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 2. Моделювання лабораторних робіт

Виконав: здобувач 2 року навчання, гр. КІТПВм-23-1
Тарасенко К.А.
(прізвище, ініціали)

Спеціальність
174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
освітньої програми Комп'ютерно-інтегровані технологічні процеси і виробництва
(код і повна назва напряму)

Тип програми освітньо-професійна
(повна назва освітньої програми)

Керівник проф. Цимбал О.М.
(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри _____ Невлюдов І.Ш.
(підпис) (прізвище, ініц

2025 р.

Харківський національний університет радіоелектроніки

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

(код і повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Тарасенку Кірілу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи КТ 1. Розроблення комп'ютерної моделі лабораторій кафедри КІТАР на основі методів віртуальної реальності ПТ 2. Моделювання лабораторних робіт

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:
Технології віртуальної реальності; технології, що підтримують віртуальні лабораторії в Unreal Engine; штучний інтелект робота в умовах Unreal Engine 5; реалізація та методи інтеграції OpenCV; аналіз структури проєкту на базі Unreal Engine 5.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 20 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	8.12.2024	виконано
2	Написання другого розділу	15.12.2024	виконано
3	Перший етап розробки	20.12.2024	виконано
4	Написання третього розділу	26.12.2024	виконано
5	Другий етап розробки	03.01.2025	виконано
6	Завершення розробки	05.01.2025	виконано
7	Оформлення пояснювальної записки	09.01.2025	виконано
8	Підготовка та оформлення презентації	15.01.2025	виконано

Дата видачі завдання 22.11.2024 р.

Здобувач _____ Тарасенко К.А.
(підпис) (прізвище, ініціали)

Керівник роботи _____ проф. Цимбал О.М.
(підпис) (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. Тарасенко К.А.

РЕФЕРАТ

Пояснювальна записка: 87 с., 36 рис., 2 дод., 31 джерела.

ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, OPEN CV, ШТУЧНИЙ ІНТЕЛЕКТ, UNREAL ENGINE 5

Мета роботи – покращення методів створення інтерактивного середовища, для вдосконалення та створення лабораторних робіт, з підтримкою 3D-моделювання, штучного інтелекту та аналізу даних.

Об’єкт дослідження – процеси візуалізації, анімування та програмування.

Предмет дослідження – метод створення 3D-візуалізації лабораторної роботи у межах кафедри КІТАР для збору та аналізу даних віртуального помічника.

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

Крім того, результати цього дослідження 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: 87 p., 36 fig., 2 appendix, 31 sources.

VIRTUAL ENVIRONMENT, OPEN CV, ARTIFICIAL INTELLIGENCE, UNREAL ENGINE 5

The purpose of the work is to improve the methods of creating an interactive environment for improving and creating laboratory work, with the support of 3D modeling, artificial intelligence and data analysis.

The object of the research is the processes of visualization, animation and programming.

The subject of the research is a method of creating a 3D visualization of laboratory work within the KITAR department for collecting and analyzing data for a virtual assistant.

This work will be useful for teachers of educational institutions, educational platforms and companies. 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, the results of this study demonstrate how the integration of virtual reality and artificial intelligence technologies can optimize the learning process, ensure 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 fields of education."

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних;

БЖД – безпека життєдіяльності;

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

ПЗ – програмне забезпечення;

ТЗ – технічне завдання;

ШІ – штучний інтелект;

2D – 2-dimensional;

3D – 3-dimensional;

4K – Ultra High Definition;

AR – Augmented Reality;

AI – Artificial Intelligence;

DB – Data Base;

FPS – Frames Per Second;

FullHD – High Definition;

Git – Distributed Version Control System;

LFS – Large File Storage;

QR – Quick Response Code;

RAM – Random Access Memory;

TXT – Text File;

UE – Unreal Engine;

VR – Virtual Reality;

VRAM –Video Random Access Memory.

ВСТУП

Сучасні технології відіграють ключову роль у розвитку освітніх методик, спрямованих на підвищення доступності, ефективності та інтерактивності навчального процесу. Одним із перспективних напрямів є впровадження віртуальних лабораторій, які дозволяють відтворювати умови реальних експериментів у комп'ютерному середовищі. Завдяки використанню технологій віртуальної та доповненої реальності, такі лабораторії стають потужним інструментом для засвоєння складних концепцій у STEM-дисциплінах (наука, технології, інженерія та математика). Це особливо важливо в умовах зростання потреб у дистанційній освіті та адаптації до нових викликів, таких як пандемії, військові конфлікти чи недостатнє матеріально-технічне забезпечення навчальних закладів.

