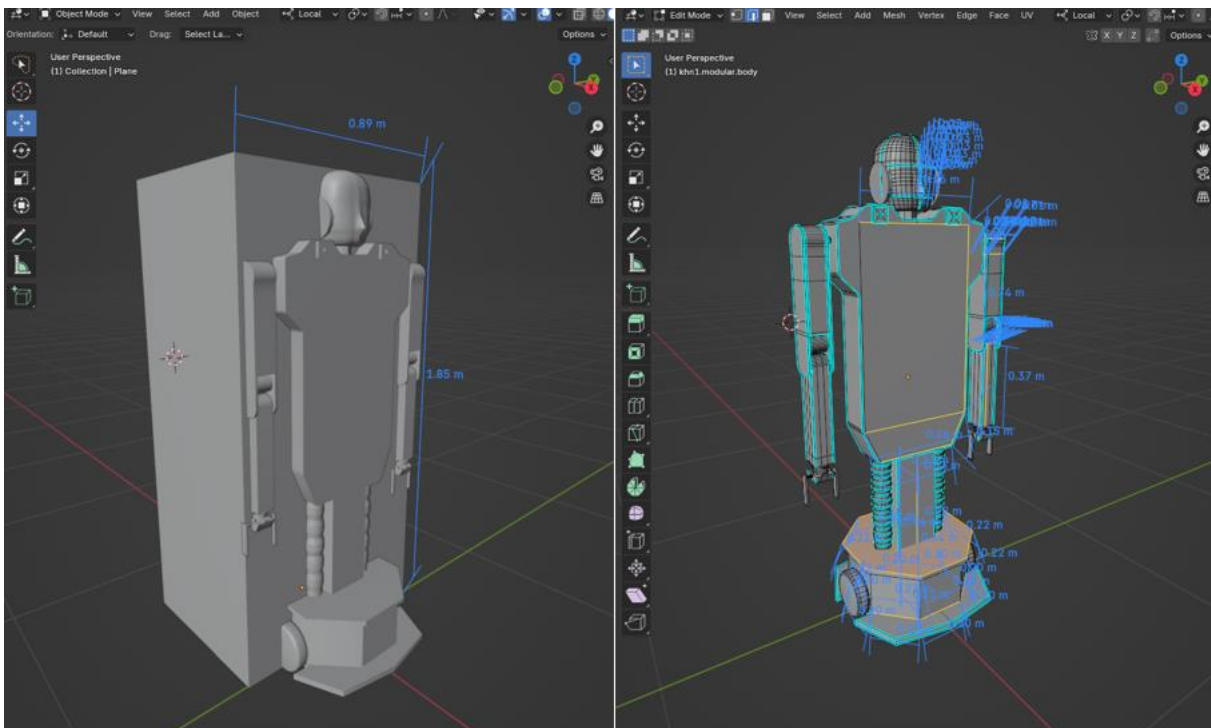


Рисунок 3.11 – Визначення розмірів моделі

Наступним кроком є створення UV-розгортки (рис. 3.12), яка необхідна для текстурювання моделі. UV-розгортка визначає, як текстури будуть накладатися на поверхню моделі, забезпечуючи її реалістичний вигляд. Після

завершення цього етапу модель текстурюється, додаються необхідні матеріали, і в результаті отримуємо готову цифрову модель (рис. 3.13), яку можна експортувати для подальшої роботи в інших програмних середовищах, у даному випадку Autodesk Maya.

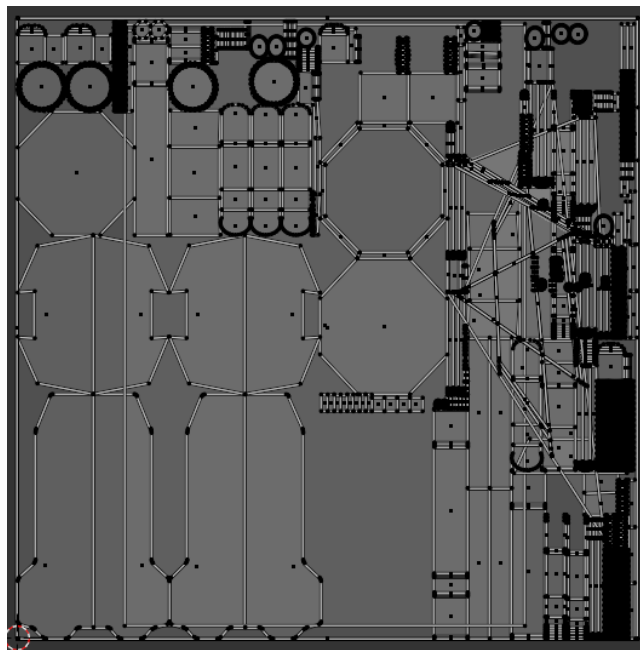


Рисунок 3.12 – UV-розгортка моделі

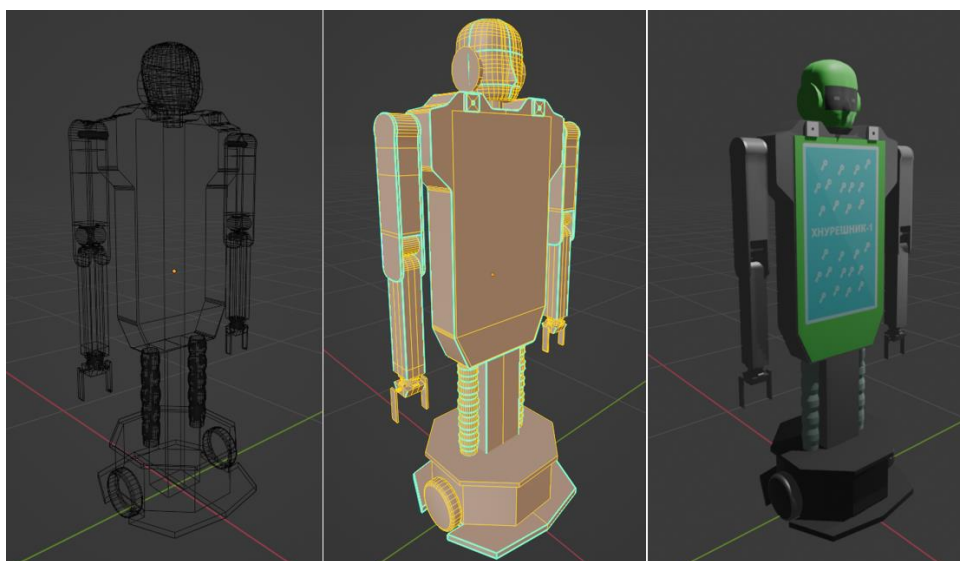


Рисунок 3.13 – Фінальний вигляд моделі

Процес експорту здійснюється через меню File – Export із вибором формату файлу, у даному випадку FBX (рис. 3.14). Цей формат забезпечує сумісність моделі з іншими програмами, зокрема з Unreal Engine 5.

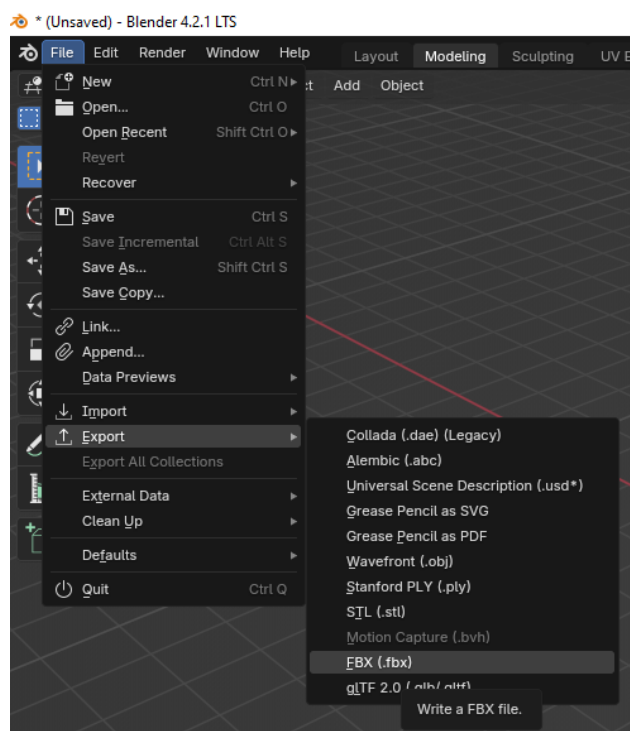


Рисунок 3.14 – Формат експорту

Під час експорту відкривається вікно налаштувань, де необхідно позначити певні параметри як на рис 3.15. Вони включають експорт текстур, матеріалів, колізій та триангульованих полігонів, що гарантує правильне відображення об'єкта в рушії. У разі необхідності текстури та матеріали можна буде змінити вже у середовищі Unreal Engine 5.

Наступним етапом йде визначення адекватності моделі [35], тобто процес оцінки, наскільки добре створена модель відповідає заданим цілям, умовам та реальним аналогам. Для цього використовують різні підходи та методики залежно від типу моделі (геометричної, фізичної, математичної тощо) та контексту її використання.

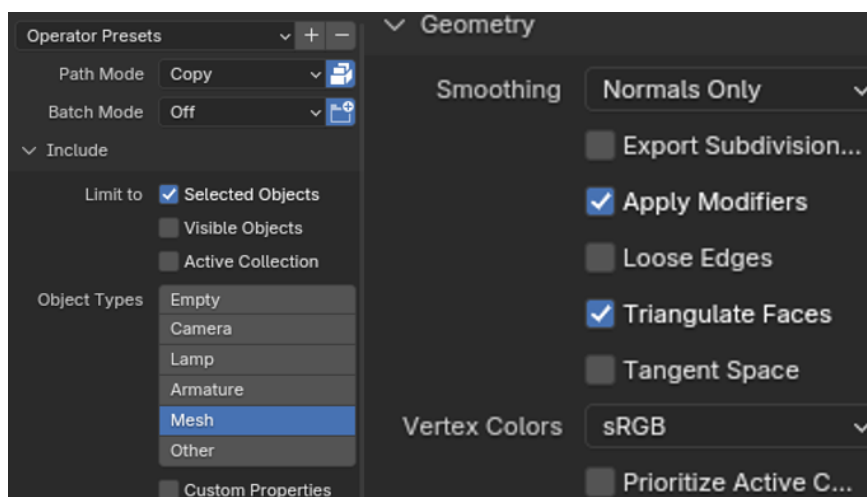


Рисунок 3.15 – Параметри експорту

У даному випадку, через відсутність точних вимог до моделі на поточному етапі, визначення її адекватності базується переважно на візуальній перевірці. Таким чином оцінка адекватності становить приблизно 80% відповідності моделі необхідним стандартам і цілям.

Візуальна перевірка включає аналіз зовнішнього вигляду моделі, її пропорцій, деталей та текстур. Вона допомагає переконатися, що модель виглядає природно, гармонійно вписується у віртуальне середовище та відповідає загальним очікуванням. У разі створення складних механізмів також можна перевірити коректність розташування основних частин, їх сумісність і взаємодію.

Хоча візуальна оцінка є важливим етапом, її результативність обмежується суб'єктивністю сприйняття. Для більш точної перевірки моделі варто залучати інші методи, зокрема тестування в Unreal Engine 5, щоб оцінити, як модель взаємодіє з іншими елементами сцени, та перевірити її поведінку під час симуляції чи анімації.

На наступному етапі роботи переходимо до створення скелетної анімації в середовищі Autodesk Maya. Після імпорту моделі в Maya необхідно побудувати ієрархію кісток (рис. 3.16), що формує основу для майбутнього скелету. Цей

процес дозволяє забезпечити можливість руху моделі під час анімації, оскільки кожна кістка відповідає за певну частину об'єкта та її переміщення.

Далі відбувається розміщення суглобів (джоїнтів) на ключових елементах моделі (рис. 3.17).

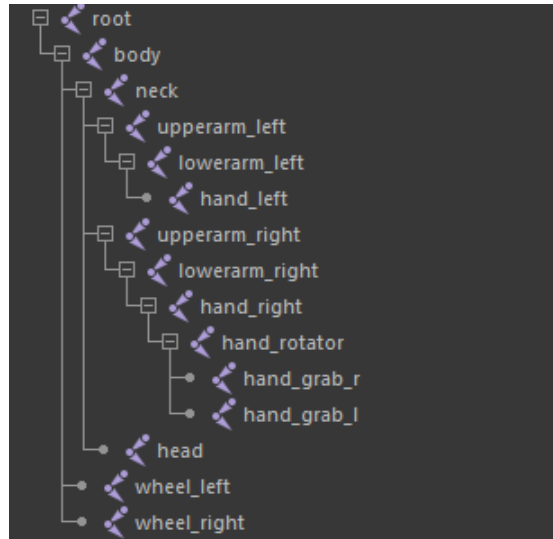


Рисунок 3.16 – Ієрархія кісток моделі

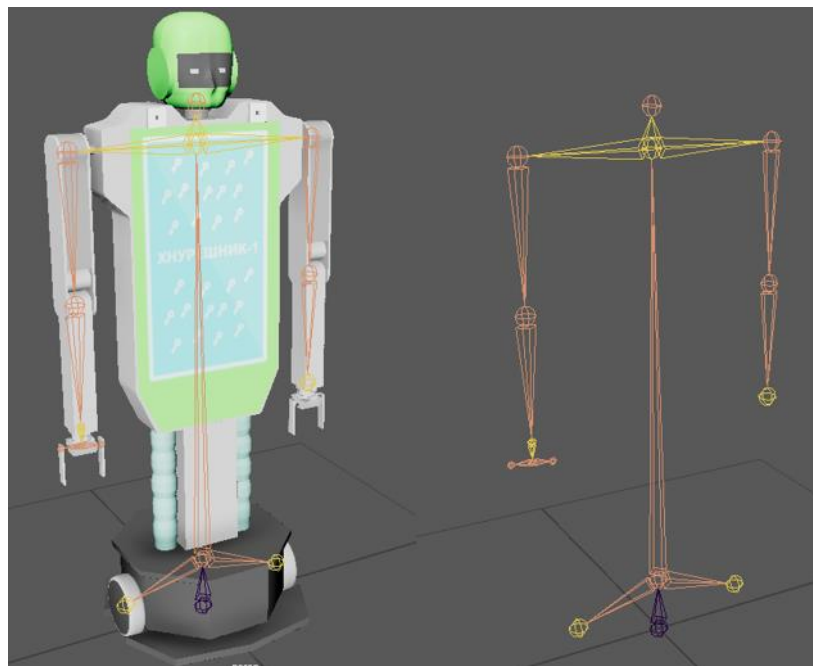


Рисунок 3.17 – Розміщення суглобів у ключових точках моделі

Важливо правильно визначити осі обертання та місця, де елементи моделі потребують переміщення, обертання чи інших видів взаємодії. Така точність допомагає забезпечити природність рухів та уникнути викривлень під час анімації.

Для спрощення роботи з анімацією потрібно оптимізувати параметри атрибутів кісток. Це досягається шляхом переміщення параметрів у матрицю, що дозволяє у Graph Editor відобразити графік зміни параметрів кісток – положення, обертання, масштабування тощо – таким чином, щоб усі криві починались з нульових координат.

Цей процес автоматизується за допомогою Python-скрипта, який виділяє необхідні кістки та вносить зміни у їх параметри. Результати виконання скрипта можна побачити на рис. 3.18. Код скрипта наведений на рис. 3.19. Такий підхід дозволяє зробити процес анімації більш ефективним і забезпечує плавність змін у рухах моделі.

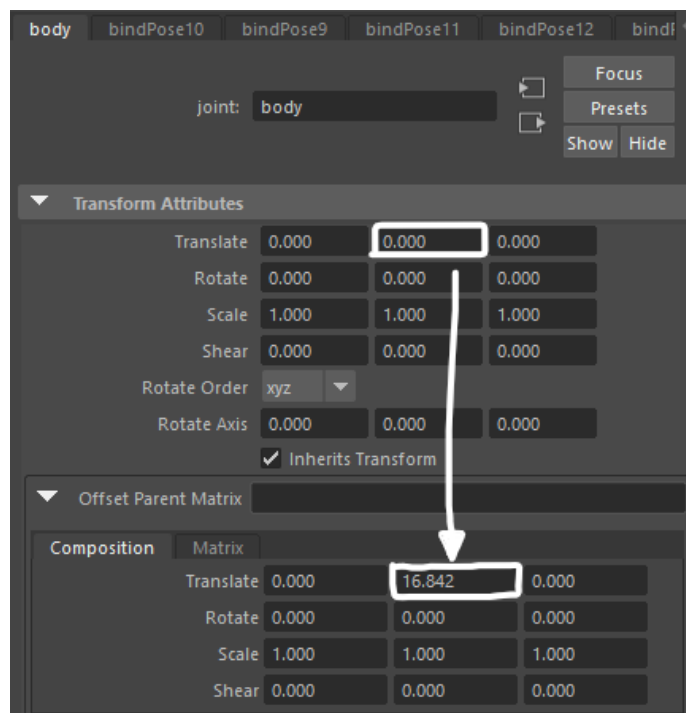


Рисунок 3.18 – Результат переміщення атрибутів у матрицю

```

import maya.api.OpenMaya as om
import maya.cmds as cmds

TRANSFORM_NODETYPES = ["transform", "joint"]

def has_non_default_locked_attributes(node):
    locked_attributes = []
    for attribute in ["translate", "rotate", "scale", "jointOrient"]:
        default_value = 1 if attribute == "scale" else 0
        for axis in "XYZ":
            if cmds.attributeQuery(attribute + axis, node=node, exists=True):
                attribute_name = "{}.{}{}".format(node, attribute, axis)
                current_value = cmds.getAttr(attribute_name)
                if cmds.getAttr(attribute_name, lock=True) and current_value != default_value:
                    return True

def reset_transforms(node):
    for attribute in ["translate", "rotate", "scale", "jointOrient"]:
        value = 1 if attribute == "scale" else 0
        for axis in "XYZ":
            if cmds.attributeQuery(attribute + axis, node=node, exists=True):
                attribute_name = "{}.{}{}".format(node, attribute, axis)
                if not cmds.getAttr(attribute_name, lock=True):
                    cmds.setAttr(attribute_name, value)

def bake_transform_to_offset_parent_matrix(node):
    if cmds.nodeType(node) not in TRANSFORM_NODETYPES:
        raise ValueError("Node {} is not a transform node".format(node))

    if has_non_default_locked_attributes(node):
        raise RuntimeError("Node {} has at least one non default locked attribute(s)".format(node))

    local_matrix = om.MMatrix(cmds.xform(node, q=True, m=True, ws=False))
    offset_parent_matrix = om.MMatrix(cmds.getAttr(node + ".offsetParentMatrix"))
    baked_matrix = local_matrix * offset_parent_matrix
    cmds.setAttr(node + ".offsetParentMatrix", baked_matrix, type="matrix")

    reset_transforms(node)

def bake_transform_to_offset_parent_matrix_selection():
    for node in cmds.ls(sl=True):
        bake_transform_to_offset_parent_matrix(node)

if __name__ == "__main__":
    bake_transform_to_offset_parent_matrix_selection()

```

Рисунок 3.19 – Скрипт для переміщення атрибутів у матрицю

Фінальним етапом у процесі створення скелету моделі є прив'язка кісток до її полігонів, що має назву «скінінг». Цей процес є ключовим для забезпечення коректного руху моделі під час анімації, оскільки саме завдяки скінінгу кожен полігон отримує зв'язок із відповідною кісткою. Це дозволяє моделі рухатися природно та без деформацій, забезпечуючи реалістичність її поведінки у віртуальному середовищі.

Скінінг може виконуватись як вручну, так і автоматично. Ручний метод особливо ефективний для складних моделей, де потрібно досягти максимальної

точності у прив'язці полігонів до кісток. Він дає змогу детально налаштувати вплив кожної кістки на певну частину моделі, що є важливим для складних механізмів або об'єктів із великою кількістю рухливих частин.

Однак у випадках, коли модель має простішу структуру або коли точність ручного налаштування не є критичною, можна скористатися автоматичним скінінгом, його параметри можна побачити на рисунку 3.20.

У цьому проєкті автоматичний підхід здатен ефективно впоратися із завданням, забезпечуючи швидкий та надійний результат. Завдяки використанню автоматичного скінінгу процес прив'язки значно прискорюється, що дозволяє зосередитись на інших аспектах роботи, таких як оптимізація руху або тестування анімацій.

Тому доречно перейти до етапу анімації, який є важливою складовою у створенні динамічних та реалістичних моделей. Анімація забезпечує рух об'єктів, персонажів або механізмів, дозволяючи відтворювати їх поведінку у віртуальному середовищі.

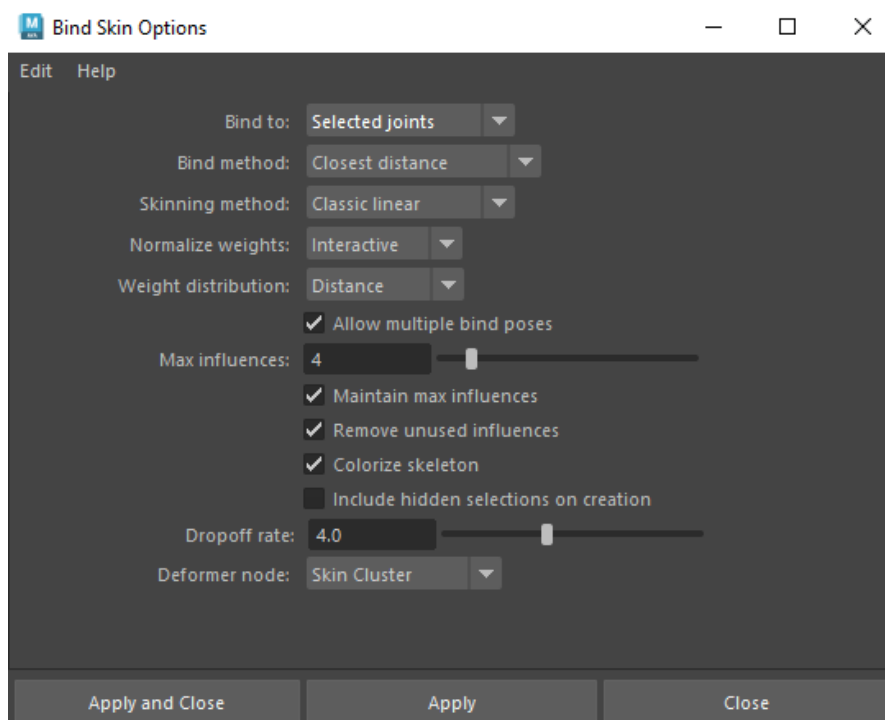


Рисунок 3.20 – Параметри скінінгу

Першим кроком на цьому етапі є створення ключових кадрів (keyframes). Це визначені положення або стани моделі у певний момент часу, які є основою для побудови анімаційної послідовності. У середовищі Maya ключові кадри створюються шляхом фіксації позиції, обертання чи масштабування обраних елементів моделі на часовій шкалі (Timeline). Наприклад, якщо потрібно анімувати підйом механічної руки, у першому кадрі фіксується її початкове положення, а в наступному – фінальне.