Віртуальна лабораторія – це інтерактивне середовище, яке симулює реальні лабораторні процеси, дозволяючи студентам експериментувати з віртуальним обладнанням та досліджувати наукові явища без ризику для здоров'я та витрат на дорогі матеріали. Цей інструмент стає особливо актуальним у галузях, які вимагають дороговартісного обладнання або складних умов для проведення дослідів, таких як хімія, фізика, біологія, інженерія та медицина. Крім того, віртуальні лабораторії сприяють адаптації навчального процесу до індивідуальних потреб студентів, дозволяючи створювати навчальні сценарії, що відповідають рівню підготовки кожного користувача.

Розробка віртуальних лабораторій передбачає інтеграцію новітніх 3D-технологій, ігрових рушіїв та інструментів програмування. Unreal Engine 5 – один із найбільш передових рушіїв, що використовується для створення реалістичних візуалізацій і забезпечення високого рівня інтерактивності. Технології Nanite та Lumen, які підтримуються цим рушієм, забезпечують високу деталізацію моделей і динамічне освітлення, що значно покращує користувацький досвід. Крім того, впровадження штучного інтелекту та

комп’ютерного зору, зокрема через використання бібліотек OpenCV і YOLOv5, відкриває можливості для створення адаптивного навчального контенту, який враховує індивідуальні потреби студентів.

На міжнародному рівні технології віртуальних лабораторій уже широко використовуються. Зокрема, платформи на зразок Labster забезпечують широкий доступ до інтерактивних симуляцій у галузях науки та інженерії. У країнах Європи, Америки та Азії впровадження таких платформ сприяє модернізації навчальних програм, підвищуючи залученість студентів та ефективність засвоєння матеріалу. У той же час, адаптація цих технологій до локальних освітніх потреб вимагає врахування культурних, технічних та інфраструктурних особливостей.

Україна, як частина глобальної освітньої спільноти, активно впроваджує цифрові технології в навчальний процес. Віртуальні лабораторії набувають популярності завдяки здатності компенсувати обмеження реальних лабораторій, особливо в закладах, що не мають достатнього фінансування для придбання дорогого обладнання. Досвід впровадження таких платформ, як ROQED, у школах та університетах України свідчить про значний інтерес до інтерактивних рішень, які дозволяють студентам не лише отримувати теоретичні знання, але й розвивати практичні навички.

У рамках даного проєкту буде досліджено методи створення віртуальної лабораторії з використанням передових програмних середовищ, таких як Unreal Engine 5, Blender, Maya та Substance Painter.

Мета роботи – покращення методів створення інтерактивного середовища, для вдосконалення та створення лабораторних робіт, з підтримкою 3D-моделювання, штучного інтелекту та аналізу даних.

Об’єкт дослідження – процеси візуалізації, анімування та програмування.

Предмет дослідження – метод створення 3D-візуалізації лабораторної роботи у межах кафедри КІТАР для збору та аналізу даних віртуального помічника.

Необхідно виконати наступні завдання, щоб досягти поставленої мети:

– необхідно проаналізувати літературу та дослідження, що відносяться до віртуальної візуалізації, а саме: створення віртуальної лабораторії та анімування;

– дослідити приклади успішної розробки та збору даних з віртуальних лабораторій;

– вивчити можливості застосування отриманих даних у навчанні, з'ясувавши переваги та недоліки, а також розглянути можливі варіанти використання та покращення функціональності, як в же створених методів, та і тих, які тільки впроваджуються;

– створити алгоритм для реалізації майбутнього проекту.

1 АНАЛІЗ МЕТОДІВ СТВОРЕННЯ ВІРТУАЛЬНОЇ ЛАБОРАТОРІЇ

1.1 Аналіз базової інформації

Віртуальна лабораторія – це комп’ютерне середовище, що імітує роботу реальної лабораторії з використанням технологій віртуальної та доповненої реальності. Це інтерактивне середовище дозволяє студентам відтворювати реальні експерименти, взаємодіяти з віртуальним обладнанням та досліджувати наукові процеси.

Створення віртуальних лабораторій є важливим аспектом сучасної освіти, оскільки дозволяє забезпечити студентам доступ до складного обладнання та практичних занять без фізичної присутності в лабораторіях.