Далі використовується інтерполяція для заповнення проміжних станів між ключовими кадрами. Цей процес автоматизується за допомогою графічного редактора анімації (Graph Editor), де можна редагувати криві руху. Криві визначають плавність, швидкість та характер переміщення об'єкта. У Graph Editor параметри, такі як позиція, обертання та масштаб, представлені у вигляді графіків, що дозволяє гнучко налаштовувати анімацію (рис. 3.21).

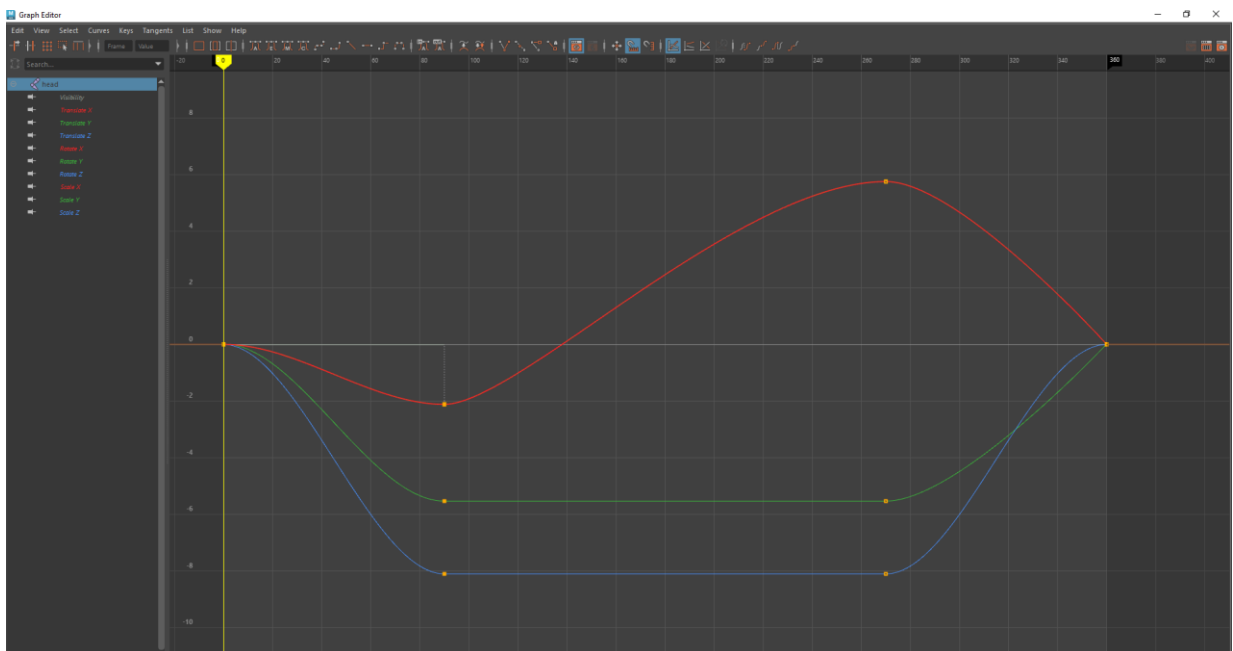


Рисунок 3.21 – Вікно редактора кривих руху

Після цього здійснюється тестування анімації. На цьому етапі аналізуються рухи об'єкта на відповідність задуму. Якщо модель рухається не

так, як планувалося, проводяться коригування кривих у Graph Editor або додаються додаткові ключові кадри для досягнення необхідного результату. Наприклад, якщо рух механізму виглядає занадто різким, криві обертання можна згладити, щоб забезпечити плавний перехід.

Приклад анімації опускання голови робота за 360 кадрів при швидкості 60 к/с можна побачити на рисунку 3.22.

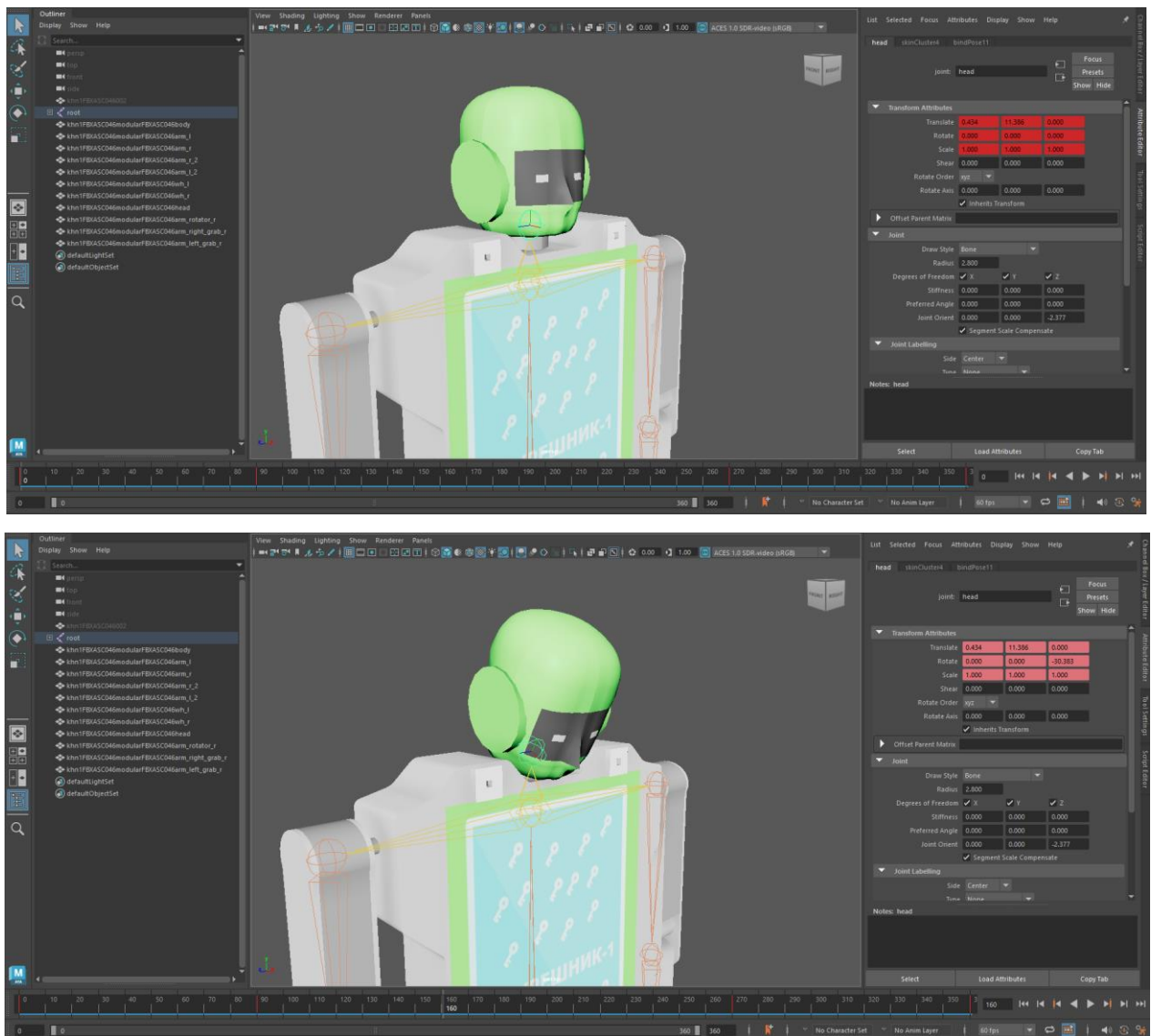


Рисунок 3.22 – Приклад анімації робота

Для складних анімаційних послідовностей можуть використовуватися додаткові техніки, такі як прив'язка (constraints), яка дозволяє автоматично

синхронізувати рухи одних елементів із іншими. Наприклад, при анімації механізму можна задати, щоб одна частина конструкції рухалася у відповідності до іншої.

Завершальним етапом є перевірка анімації у фінальному вигляді, включно із взаємодією з іншими об'єктами сцени. У цьому процесі важливо враховувати не лише рухи, але й швидкість виконання анімації, щоб вона гармонійно вписувалася у загальну динаміку сцени. Якщо результат задовільний, анімація експортується у форматі, який підтримується рушієм Unreal Engine 5, для подальшої інтеграції у проєкт, у даному випадку FBX.

### 3.4 Реалізація баз даних

SQL база даних використовується для зберігання великих обсягів даних, що постійно змінюються або потребують тривалого зберігання, таких як історія взаємодій користувачів, результати QR-сканувань, а також аналітика продуктивності робота.

Unreal Engine Data Table [36] найкраще підходить для зберігання статичних даних, що використовуються в рамках ігрової логіки та взаємодії робота з середовищем, таких як координати точок, шаблони для алгоритмів і поведінки робота, а також для тимчасових налаштувань сесії.

Зміст і тип даних мають вплив на вибір місця для їх зберігання: якщо дані повинні змінюватися в реальному часі та зберігатися після сесії, їх доцільно зберігати в базі даних. Якщо ж це статичні налаштування чи тимчасова інформація, вони можуть залишатися в Data Table UE5.

Отже буде доцільним встановити такі параметри бази даних User Data Table як ім'я користувача, кількість взаємодій з областю Check Box, кількість взаємодій з інтерактивним об'єктом, час взаємодії з об'єктом, успішність виконання задачі.

Параметрами бази даних Robot Data Table встановимо тип робота, що виконує дію, фактичні координати робота, цільові координати, інструкцію, вибір алгоритм руху, наявність перешкод на шляху, тривалість сесії.

Параметрами SQL бази даних встановимо ідентифікатор QR-коду, тип модуля, кількість запусків, кількість успішних та неуспішних сканувань, дату та час виконання сканування.

Також до додаткових параметрів симуляції, які можна зберігати та аналізувати, можна віднести рівень освітлення, температура модулів, стан батареї, рівень шуму та інші, однак вони не будуть використовуватися у даному проєкті, оскільки потребують точного налаштування та синхронізації з реальними аналогами.

Перейдемо до інтеграції локальної бази даних у проєкт [37]. Для цього створимо нову бібліотеку Blueprint Library типу Public у середовищі UE5. Це автоматично згенерує необхідні нам 2 файли формату .cpp у папці Private та 2 файли формату .h у папці Public, як показано на рисунку 3.23. Інші файли розглядати немає потреби, у даному випадку вони залишаються за замовчуванням.