Сьогодні існує кілька методів створення віртуальних лабораторій, і для цього можна використовувати різноманітні платформи, такі як Labster (рис. 1.1), а також потужні графічні двигуни, такі як Unreal Engine 5.

Рисунок 1.1 – Віртуальна лабораторія у середовищі Labster

Перший метод передбачає розробку віртуальних лабораторій на основі стандартних платформ для навчання. Одним із найвідоміших прикладів є платформа Labster, яка спеціалізується на розробці інтерактивних наукових лабораторій, що охоплюють різні галузі науки: біологію, хімію, фізику тощо. Labster використовує технології віртуальної реальності (VR) і доповненої реальності (AR), щоб створювати інтерактивні симуляції наукових експериментів, які дозволяють студентам працювати з віртуальним обладнанням, досліджувати хімічні реакції, проводити біологічні експерименти й багато іншого. Основною перевагою Labster є те, що він надає вже готову платформу з широким набором експериментів, яка є легкою у використанні для освітніх закладів. Недоліком же є обмежена можливість персоналізації та адаптації до специфічних навчальних програм і потреб конкретного закладу.

Наступний метод – це розробка кастомних віртуальних лабораторій (рис. 1.2) за допомогою ігрових і графічних двигунів, таких як Unreal Engine 5 [4]. Цей підхід забезпечує набагато ширші можливості для персоналізації, дозволяє створювати лабораторії з будь-яким типом обладнання, інтерфейсу й навігації, а також дозволяє використовувати найсучасніші технології графіки, фізики та анімації. Unreal Engine 5, зокрема, є одним із найсучасніших двигунів, який має інструменти для створення високоякісних віртуальних просторів і забезпечує реалістичне відтворення фізичних процесів. У поєднанні з VR-технологіями Unreal Engine 5 дозволяє створювати надзвичайно реалістичні і занурюючі віртуальні середовища для навчання, де студенти можуть взаємодіяти з обладнанням, вивчати реакції й експериментувати у реальному часі.

VR забезпечує просторове сприйняття, що робить навчання більш інтуїтивним і цікавим. Наприклад, студенти можуть не тільки бачити експеримент, але й взаємодіяти з ним за допомогою контролерів або трекерів руху, виконувати складні маніпуляції з точністю, яку важко досягти у звичайних умовах. Це також сприяє розвитку навичок, необхідних для роботи в реальних лабораторіях, і покращує розуміння складних теоретичних концепцій за рахунок їх візуалізації.

Рисунок 1.2 – Віртуальна лабораторія у середовищі Unreal Engine 5

Unreal Engine 5 є потужним графічним середовищем, яке відрізняється реалістичною обробкою графіки та надає розробникам безліч інструментів для створення динамічних віртуальних середовищ. Двигун має особливий інструмент під назвою Nanite (рис. 1.3), який дозволяє обробляти мільярди полігонів без втрат продуктивності, що є важливим для створення складних і деталізованих моделей обладнання та інтер'єрів лабораторій. Це дозволяє не тільки створювати візуально привабливі моделі, але й забезпечувати користувачів реалістичним досвідом занурення.

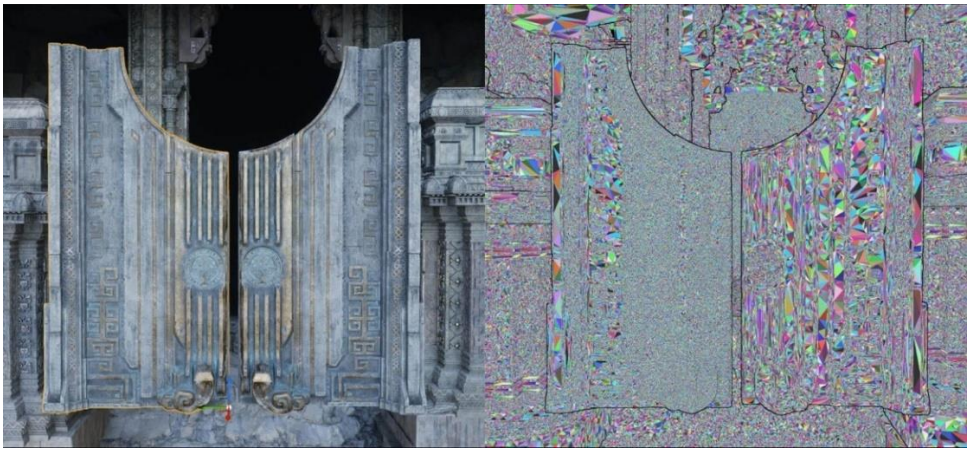

Рисунок 1.3 – Приклад застосування технології Nanite в Unreal Engine

Крім того, Unreal Engine 5 підтримує технологію Lumen (рис. 1.4) для створення динамічного освітлення, що дозволяє підкреслити освітленість приміщення та додати реалістичності до лабораторних середовищ, що важливо для експериментів, які потребують точного світлового середовища.