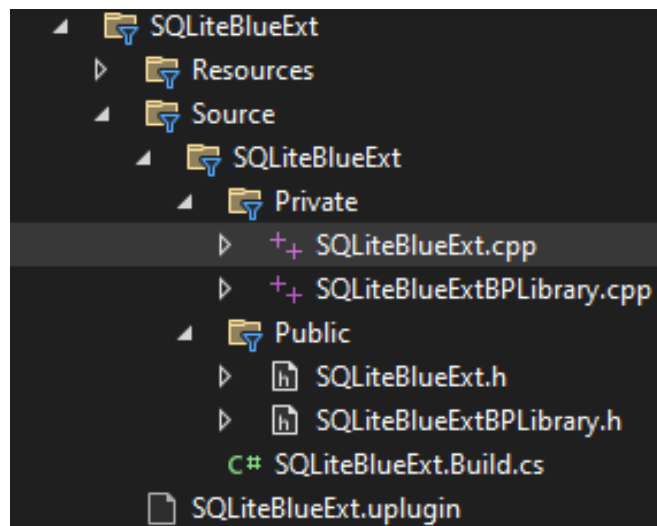


Рисунок 3.23 – Структура створеної Blueprint бібліотеки

На початковому етапі необхідно додати блок коду у файл SQLiteBlueExtBPLibrary.h, який відповідатиме за ініціалізацію бази даних, внесення параметрів та її завершення. Ці функції будуть доступні для виклику через спеціально створений вузол у Blueprint. Приклад відповідного коду наведено на рисунку 3.24.

```
#pragma once
#include "CoreMinimal.h"
#include "Kismet/BlueprintFunctionLibrary.h"
#include "SQLiteDatabase.h"
#include "SQLiteBlueExtBPLibrary.generated.h"

UCLASS()
class USQLiteBlueExtBPLibrary : public UBlueprintFunctionLibrary
{
    GENERATED_BODY()

private:
    static FSQLiteDatabase SQLiteDB;

    static bool bIsDatabaseInitialized;

public:
    UFUNCTION(BlueprintCallable)
    static bool SQL_InitializeDatabase();

    UFUNCTION(BlueprintCallable)
    static bool SQL_AddOrUpdateModuleInteraction(FString ModuleName);

    UFUNCTION(BlueprintCallable)
    static bool SQL_RecordModuleTime(FString ModuleName, FString Time);

    UFUNCTION(BlueprintCallable)
    static void SQL_CloseDatabase();
};
```

Рисунок 3.24 – Блок коду файлу SQLiteBlueExtBPLibrary.h

Даний код представляє клас USQLiteBlueExtBPLibrary, створений для роботи з базою даних SQLite у контексті Unreal Engine, із можливістю інтеграції в Blueprints. Клас успадковує функціонал від UBlueprintFunctionLibrary, що дозволяє використовувати його методи як вузли в Blueprints. Він містить статичний об'єкт бази даних FSQLiteDatabase та прапорець bIsDatabaseInitialized, який вказує, чи база даних була ініціалізована. Клас реалізує три функції:

SQL\_InitializeDatabase – ініціалізує базу даних. Функція є викликаємою через Blueprints.

SQL\_AddOrUpdateModuleInteraction – додає або оновлює інформацію про взаємодію з модулем у базі даних. Також доступна в Blueprints.

SQL\_RecordModuleTime – записує час обраної взаємодії.

SQL\_CloseDatabase – закриває базу даних.

Функції мають атрибуту BlueprintCallable, що дозволяє їх викликати безпосередньо з Blueprints. Код також передбачає використання метаданих для вузлів у Blueprints, таких як назва, ключові слова, категорії, але це частково описано у коментарях для створення інших подібних вузлів.

Наступним кроком є написання коду у файлі SQLiteBlueExtBPLibrary.cpp. Цей код буде реалізовувати взаємодію з базою даних SQLite у проєкті. Спочатку визначаються глобальні змінні для об'єкта бази даних та стану її ініціалізації (рис. 3.25).

```
// Copyright Epic Games, Inc. All Rights Reserved.
#include "SQLiteBlueExtBPLibrary.h"
#include "SQLiteBlueExt.h"

#include <cstdlib>
#include <fstream>
#include <iostream>
#include <string>

#include "SQLiteDatabase.h"
#include "Misc/Paths.h"

#include "SQLiteTypes.h"
#include "HAL/FileManager.h"
#include "SQLiteResultSet.h"

FSQLiteDatabase USQLiteBlueExtBPLibrary::SQLiteDB;
bool USQLiteBlueExtBPLibrary::bIsDatabaseInitialized = false;
```

Рисунок 3.25 – Блок коду визначення глобальних змінних для об'єкта бази даних та стану її ініціалізації

У функції SQL\_InitializeDatabase відбувається створення або відкриття бази даних за вказаним шляхом, а також створення таблиці з полями та назвами.

Код функцій SQL\_InitializeDatabase представлено на рисунку 3.26.

```

bool USQLiteBlueExtBPLibrary::SQL_InitializeDatabase()
{
    if (bIsDatabaseInitialized)
    {
        return true;
    }

    FString DbPath = FPaths::Combine(FPaths::ProjectDir(), TEXT("Database/MyDatabase.db"));
    SQLiteDB.Open(*DbPath, ESQLiteDatabaseOpenMode::ReadWriteCreate);

    FString CreateTableQuery = TEXT(
        "CREATE TABLE IF NOT EXISTS Modules ("
        "ModuleName TEXT PRIMARY KEY, "
        "Interactions INTEGER DEFAULT 0, "
        "LastInteractionTime TEXT);"
    );
    SQLiteDB.Execute(*CreateTableQuery);

    bIsDatabaseInitialized = true;
    return true;
}

```

Рисунок 3.26 – Блок коду функції SQL\_InitializeDatabase

Функція SQL\_AddOrUpdateModuleInteraction (рис. 3.27) додає новий запис або оновлює існуючий, збільшуючи кількість взаємодій для конкретного модуля. Вона використовує запити SELECT, UPDATE та INSERT для роботи з даними.

Функція SQL\_CloseDatabase (рис. 3.28) закриває базу даних і скидає статус ініціалізації, забезпечуючи коректне завершення роботи з нею. Код включає перевірки, щоб уникнути помилок, і логування для відстеження стану операцій.

Для активації попередньо увімкнених плагінів у середовищі Unreal Engine 5 для взаємодії з бібліотекою SQLite треба у файл SQLiteBlueExt.Build.cs додати такий блок коду, як на рисунку 3.29.

Також у кодї наявний блок для запису часу активності робота, код функції SQL Add Or Update Module Interaction майже аналогічний до функції SQL\_RecordModuleTime. У SQL\_RecordModuleTime обробляється текстове значення LastInteractionTime, яке є рядком (FString), представляючи час у певному форматі. Функція оновлює або вставляє значення для LastInteractionTime.

```

bool USQLiteBlueExtBPLibrary::SQL_AddOrUpdateModuleInteraction(FString ModuleName)
{
    const TCHAR* SelectQuery = TEXT("SELECT Interactions FROM Modules WHERE ModuleName = :ModuleName LIMIT 1");
    FSQLitePreparedStatement SelectStatement;
    SelectStatement.Create(SQLiteDatabase, SelectQuery, ESQLitePreparedStatementFlags::Persistent);
    SelectStatement.SetBindingValueByName(TEXT(":ModuleName"), ModuleName);

    if (SelectStatement.Execute() && SelectStatement.Step() == ESQLitePreparedStatementStepResult::Row)
    {
        int32 CurrentInteractions = 0;
        SelectStatement.GetColumnValueByName(TEXT("Interactions"), CurrentInteractions);

        int32 NewInteractions = CurrentInteractions + 1;
        const TCHAR* UpdateQuery = TEXT("UPDATE Modules SET Interactions = :NewInteractions WHERE ModuleName = :ModuleName");
        FSQLitePreparedStatement UpdateStatement;
        UpdateStatement.Create(SQLiteDatabase, UpdateQuery, ESQLitePreparedStatementFlags::Persistent);
        UpdateStatement.SetBindingValueByName(TEXT(":NewInteractions"), NewInteractions);
        UpdateStatement.SetBindingValueByName(TEXT(":ModuleName"), ModuleName);
        UpdateStatement.Execute();
    }
    else
    {
        const TCHAR* InsertQuery = TEXT("INSERT INTO Modules (ModuleName, Interactions) VALUES (:ModuleName, 1)");
        FSQLitePreparedStatement InsertStatement;
        InsertStatement.Create(SQLiteDatabase, InsertQuery, ESQLitePreparedStatementFlags::Persistent);
        InsertStatement.SetBindingValueByName(TEXT(":ModuleName"), ModuleName);
        InsertStatement.Execute();
    }

    return true;
}

```

Рисунок 3.27 – Блок коду функції SQL\_AddOrUpdateModuleInteraction

```

void USQLiteBlueExtBPLibrary::SQL_CloseDatabase()
{
    if (bIsDatabaseInitialized)
    {
        SQLiteDatabase.Close();
        bIsDatabaseInitialized = false;
    }
}

```

Рисунок 3.28 – Блок коду функції SQL\_CloseDatabase

```

PublicDependencyModuleNames.AddRange(
    new string[]
    {
        "Core",
        // ... add other public dependencies that you statically link with here ...
        "DatabaseSupport",
        "SQLiteCore",
        "SQLiteSupport",
    }
);

```

Рисунок 3.29 – Блок коду файлу SQLiteBlueExt.Build.cs

Тепер ми маємо можливість викликати створені функції у середовищі Unreal Engine, що представлено на рисунку 3.30.

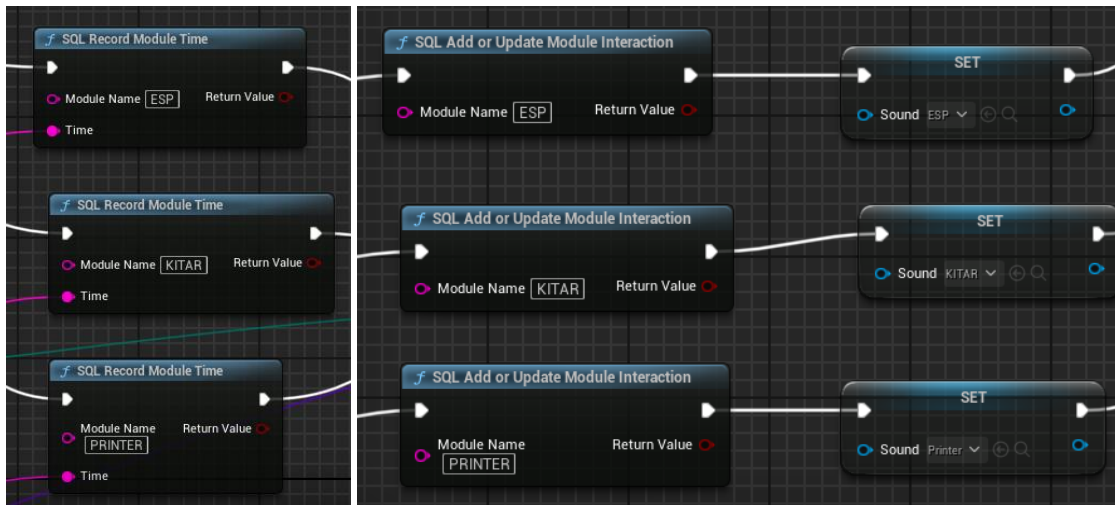


Рисунок 3.30 – Реалізовані функції у середовищі UE5

Тепер перейдемо до створення Data Table в Unreal Engine 5. Для цього скористаємося функцією на рисунку 3.31, попередньо створивши структуру таблиці, в процесі якої можемо задати назви елементів та типи даних, які вони будуть зберігати.

На рисунку 3.32 представлено три основні рядки таблиці: Name, Number та Success, з типами даних String, Integer та Boolean відповідно. Вкладка Default Value дозволяє задати значення змінних за замовчуванням для кожного елемента.

Для взаємодії з таблицею використовуємо блоки коду Blueprint, що зображені на рисунку 3.33.

За допомогою цих блоків можна додавати або викликати змінні, які зберігаються у таблиці. Основними нодами для цього є Make\_User\_Structure та Break\_User\_Structure, які використовуються у зв'язці з іншими нодами для коректної роботи з даними таблиці.