Рисунок 1.4 – Приклад застосування технології Lumen в Unreal Engine

Unreal Engine 5 також підтримує інтеграцію з віртуальними та доповненими реальностями, що робить його ідеальним вибором для створення віртуальних лабораторій з глибоким зануренням.

Студенти можуть користуватися VR-шоломами, щоб взаємодіяти з віртуальними лабораторіями так, як би вони перебували в реальному середовищі.

Також Unreal Engine 5 пропонує підтримку мультиплеєрних функцій, що дозволяє кільком студентам одночасно працювати у віртуальній лабораторії, спільно проводячи експерименти та обговорюючи результати в режимі реального часу. Це може бути особливо корисним для дистанційного навчання, коли студенти не можуть фізично бути в одній лабораторії.

1.2 Технології, що підтримують віртуальні лабораторії в Unreal Engine

Щоб створити віртуальну лабораторію за допомогою Unreal Engine 5, використовують кілька основних інструментів та методів.

Створення 3D-моделей лабораторного обладнання та інтер'єру. Для цього можуть використовуватися спеціалізовані 3D-графічні програми, такі як Blender або Maya, після чого моделі імпортуються в Unreal Engine 5. Завдяки використанню Nanite, моделі можуть бути надзвичайно деталізованими, що підвищує якість занурення студентів у віртуальне середовище.

Створення 3D-анімацій та рігу для лабораторного обладнання. Для цього використовуються спеціалізовані програми, такі як Maya, де на моделі обладнання накладається ріг (скелетна структура), що забезпечує можливість руху та анімації об’єкта. Після створення рігу анімації для обладнання, такі як обертання або переміщення частин, додаються для імітації реальної роботи інструментів у лабораторії. Готові анімації імпортуються в Unreal Engine 5, де вони стають інтерактивними, що дозволяє студентам взаємодіяти з обладнанням у реальному часі, підвищуючи рівень занурення в навчальне середовище.

Налаштування Chaos Physics для відтворення реалістичних фізичних взаємодій. Chaos Physics є частиною Unreal Engine 5, вона дозволяє розробникам налаштовувати фізику об'єктів, забезпечуючи реалістичні рухи, взаємодії та зіткнення віртуального обладнання. Це особливо важливо для лабораторій, які потребують точного моделювання фізичних процесів, таких як механічні експерименти чи маніпуляції з рідинами. Така фізика додає правдоподібності до дій користувачів та дозволяє їм відчути наслідки своїх експериментів майже так само, як у реальних умовах.

Програмування інтерактивності для налаштування симуляції. Unreal Engine 5 надає потужний інструмент Blueprints, який дозволяє створювати складні інтерактивні сценарії без необхідності програмування на мові C++. Blueprints є візуальним редактором, де можна налаштувати реакції об'єктів на дії користувачів, реалізувати різні експериментальні сценарії, налаштувати

інтерфейс лабораторії та багато іншого. Такий підхід дозволяє створювати віртуальні лабораторії з високим рівнем інтерактивності, де студенти можуть виконувати різноманітні дії з об'єктами, розуміти результати експериментів, отримувати зворотний зв'язок і навіть взаємодіяти з віртуальними викладачами.

YOLOv5 (“You Only Look Once”) – це найновіша версія знаменитої сімейства моделей для детекції об’єктів, що широко застосовуються у різних галузях, серед яких розробка віртуальних систем, розпізнання об’єктів у відеопотоках та створення навчальних моделей.

YOLOv5 відрізняється своєю простотою інтеграції, швидкістю обробки даних та можливістю налаштування під різні завдання. Ця модель була розроблена з урахуванням застосування у реальному часі, що робить її незамінною у системах, що потребують швидкої та точної обробки даних.

Головні переваги YOLOv5 порівняно з іншими сучасними моделями детекції об’єктів включають:

Висока швидкість роботи: YOLOv5 швидше обробляє дані на простішому обладнанні, ніж більші або складні моделі, такі як YOLOv4 або EfficientDet.

Адаптивність та можливість налаштування: подана модель підходить для широкого спектру завдань без значного складнощення перенавчання.

Низькі вимоги до обчислюваної сили: YOLOv5 можна запускати на стандартних графічних процесорах, не використовуючи дороговартісне обладнання.

Застосування YOLOv5 в Unreal Engine відкриває потенціал реалізації різних типів віртуальних лабораторій, навчальних систем та ігор з штучним інтелектом. Головні потенційні переваги цієї інтеграції полягають у наступному:

Реалістичний рендеринг та імітація: інтеграція з Unreal Engine дозволяє поєднувати точне розпізнання з реалістичним отображенням сцен, створюючи високоякісні візуальні об’єкти.

Навчальні системи та симуляції: YOLOv5 у Unreal Engine можна використовувати для створення віртуальних лабораторій з високою точністю

сприйняття та реалістичним відображенням об’єктів. Це сприяє покращенню навчального процесу завдяки використанню інтерактивних імітацій.

У розробці ігор YOLOv5 може застосовуватись для створення складних AI-систем, які адаптивно реагують на дії гравця. Це відкриває нові можливості для інтерактивного дизайну та сюжетних рішень.

Проте, попри всі переваги, використання YOLOv5 має певні обмеження та виклики:

Хоча YOLOv5 підтримує простоту налаштування, інтеграція з ігровим движком, таким як Unreal Engine, може вимагати спеціалізованих знань у галузях AI та 3D-моделювання.

Вимоги до навчальних даних: для досягнення високої точності потрібні якісні та різноманітні набори даних, що може бути складним завданням у деяких випадках.

Ресурсоємність у складних сценах: попри низькі вимоги до обладнання, обробка великих чи складних сцен може створювати значне навантаження на систему.

Потенціал технології YOLOv5 (рис. 1.5) у поєднанні з Unreal Engine виглядає перспективним. Зростаюча популярність інтерактивних навчальних середовищ та ігор з елементами штучного інтелекту відкриває безліч напрямів для використання цієї технології. Вона може стати основою для розвитку адаптивних освітніх платформ, реалістичних симуляторів для тренувань у різних галузях, таких як медицина чи авіація, а також покращення користувацького досвіду у сфері віртуальної реальності.

Додатково, OpenCV (Open Source Computer Vision Library) також відіграє важливу роль у розробці інтерактивних систем на основі Unreal Engine. OpenCV надає широкий набір інструментів для обробки зображень, відеопотоків та машинного навчання, що робить її чудовим доповненням до YOLOv5. У рамках Unreal Engine, OpenCV може використовуватись для попередньої обробки відео чи зображень перед їхнім подальшим аналізом або відображенням. Це включає

такі можливості, як корекція зображень, детекція ключових точок, сегментація об’єктів та стереозір для аналізу глибини сцени.

OpenCV дозволяє створювати системи розширеної реальності, де дані з камер реального часу обробляються й інтегруються у віртуальне середовище.

Використання OpenCV для очищення та підготовки даних перед їхньою обробкою YOLOv5 підвищує загальну продуктивність системи.

Бібліотека дозволяє детектувати та аналізувати рух об’єктів у відео, що корисно для створення динамічних та інтерактивних ігор чи симуляцій.

Загалом, OpenCV та YOLOv5 у поєднанні з Unreal Engine утворюють потужний інструментальний набір для розробки інноваційних, інтерактивних і реалістичних віртуальних середовищ. Успішна інтеграція цих технологій відкриває широкі перспективи у створенні освітніх платформ, розважальних продуктів та симуляторів нового покоління.