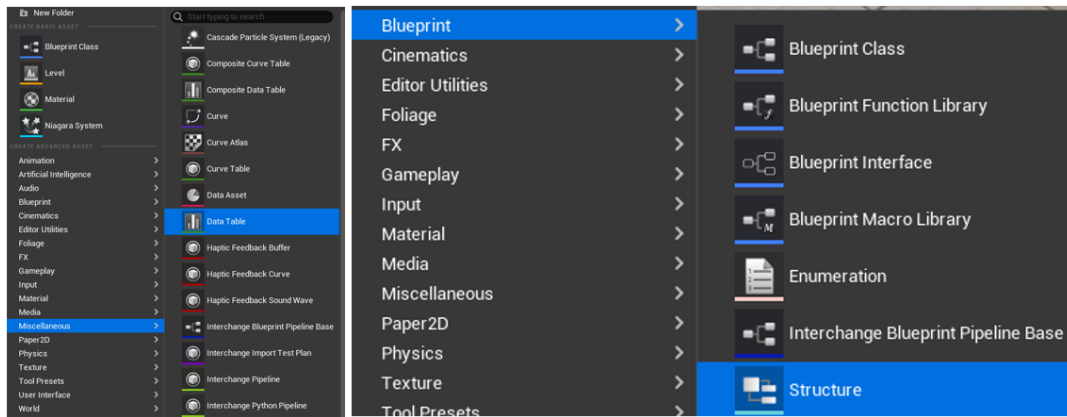


Рисунок 3.31 – Функції для створення шаблону Data Table та її структури

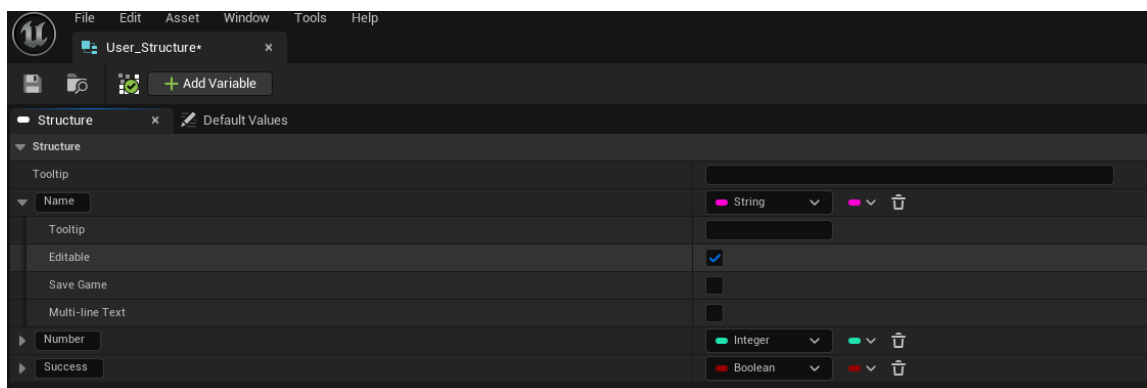


Рисунок 3.32 – Структура Data Table

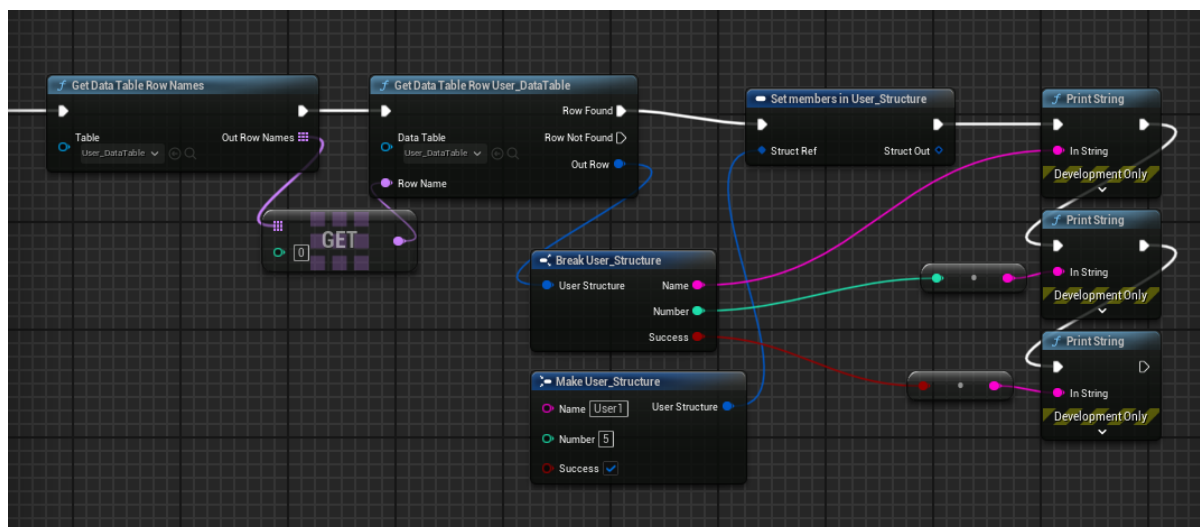


Рисунок 3.33 – Блок Blueprint для взаємодії з Data Table

### 3.5 Реалізація систем зі застосуванням штучного інтелекту

Штучний інтелект (AI) може бути використаний для аналізу різних параметрів роботи робота та інших, які зберігаються у базі даних.

AI здатний оптимізувати маршрути, коригувати алгоритми для зменшення помилок, прогнозувати затримки сигналів і покращувати роботу. Крім того, AI може адаптувати роботу до змін середовища, передбачати проблеми, та автоматично коригувати стратегії для підвищення ефективності. На виході система може забезпечити точнішу роботу робота, зменшити кількість помилок, а також надати прогнози та рекомендації для подальшого покращення продуктивності.

У даній частині роботи зосередимося на одному з фінальних етапів аналізу зібраної інформації.

Штучний інтелект може отримувати дані з різних джерел, зокрема з бази даних, і проводити їх детальний аналіз. На основі цих даних AI здатний виявляти закономірності, прогнозувати можливі проблеми та оцінювати ефективність роботи системи.

В результаті аналізу AI виводить результати у вигляді текстових звітів, що включають висновки про продуктивність роботи, рекомендації щодо оптимізації його роботи, а також прогнози щодо подальшої роботи та необхідності технічного обслуговування.

У даному проєкті використовується koboldcpp-1.81.1 [38] для аналізу та обробки текстової інформації (рис. 3.34) та модель Llama (рис. 3.35) у форматі ggml та версії Llama-2-13B-chat-GGML / llama-2-13b-chat.ggmlv3.q2\_K.bin [39], якої буде достатньо у даному випадку.

KoboldCPP представляє з себе оптимізовану для локального використання версію штучного інтелекту, яка працює з великими мовними моделями, а Llama є сім'єю мовних моделей від Meta, що забезпечує високоякісну генерацію тексту та розуміння мови.

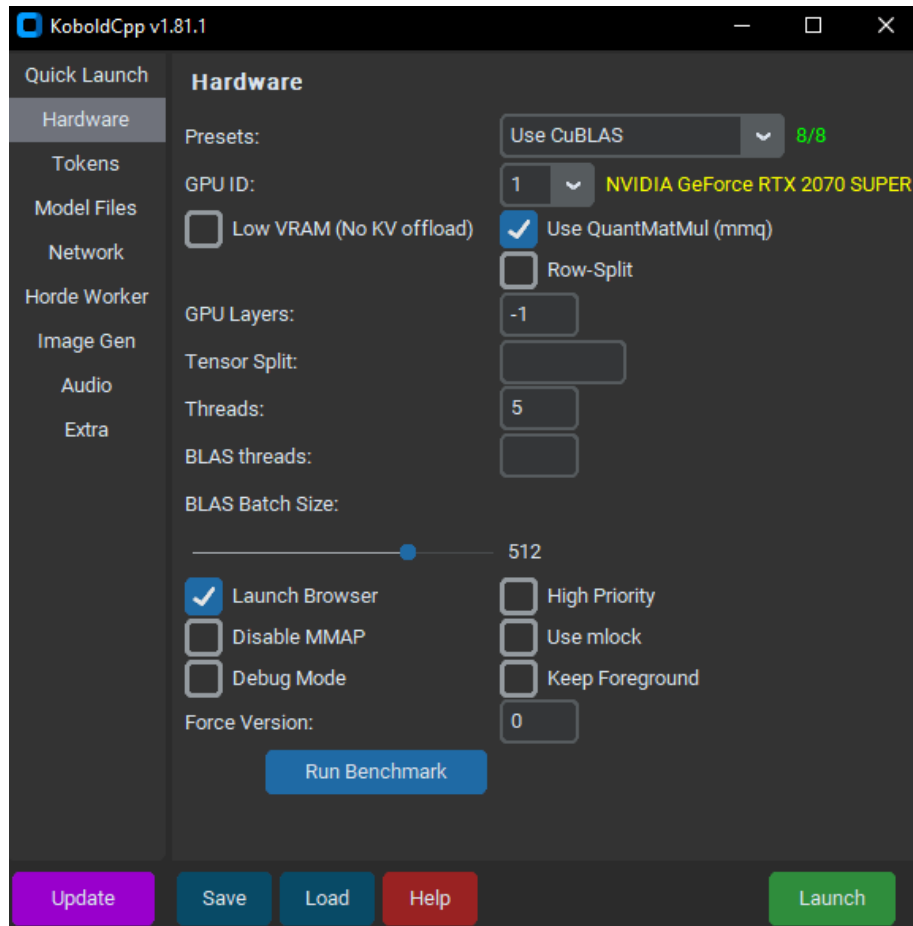


Рисунок 3.34 – Інтерфейс KoboldCPP



Рисунок 3.35 – Обрана модель репозиторію для генерації тексту для KoboldCPP

KoboldCPP підтримує можливість розподілу моделей на GPU/CPU по шарах, що дає змогу перенести частину навантаження на графічний процесор. Це дозволяє пришвидшити роботу моделі та зменшити використання оперативної пам'яті. Якщо у вас є відеокарта Nvidia, ви можете ефективно розподілити частину процесів на GPU для кращої продуктивності. Для цього необхідно вибрати пресет CuBLAS у лаунчері та встановити кількість шарів, які ви хочете призначити для обробки на відеокарті.

Оскільки програма KoboldCPP має відкритий вихідний код на C++, інтеграція її в інші системи, зокрема з Unreal Engine, передбачає внесення змін як до вихідного файлу C++ проєкту KoboldCPP, так і до коду в Unreal Engine. Для цього необхідно додати функціональність, яка дозволяє Unreal Engine передавати дані в KoboldCPP, а сама програма повинна мати можливість приймати ці дані для подальшої обробки. Однак через обмеження часу та ресурсів, було вирішено обмежитися розглядом ручного використання програми без автоматизованої взаємодії з рушієм. В результаті, користувач буде здійснювати передачу даних та запуск програми вручну, без необхідності інтеграції на рівні коду, проте з майбутнім оновленням проєкту стане можливим інтеграція і такого функціоналу, який був описаний вище.

Розглянемо детальніше процес взаємодії з KoboldCPP. Для початку необхідно вказати шлях до завантаженої текстової моделі (рис. 3.36) у відповідному полі після запуску програми. Інші параметри можна залишити за замовчуванням, оскільки вони вже налаштовані для стандартної роботи.

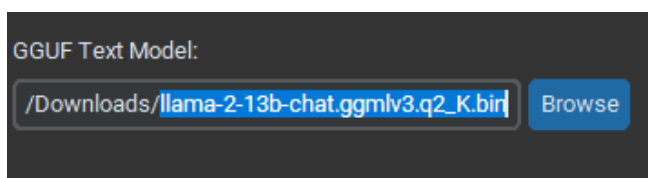


Рисунок 3.36 – Pole для вводу локального шляху до файлу текстової моделі

Після запуску програми відкривається вікно браузера за локальною адресою `http://localhost:5001/`, де користувач може вставити необхідний текст (рис. 3.37).