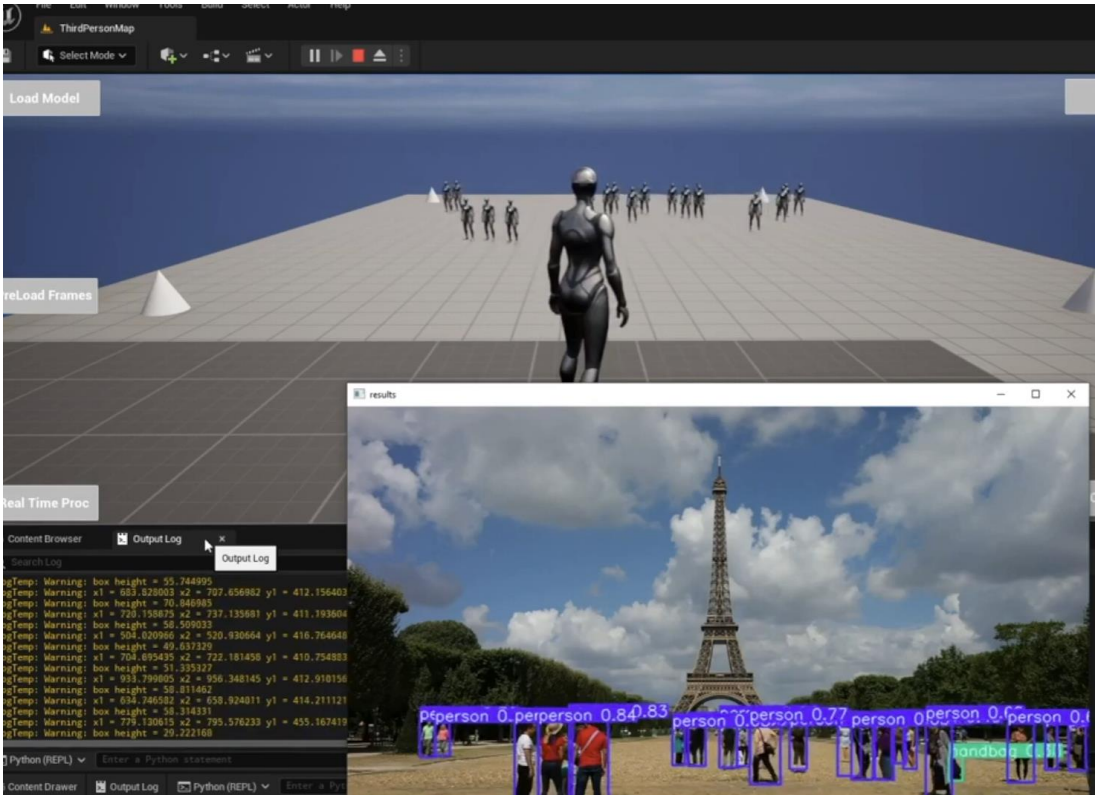

Рисунок 1.5 – Приклад застосування технології YOLOv5 в Unreal Engine 5

1.3 Вибір додаткових програмних середовищ

Для досягнення високої якості та інтерактивності моделей віртуальної лабораторії необхідно обрати відповідні програми для кожного з етапів, включаючи 3D-моделювання, анімацію, текстурування, рендеринг, а також інструменти для програмування. Одними з найбільш популярних інструментів для 3D-моделювання та анімації є Blender та Maya, обидва з яких активно використовуються в індустрії, але мають свої особливості та обмеження.

Blender і Maya – це дві провідні програми для 3D-моделювання, анімації та рендерингу, але вони мають різні сильні сторони, що залежить від специфічних потреб проекту. Blender є безкоштовним і відкритим програмним забезпеченням, яке надає широкий набір інструментів для моделювання, текстурування, рендерингу, анімації та створення спецефектів. Це робить його ідеальним вибором для тих, хто шукає економічно вигідне рішення, а також для індивідуальних розробників або малих студій, що не мають великих бюджетів.

Maya, з іншого боку, є платним професійним програмним забезпеченням, яке часто використовується великими студіями для створення високоякісних анімацій, фільмів і відеоігор. Однією з основних переваг Maya є її потужні інструменти для анімації, особливо в плані рігінгу та створення складних анімаційних процесів. Вона також забезпечує більшу сумісність з іншими програмами і платформами в індустрії, що робить її стандартом для багатьох великих проектів.

Однією з основних проблем при роботі з Blender і Maya є несумісність рігу між цими програмами. Ріг – це система кісток та контролерів, яка дозволяє анімувати 3D-моделі, забезпечуючи їх рух. Враховуючи, що Blender та Maya використовують різні алгоритми та підходи до створення рігу, переносити анімації між цими програмами може бути складно. Наприклад, ріг, створений у Maya, не буде автоматично працювати в Blender без значних змін і налаштувань. Це пов’язано з різними форматами даних, які обробляються в кожній програмі, а також з особливостями функціонування контролерів та анімаційних систем.