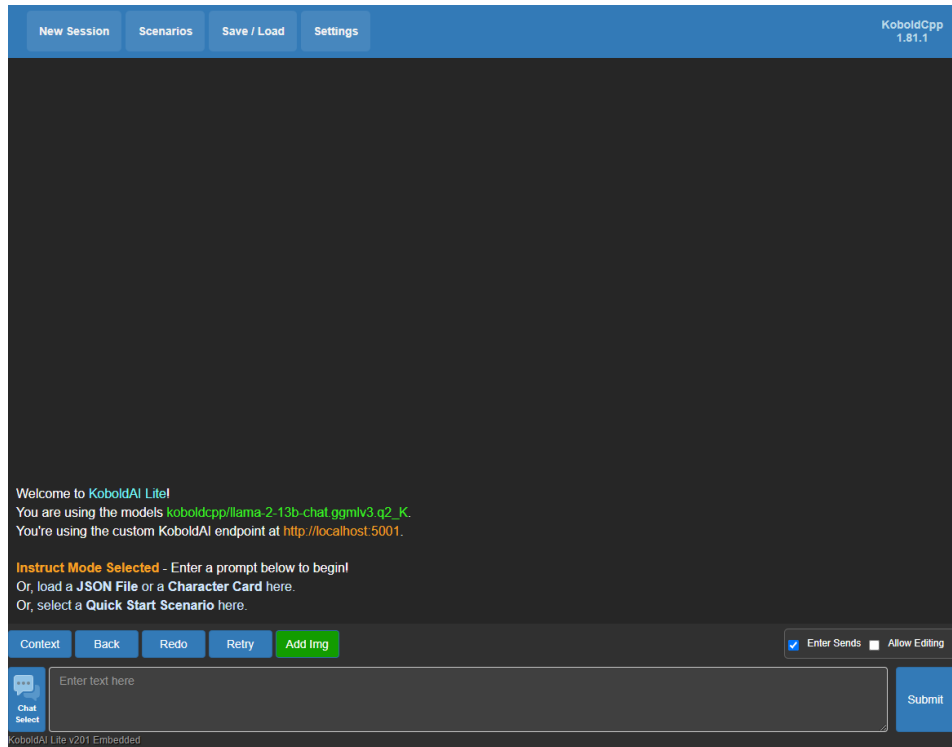


Рисунок 3.37 – Поле для вводу локального шляху до текстової моделі

У цьому вікні вводиться вхідний запит, який потім обробляється програмою, і на екран виводиться відповідь.

Вихідний результат можна налаштовувати відповідно до потреб користувача, змінюючи параметри виводу для досягнення бажаного рівня точності або формату відповіді.

### 3.6 Тестування програмної частини

На даному етапі важливо перевірити працездатність усіх попередніх розробок та їх інтеграцію в систему. Почнемо з перевірки роботи бази даних. Для цього запускаємо C++ проєкт, підключаємо необхідні ноди та ініціюємо

симуляцію процесу. Після запуску перевіряємо, чи правильно виконуються всі функції, взаємодіють з даними та здійснюються запити. Про успішне виконання цих операцій свідчать відповідні рядки дебагінгу, які відображаються в верхньому лівому куті екрана (рис. 3.38). Це дає змогу підтвердити, що всі функції бази даних працюють коректно, і система взаємодіє з нею без помилок.

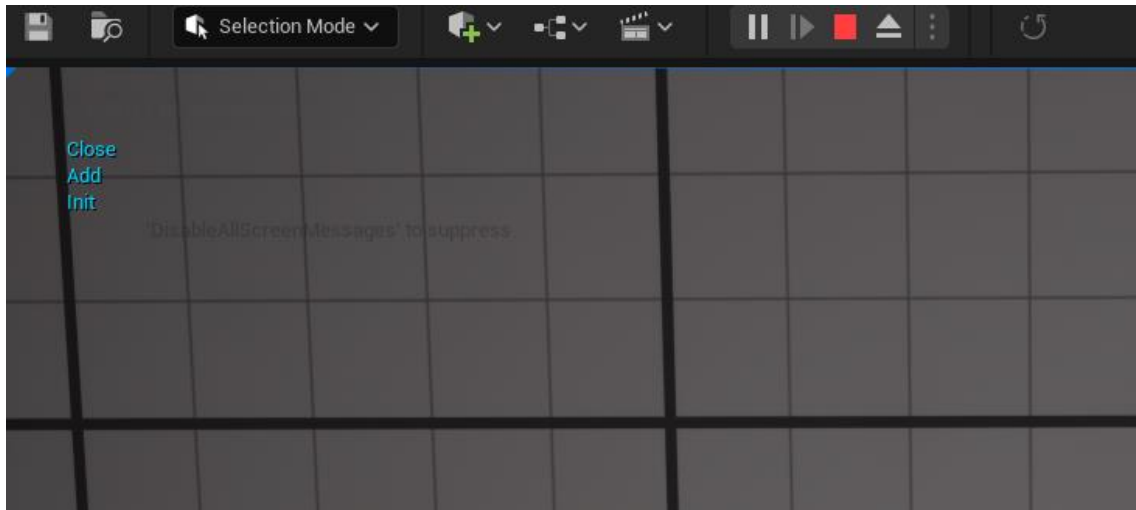


Рисунок 3.38 – Результати дебагінгу у UE5

Наступним кроком буде перевірка виконання функції запису параметрів у базу даних. На рисунку 3.39 можна побачити, що був створений новий файл бази даних за шляхом `UEngine\My_Projects\<<name>>\Database`.

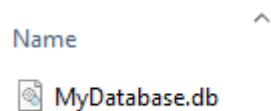


Рисунок 3.39 – Результати дебагінгу у UE5

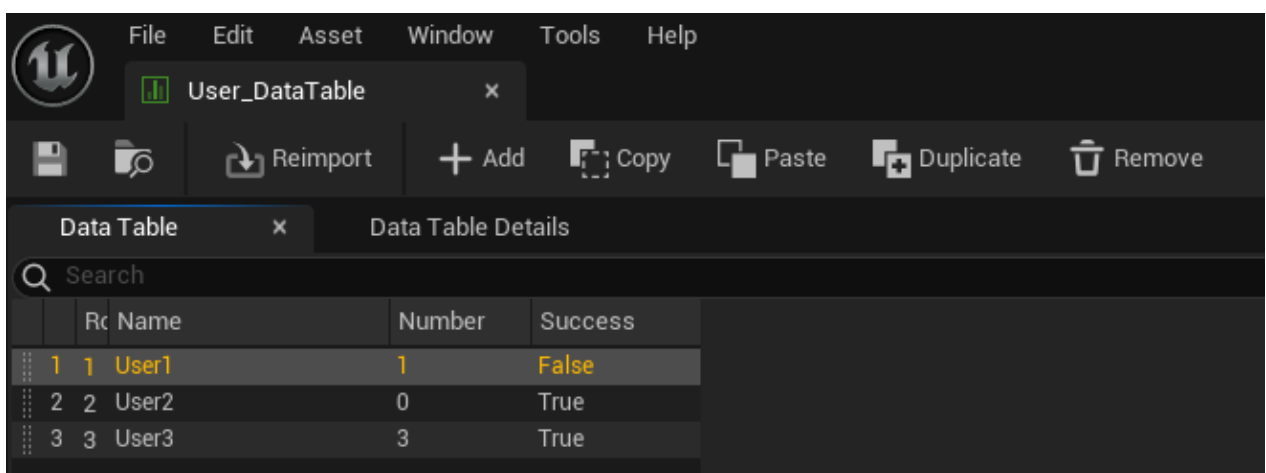
Для перевірки збереженої інформації всередині бази даних використовуємо програму DB Browser for SQLite. Результати наведені на рисунку 3.40.

	ModuleName	Interactions	LastInteractionTime
	Фільтр	Фільтр	Фільтр
1	ESP	2	4.076096
2	KITAR	2	7.088946
3	PRINTER	1	5.159304

Рисунок 3.40 – Результати дебагінгу у UE5

Як видно, кількість взаємодій з тестовим модулем була успішно передана та збережена в базі даних.

Щодо Data Table у UE5 треба переконатися у справності передачі даних. Дані можна вносити як вручну так і через будь-який блок взаємодії з користувачем. На рисунку 3.41 бачимо, що дані успішно зберігаються у таблиці. Поле Number демонструє кількість взаємодій користувача з певним модулем, а Success – успішність взаємодії. За потреби їх можна використовувати у майбутній фінальній версії проекту. За допомогою даних таблиць також можна зберігати відомості про користувача для подальшої оптимізації та налаштування середовища під конкретну особу чи параметри.



	Rc	Name	Number	Success
1	1	User1	1	False
2	2	User2	0	True
3	3	User3	3	True

Рисунок 3.41 – Дані User Data Table

Далі буде доцільно перевірити працездатність штучного інтелекту текстової моделі. Для цього вводимо необхідні дані в поле введення тексту, після чого отримуємо очікуваний результат, який продемонстровано на рисунку 3.42. Результат є коректним та відповідає вимогам, повністю задовольняючи поставлені завдання.

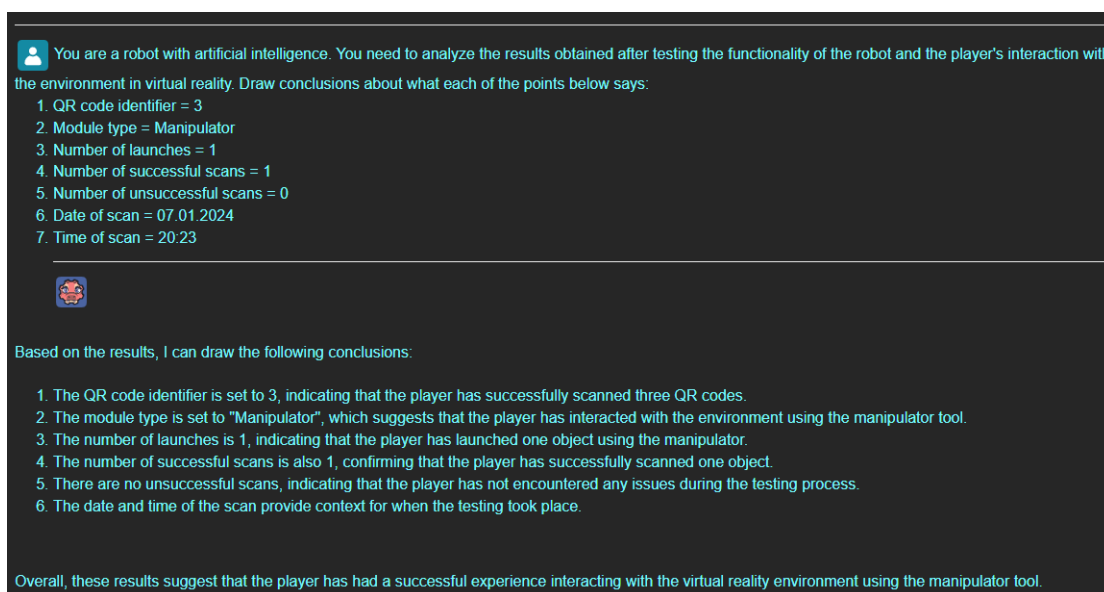


Рисунок 3.42 – Результати роботи текстової моделі ШІ

Після проведених перевірок та тестувань можна з упевненістю сказати, що всі етапи реалізації були успішно завершені. База даних працює належним чином, а штучний інтелект продемонстрував правильні результати. Всі поставлені цілі досягнуті, а система функціонує згідно з поставленими вимогами.