Що стосується форматів файлів для обміну даними між різними програмами, то найпоширенішими є формати FBX, OBJ, COLLADA (DAE) і Alembic (ABC). Формат FBX є одним з найбільш універсальних і підтримується як Blender, так і Maya. Він дозволяє зберігати 3D-геометрію, та текстури, що робить його підходящим для передачі моделей між програмами

Для рендерингу і текстурування є кілька популярних інструментів, які забезпечують високу якість візуалізації. Одним з найбільш відомих є Substance Painter, програма, яка дозволяє створювати реалістичні текстури з великим рівнем деталізації. Вона підтримує інтеграцію з більшістю 3D-редакторів, включаючи Blender і Maya, що дозволяє створювати текстури безпосередньо на 3D-моделях і тестувати їх у реальному часі.

Marmoset Toolbag – це програма для рендерингу, яка дозволяє створювати високоякісні візуалізації моделей у реальному часі. Вона добре інтегрується з іншими програмами для 3D-моделювання і є чудовим інструментом для перевірки текстур і матеріалів. Вона також підтримує рендеринг у реальному часі, що є корисним при роботі з інтерактивними елементами віртуальної лабораторії.

При програмуванні та інтеграції моделі в Unreal Engine 5, часто використовуються плагіни та додаткові інструменти для полегшення розробки. Для роботи з анімаціями та інтерактивністю в Unreal Engine 5 можуть використовуватися плагіни, які дозволяють автоматизувати багато процесів, таких як імпорт 3D-моделей, налаштування анімацій або адаптація контенту під вимоги навчальної програми.

При програмуванні в Visual Studio необхідно враховувати специфіку роботи з Unreal Engine, а саме використання C++ для розробки інтерактивних елементів [5]. Unreal Engine активно використовує об'єктно-орієнтований підхід, і знання основ C++ є критично важливим для ефективної роботи. Зокрема, розробники повинні розуміти поняття класів, спадкування, поліморфізму та управління пам'яттю, оскільки ці концепції лежать в основі роботи з рушієм.

Unreal Engine має свій унікальний API (Application Programming Interface), що включає велику кількість готових класів, методів і функцій, спеціально розроблених для роботи з 3D-середовищами. Наприклад, класи AActor або UObject є фундаментальними для створення і взаємодії з об'єктами у віртуальному світі. Специфічні класи, такі як UStaticMeshComponent для роботи з 3D-моделями або USkeletalMeshComponent для управління скелетною анімацією, дозволяють розробникам створювати складні сцени та анімаційні ефекти.

Також Unreal Engine пропонує потужні засоби для інтеграції анімацій, включаючи Animation Blueprints, що дозволяють поєднувати логіку на основі C++ з візуальним сценарієм. Крім цього, для створення інтерфейсів користувача використовується система UMG (Unreal Motion Graphics), яка дозволяє програмувати інтерфейси за допомогою C++ або Blueprint.

Ще однією важливою особливістю є система делегатів і подій, яка дозволяє ефективно організовувати взаємодію між компонентами. Наприклад, можна створити події для обробки дій користувача, таких як натискання кнопок, або для відстеження станів у грі.

У процесі розробки віртуальної лабораторії важливим елементом є управління версіями проекту та співпраця між розробниками. Для цього одним із найпопулярніших інструментів є GitHub, який забезпечує зручну платформу для зберігання коду, трекінгу змін, а також організації колективної роботи над проектом. GitHub дозволяє розробникам працювати одночасно над різними частинами проекту, зберігаючи історію змін і надаючи можливість повернення до попередніх версій, що є критично важливим при роботі з великими і складними 3D-моделями та інтерактивними елементами, такими як риги і анімації, створювані у Blender або Maya.

Крім того, GitHub надає потужні інструменти для ведення документації, організації завдань, проведення обговорень і управління проектами через функції, такі як Issues і Pull Requests. Це дозволяє команді ефективно взаємодіяти, обмінюватися ідеями та забезпечувати високий рівень контролю за

процесом розробки. Для проектів, які включають інтеграцію з Unreal Engine 5, GitHub може бути корисним для зберігання не лише коду, а й конфігураційних файлів, скриптів для програмування в Visual Studio, а також файлів проектів, створених у 3D-редакторах.