### 3.7 Розрахункова частина

На підставі отриманих результатів досліджень та проведених експериментальних випробувань можна провести розрахунки наведені нижче. Ці розрахунки стають основою для подальшого аналізу, на якому базуються

рекомендації щодо оптимізації або удосконалення існуючих методів, а також для прогнозування результатів у подібних умовах.

Крім того, існує можливість проведення додаткових різноманітних розрахунків на основі зібраних даних, що відкриває нові варіанти для аналізу досліджуваних об'єктів. Ці розрахунки дозволяють вивчати взаємозв'язки між параметрами віртуального середовища, його об'єктів та фізичними аналогами, а також проводити перевірку та оцінку впливу різних факторів на кінцеві результати.

Для оцінки часу продуктивності користувачів (3.1) припустимо, що віртуальний робот обробляє запити студентів у віртуальній лабораторії. Зібрані дані відповіді на 5 запитів займали наступний час (у секундах): 1.2, 1.5, 1.8, 1.0, 1.3.

$$T = \frac{\sum_{i=1}^n T_i}{n} \quad (3.1)$$

де  $T_i$  – час виконання  $i$ -м студентом;  
 $n$  – кількість студентів.

$$T = \frac{1.2 + 1.5 + 1.8 + 1.0 + 1.3}{5} = \frac{6.8}{5} = 1.36 \text{ с}$$

Отже, середній час відповіді робота становить 1.36 секунди. Це показує, що робот працює досить швидко і відповідає в режимі реального часу, що підходить для освітніх задач. Якщо допустимий час відповіді визначений як <2 секунди, робот функціонує ефективно.

Далі можемо оцінити точність роботи штучного інтелекту (3.2). Припустимо, ШІ аналізує відповіді студентів у віртуальній лабораторії. Зібрані дані:  $T_P = 9$ ,  $T_N = 8$ ,  $F_P = 1$ ,  $F_N = 2$ .

$$A = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (3.2)$$

де  $T_P$  – правильно класифіковані правильні відповіді;  
 $T_N$  – правильно класифіковані неправильні відповіді;  
 $F_P$  – неправильно класифіковані правильні відповіді як помилки;  
 $F_N$  – неправильно класифіковані помилкові відповіді як правильні.

$$A = \frac{9 + 8}{9 + 8 + 1 + 2} = \frac{17}{20} = 0.85$$

Отже, ефективність роботи ШІ дорівнює 0.85 або 85%. Точність свідчить про високу ефективність алгоритму. Для навчального процесу це є прийнятним результатом, але можна вдосконалити, наприклад, шляхом додаткового навчання моделі.

Також можна провести розрахунок розміру бази даних (3.3) та прогнозу збільшення бази даних за рік використання після умовного впровадження (3.4).

Припустимо, в базі даних є таблиця з розділами типу String та Integer, яка містить 100 заповнених рядків, що відповідає результатам взаємодії 100 користувачів. Загальний розмір файлу становить 12 КБ, кожен запис додає ще близько 43.84 байт.

$$S = \sum_{i=1}^n (N_{r_i} \cdot S_{r_i}) + S_{DB} \quad (3.3)$$

де  $N_{r_i}$  – кількість заповнених рядків  $i$ -ї таблиці;  
 $S_{r_i}$  – розмір кожного рядка  $i$ -ї таблиці;

$S_{DB}$  – загальний розмір файлу порожньої бази даних.

$$S = (100 \cdot 0,04384) + 12 = 16,384 \text{ КБ}$$

Отримуємо, що загальний обсяг бази даних складає 16,384 КБ. Це досить компактний обсяг для навчальної системи, який може бути легко керований стандартними інструментами баз даних.

Тепер припустимо, що щодня додається 30 нових записів, кожен із розміром 43.84 байт.

$$S_f = S + (N_{rd} \cdot 365 \cdot S_{rd}) \quad (3.4)$$

де  $N_{rd}$  – кількість нових записів за день;

$S_{rd}$  – розмір одного запису.

$$S_f = 16,384 + (30 \cdot 365 \cdot 0,04384) = 496,432 \text{ КБ}$$

Через рік обсяг бази даних зросте до 496,432 КБ. Це важливо враховувати при плануванні ресурсів для зберігання даних.

Отже, на основі зібраних даних можна проводити різноманітні розрахунки в залежності від типу даних наявних у віртуальному середовищі, базі даних чи на основі фізичних властивостей об'єктів та цілей, які необхідно досягти. Це може включати статистичні обчислення, такі як визначення середнього значення, медіани, дисперсії та стандартного відхилення, а також побудову графіків для візуалізації інформації. Можна також застосовувати методи прогнозування для передбачення майбутніх значень. Для оптимізації можна шукати найбільш ефективні рішення в задачах, пов'язаних із зниженням витрат чи покращенням процесів. Якщо потрібно, можна застосовувати техніки машинного навчання або обробляти результати аналізу даних штучним інтелектом. Також можна проводити технічні розрахунки, що стосуються

конкретних процесів, наприклад, енергетичних витрат системи чи порівнянню потужності елементів фізичної установки з віртуальною.

### 3.8 Охорона праці

Забезпечення збереження здоров'я та життя, захист від різноманітних небезпек, а також створення комфортних умов для нормальної життєдіяльності є базовими факторами. Безпека охоплює широкий спектр аспектів, починаючи від фізичного та психологічного захисту до забезпечення прав і свобод людини в усіх сферах життя [40].

«Діяльність» – це специфічна форма активного ставлення людини до світу, яка дозволяє їй змінювати його і перетворювати. Вона є метою, інструментом, процесом і результатом людського існування. Діяльність здійснюється в повсякденному житті, роботі, наукових, військових, соціальних та культурних сферах. Без задоволення базових потреб, зокрема потреби в безпеці, людина не зможе втілити свої соціальні потреби, потребу в признанні та самовираженні. Для забезпечення цього процесу в організмі людини відбувається складний біологічний процес, що підтримує гарне здоров'я та працездатність – життєдіяльність. Таким чином, безпека життєдіяльності (БЖД) передбачає комфортну і безпечну взаємодію людини з навколишнім середовищем, включаючи природне, техногенне та соціально-політичне оточення.

Для створення безпечного стану довкілля, виробництва, побутових умов для комфортної життєдіяльності людини необхідне ефективне правове забезпечення – законодавство з безпеки життєдіяльності, яке ґрунтується на Конституції України.

Законодавство з охорони праці включає: Закон України «Про охорону праці», Кодексу законів про працю, Закон України «Про загально- обов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності» та прийнятих до них нормативно-правових актів з охорони праці.

Для реалізації принципів у сфері охорони праці на кожному конкретному підприємстві, виробництві, організації, компанії тощо, необхідно дотримуватися за допомогою законодавчих правових і нормативних актів:

- суцільного технічного контролю за станом виробництва, технології та продукції;
- сприянню підприємствам, організаціям, компаніям у створенні безпечних та нешкідливих умов праці.

Безпека життєдіяльності (БЖД) людини складається з:

- БЖД в умовах виробничої діяльності людини;
- БЖД в повсякденних умовах побуту;
- БЖД в надзвичайних ситуаціях і екстремальних умовах.

Система «Людина-Машина-Середовище» має дві цілі:

Перша ціль – в процесі діяльності людина прагне досягти позитивного результату.

Друга ціль – в процесі діяльності людина прагне уникнути небажаних наслідків цієї діяльності для свого здоров'я. З позицій БЖД головною в системі Л-М-С є безпека людини.

Виконаємо декомпозицію системи Людина-Машина-Середовище, використовуючи функціонально-структурний підхід і представляючи «людину» трьома функціональними частинами:

- Л1: людина, що виконує певні цілеспрямовані дії (управління тех. устаткуванням, прибирання квартири, приготування сніданку і ін.);
- Л2: людина, що розглядається з погляду безпосереднього впливу на середовище (тепло- і волого виділення, споживання кисню і ін.);
- Л3: людина з погляду свого психофізіологічного стану під впливом зовнішніх факторів (утомленість, неприємні відчуття при жаркій погоді і ін.);
- М1: основне технологічне устаткування (комп'ютер, верстат, лабораторна установка та ін.);

– М2: устаткування, що виконує функції аварійного захисту, виключає виникнення небезпечних чинників (захисне заземлення, занулення, автоматичне вимкнення);

– М3: вплив технологічного устаткування на виробниче середовище і людину тобто виникнення шкідливих виробничих чинників, (підвищення концентрації пилю в приміщенні, зміна параметрів мікроклімату).

Також важливо дотримуватися основних правил техніки безпеки при роботі з комп'ютером. Для безпеки і для безперебійної роботи обладнання, слід пам'ятати кілька важливих рекомендацій. По-перше, важливо підтримувати правильну позу та забезпечити ергономічність робочого місця. Не забувайте про регулярні перерви, щоб уникнути перенапруження, розім'ятися та дати відпочити очам. Також важливо робити короткі перерви щогодини: встаньте, розтягніться та погляньте на віддалені об'єкти для зменшення навантаження на зір. Регулярно створюйте резервні копії важливих даних і файлів, щоб убезпечити себе від можливих втрат у разі поломки обладнання чи інших непередбачуваних ситуацій. Рекомендується використовувати зовнішні жорсткі диски або хмарні сховища для додаткової безпеки. Будьте обережні під час перегляду інтернет-сторінок і відкриття листів з вкладеннями. Уникайте переходу за сумнівними посиланнями та завантаження файлів з ненадійних джерел. Для захисту від шкідливих програм використовуйте антивірусне програмне забезпечення, яке регулярно оновлюйте. Після завершення роботи з комп'ютером вимикайте його належним чином, щоб уникнути пошкодження даних і системи. Також обережно ставтеся до комп'ютерного обладнання: не залишайте поблизу рідин чи їжі, щоб уникнути розливу, і перевірте, чи не перетираються та не пошкоджені кабелі.

Дотримуйтеся ліцензійних угод, використовуючи тільки ліцензоване програмне забезпечення, щоб забезпечити безпеку та законність вашої роботи з комп'ютером.

### 3.9 Висновки до третього розділу

У третьому розділі було детально розглянуто процес розробки програмної частини віртуального середовища, орієнтованого на створення інтерактивної віртуальної лабораторії. Основним інструментом розробки обрано Unreal Engine 5, який забезпечує широкий функціонал для створення високоякісного тривимірного середовища, інтеграції штучного інтелекту та роботи з базами даних.

На етапі аналізу засобів розробки було визначено ключові переваги UE5, зокрема його можливості для створення інтерактивних симуляцій. Завдяки інтеграції з Visual Studio та використанню мови програмування C++ було забезпечено ефективність реалізації основних функціональних компонентів системи.

Розробка програмної частини передбачала кілька ключових етапів. По-перше, створення 3D-моделей об'єктів віртуальної лабораторії та їх інтеграція в середовище. По-друге, створення бази даних для збору, зберігання та обробки інформації про діяльність користувачів і результати експериментів. А також впровадження алгоритмів штучного інтелекту для автоматизації аналізу даних і надання рекомендацій користувачам.

Крім того, було проведено тестування програмної частини, функціональне та навантажувальне тестування системи, яке дозволило виявити та усунути технічні помилки, оптимізувати роботу алгоритмів і покращити взаємодію користувачів із системою.

Наприкінці було розглянуто аспекти охорони праці та безпеки при використанні віртуальної лабораторії.

Таким чином, у третьому розділі успішно реалізовано всі заплановані завдання щодо створення програмної частини віртуального середовища. Результати роботи закладають основу для впровадження інтерактивної віртуальної лабораторії, яка забезпечує сучасний рівень навчання та значно покращує доступність і якість освітнього процесу.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи була вивчена предметна область 3D-технологій віртуального простору, особливості інтеграції технологій штучного інтелекту, методів створення баз даних у різних галузях науки та промисловості. Було проаналізовано види 3D-моделювання, зв'язок 3D з інженерним програмами, шляхи впровадження технологій з використанням 3D у сфери навчання та виробництва. Були виявлені переваги та недоліки даного впровадження.

У рамках роботи проведено аналіз існуючих віртуальних лабораторій та інструментів, які використовуються для створення інтерактивних освітніх середовищ. Вивчено методи використання баз даних для зберігання та обробки великих обсягів даних, пов'язаних з навчальними процесами та експериментами. Досліджено можливості оптимізації взаємодії між користувачами та віртуальним середовищем за допомогою інтеграції штучного інтелекту, який забезпечує автоматизацію аналізу даних та надання рекомендацій.

Було виконано наступні завдання:

- розроблено концепцію віртуальної лабораторії з визначенням основних функцій та можливостей, включаючи штучний інтелект та бази даних для підтримки експериментів та аналізу даних;
- проведено аналіз та створено структуру бази даних для зберігання інформації про експерименти, результати досліджень та дані користувачів;
- визначено вимоги до платформи, а саме використання Unreal Engine 5 з C++ для створення десктопного застосунку;
- розроблено алгоритм дій для створення віртуальної лабораторії, включаючи етапи проектування, програмування, тестування та популяризації серед потенційних користувачів;
- проведено ключові розрахунки та тестування функціональності елементів віртуальної лабораторії;

У результаті була досягнута основна мета роботи: розробка інтерактивної віртуальної лабораторії та інтеграція у неї DB та AI для аналізу даних, яка надає студентам і дослідникам доступ до віртуальних експериментів та системи допомоги та відгуку. Реалізація цього проекту буде сприяти розвитку онлайн-освіти, забезпечуючи інноваційний підхід до проведення наукових експериментів у віртуальному середовищі, що дозволяє ефективно навчатися та виконувати дослідження незалежно від місця знаходження.

Також треба зазначити, що дана робота безпосередньо пов'язана з ціллю «Якісна освіта» із Цілей Сталого Розвитку, проголошених ООН. Розробка інтерактивної віртуальної лабораторії сприяє забезпеченню доступу до інноваційних освітніх технологій для студентів та викладачів, незалежно від їхнього місця проживання чи матеріальних можливостей. Віртуальні лабораторії дозволяють усім учасникам навчального процесу отримати якісні знання, розвивати практичні навички та проводити наукові дослідження в умовах, наближених до реальних.

Проект забезпечує зменшення нерівності в доступі до освітніх ресурсів, сприяючи формуванню рівних можливостей для здобуття освіти. Крім того, віртуальне середовище значно скорочує витрати на обладнання та матеріали, що є важливим для навчальних закладів із обмеженим бюджетом.

Інтеграція технологій 3D, AI та DB дозволяє створити середовище для персоналізованого навчання, де кожен студент отримує підтримку, відповідно до своїх потреб і темпів навчання. Це сприяє формуванню більш гнучкої, інтерактивної та доступної системи освіти, яка відповідає сучасним викликам глобалізації та цифровізації. Реалізація таких проектів також стимулює розвиток цифрової грамотності, критичного мислення та інноваційного підходу до вирішення завдань, що є ключовими аспектами якісної освіти.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва»; «Комп'ютеризовані та робототехнічні системи» / Упоряд.: Ігор Невлюдов, Роман Артюх, Володимир Безкоровайний, Наталія Демська, Владислав Євсєєв, Олександр Филипченко, Олександр Цимбал. Харків: ХНУРЕ, 2023. 56 с.

3. Аналіз можливостей впровадження AI та 3D-технологій у різні галузі навчання / М.В. Склярів, К.А. Тарасенко, О.М. Цимбал / Збірник матеріалів конференції «Цифрові інновації та сталий розвиток \ Digital innovations and sustainable development» [Електронний ресурс] – <https://tapr.nure.ua/dijalnist-kafedri/naukova-robota/digital-innovations-and-sustainable-development> – Харків: ХНУРЕ, 2024. с. 32.

4. Adobe. Творчість і дизайн [Електронний ресурс]: Що таке VR? Поняття віртуальної реальності. – Режим доступу: <https://www.adobe.com/ua/products/substance3d/discover/what-is-vr.html>

5. LinkedIn Pulse [Електронний ресурс]: What is the Difference between Augmented Reality (AR) vs. Virtual Reality (VR) vs. Mixed Reality (MR)? by Shah Reyad Farabi, 2022. – Режим доступу: <https://www.linkedin.com/pulse/what-difference-between-augmented-reality-ar-vs-vr-mr-reyad-farabi/>

6. Adobe. Творчість і дизайн [Електронний ресурс]: Що таке доповнена реальність? – Режим доступу: <https://www.adobe.com/ua/products/substance3d/discover/what-is-augmented-reality.html>

7. Adobe. Творчість і дизайн [Електронний ресурс]: Що таке змішана реальність? – Режим доступу: <https://www.adobe.com/ua/products/substance3d/discover/mixed-reality.html>

8. IT Enterprise [Електронний ресурс]: Internet of Things, IoT. – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot>

9. IT Enterprise [Електронний ресурс]: Augmented Reality, AR. Industry 5.0 – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/dopolnennaja-realnost-ar>

10. Sigma Software University [Електронний ресурс]: Що таке база даних (БД)? – Режим доступу: <https://university.sigma.software/what-is-database/>

11. Ukrainian Digital Community [Електронний ресурс]: ШІ та революція в digital: як штучний інтелект змінює гру. – Режим доступу: <https://ukrainiandigital.com/strong-yak-stvoryty-vlasne-onlayn-navchannia-strong/>

12. Research Gate [Електронний ресурс]: A behavior tree example for the patrol application. – Режим доступу: [https://www.researchgate.net/figure/A-behavior-tree-example-for-the-patrol-application\\_fig3\\_287209460](https://www.researchgate.net/figure/A-behavior-tree-example-for-the-patrol-application_fig3_287209460)

13. Unreal Engine Forums [Електронний ресурс]: [UE5] Unreal Engine Support for Machine Learning. – Режим доступу: <https://forums.unrealengine.com/t/ue5-unreal-engine-support-for-machine-learning/514171>

14. Криворізький державний педагогічний університет [Електронний ресурс]: Віртуальна хімічна лабораторія. – Режим доступу: <https://kdpu.edu.ua/khimii-ta-metodyky-ii-navchannia/tsikava-khimiiia/dlia-vseznaiook/5928-virtualna-khimichna-laboratoriia.html>

15. PHET Interactive Simulations [Електронний ресурс]: Лабораторія конденсаторів. – Режим доступу: <https://phet.colorado.edu/uk/simulations/capacitor-lab>

16. НУШ. Нова українська школа. Смарт освіта [Електронний ресурс]: Віртуальна наукова лабораторія, 2023: корисні ресурси для учнів і педагогів. –

Режим доступа: <https://nus.org.ua/news/virtualna-naukova-laboratoriya-korysni-resursy-dlya-uchniv-i-pedagogiv/>

17. DAIC [Электронный ресурс] Osso VR Expands Cardiology Training through ACC Collaboration. – Режим доступа: <https://www.dicardiology.com/content/osso-vr-expands-cardiology-training-through-acc-collaboration>

18. Unreal Engine [Электронный ресурс]: Unreal Engine 5. – Режим доступа: <https://www.unrealengine.com/en-US/unreal-engine-5>

19. Dev Epic Games Documentation [Электронный ресурс]: Artificial Intelligence. – Режим доступа: <https://dev.epicgames.com/documentation/en-us/unreal-engine/artificial-intelligence-in-unreal-engine>

20. Fab [Электронный ресурс]: HiSQLite3(SQLite3 Blueprint For UE5). – Режим доступа: <https://www.fab.com/listings/0c41df45-60b0-4307-a1a0-6cfe1fc33035>

21. Visual Studio Microsoft [Электронный ресурс]: Visual Studio. – Режим доступа: <https://visualstudio.microsoft.com/>

22. Dev Epic Games Documentation [Электронный ресурс]: Blueprints Visual Scripting. – Режим доступа: <https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprints-visual-scripting-in-unreal-engine>

23. Dev Epic Games Documentation [Электронный ресурс]: Setting Up Visual Studio. – Режим доступа: <https://dev.epicgames.com/documentation/en-us/unreal-engine/setting-up-visual-studio-development-environment-for-cplusplus-projects-in-unreal-engine>

24. Visual Studio Code [Электронный ресурс]: Your code editor. Redefined with AI. – Режим доступа: <https://code.visualstudio.com/>

25. JetBrains [Электронный ресурс]: Rider. – Режим доступа: <https://www.jetbrains.com/en-en/rider/>

26. Blender [Электронный ресурс]: Blender 4.3. A stroke of genius. – Режим доступа: <https://www.blender.org/>

27. Autodesk [Електронний ресурс]: Autodesk Maya: Create expansive worlds, complex characters, and dazzling effects. – Режим доступу: [https://www.autodesk.com/products/maya/overview?term=1-](https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription)

[YEAR&tab=subscription](https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription)

28. SQLite [Електронний ресурс]: What Is SQLite?. – Режим доступу: <https://www.sqlite.org/>

29. DOU Форум [Електронний ресурс]: Розбираємось з реляційними та нереляційними базами даних. – Режим доступу: <https://dou.ua/forums/topic/49585/>

30. SQLite Browser [Електронний ресурс]: DB Browser for SQLite. – Режим доступу: <https://sqlitebrowser.org/>

31. GitHub [Електронний ресурс]: Introduction to GitHub. – Режим доступу: <https://github.com/skills/introduction-to-github>

32. Git LFS [Електронний ресурс]: An open source Git extension for versioning large files. – Режим доступу: <https://git-lfs.com/>

33. Git SCM [Електронний ресурс]: SCM tools. – Режим доступу: <https://git-scm.com/>

34. Blender Extensions [Електронний ресурс]: MeasureIt. – Режим доступу: <https://extensions.blender.org/add-ons/measureit/>

35. CORE AC [Електронний ресурс]: Розробка підходу до перевірки адекватності моделі прийняття рішень з нечіткими параметрами. О. А. Шаригін. – Режим доступу: <https://core.ac.uk/download/pdf/52159529.pdf>

36. Dev Epic Games Documentation [Електронний ресурс]: Working with Data in Unreal Engine. Data tables. – Режим доступу: <https://dev.epicgames.com/community/learning/tutorials/Gp9j/working-with-data-in-unreal-engine-data-tables-data-assets-uproperty-specifiers-and-more>

37. Coding Babble [Електронний ресурс]: Using SQLite with Unreal. – Режим доступу: <https://codingbabble.com/posts/using-sqlite-with-unreal/>

38. GitHub [Електронний ресурс]: KoboldCPP repository. – Режим доступу: <https://github.com/LostRuins/koboldcpp>

39. GitHub [Електронний ресурс]: Llama 2 AI model. – Режим доступу:  
<https://github.com/meta-llama/llama>

40. Методичні вказівки «Основи охорони праці»/ Упоряд.: Т.Є. Стиценко,  
В.А. Айвазов, О.В. Мамонтов. Харків: ХНУРЕ, 2018. 120 с.