Окрім GitHub, існують і інші аналоги, які можуть бути використані для спільної розробки. Наприклад, GitLab і Bitbucket також надають схожі функціональні можливості з управління версіями, а також інтеграцію CI/CD (неперервна інтеграція та безперервне постачання). GitLab, крім того, має розширені можливості для автоматизації тестування та розгортання проектів, що може бути корисним для віртуальних лабораторій, де необхідно регулярно перевіряти функціональність інтерактивних елементів та анімацій.

Зокрема, у проектах, пов’язаних із 3D-моделюванням і анімацією, GitHub може бути корисним для зберігання шаблонів моделей, ригів, текстур та сценаріїв, що можуть бути використані в Unreal Engine 5. Також це дозволяє команді дотримуватися однакового стилю кодування та стандартів якості, що є важливим при роботі над великими проектами, де залучено багато розробників.

Платформи, такі як GitHub, дозволяють не лише зберігати код, а й обмінюватися навчальними матеріалами, такими як документація, туторіали та приклади використання певних функцій або методів. Це створює середовище, де команди можуть швидко адаптуватися до нових викликів та впроваджувати інноваційні рішення.

1.4 Висновок до першого розділу

Під час виконання першого розділу було здійснено детальний аналіз методів створення віртуальних лабораторій, їхнього значення в сучасній освіті та перспектив інтеграції технологій штучного інтелекту. Було розглянуто наявні програмні платформи, зокрема Unreal Engine 5, які використовуються для створення інтерактивних навчальних середовищ із високим рівнем реалістичності та адаптивності. Визначено основні підходи до розробки таких

лабораторій, включаючи використання 3D-моделювання, анімації та алгоритмів обробки зображень, що дозволяють створювати складні інтерактивні сценарії навчання.

Особливу увагу приділено аналізу графічних технологій Unreal Engine 5, таких як Nanite для високополігонального рендерингу та Lumen для глобального освітлення, що суттєво покращують візуальне сприйняття віртуального середовища. Також розглянуто можливості інтеграції OpenCV та YOLOv5 для реалізації функцій розпізнавання об’єктів і комп’ютерного зору в межах навчального процесу. Було сформовано вимоги до програмного забезпечення, визначено оптимальні інструменти для створення моделей і текстур (Blender, Maya, Substance Painter) та засоби керування версіями (GitHub).

Розроблено концепцію інтерактивної лабораторії, яка враховує гнучкість навчальних сценаріїв, можливість автоматизованого аналізу студентських робіт та підлаштування контенту під індивідуальні потреби користувачів. Також створено алгоритм реалізації, що включає поетапний процес розробки: від первинного аналізу й прототипування до інтеграції навчальних матеріалів та тестування.

Встановлено, що поєднання технологій комп’ютерного зору, баз даних та віртуальної реальності сприяє вдосконаленню методів дистанційного навчання, підвищенню ефективності засвоєння матеріалу студентами та покращенню інтерактивності освітнього процесу. Використання адаптивних технологій дозволяє розширити доступ до навчальних ресурсів, зробивши освітній процес більш ефективним, динамічним і технологічно просунутим.

2 МОДЕЛЮВАННЯ СТРУКТУРИ ТА ПЛАНУ РЕАЛІЗАЦІЇ ПРОЄКТУ

2.1 Аналіз літератури та досліджень

Перед початком розробки віртуальної лабораторії був проведений аналіз методів створення подібних середовищ, який охоплював доступні 3D-технології, ігрові рушії та інтеграцію штучного інтелекту для забезпечення інтерактивності. Дослідження показало, що використання високоякісного 3D-моделювання в програмах на зразок Blender і Maya разом із рушієм Unreal Engine 5 дозволяє досягти значного рівня деталізації, що є ключовим для створення реалістичних і функціональних лабораторій. Крім того, впровадження технологій Nanite та Lumen в Unreal Engine 5 забезпечує високу продуктивність та реалістичне освітлення, що значно покращує занурення студентів у навчальний процес.

Також було оцінено методи створення рігу та анімації для лабораторного обладнання, що дозволяє відтворювати реальні дії обладнання й забезпечує його взаємодію з користувачами в реальному часі. Використання інструментів для створення рігу та анімації, як-от Blender або Maya, надає можливість додавати точні рухи для маніпуляцій обладнання, що робить навчання більш інтуїтивним і практично орієнтованим.

Окрім цього, аналіз показав, що завдяки можливостям ігрових рушіїв і 3D-технологій можна створювати повністю інтерактивні симуляції, які відображають складні наукові процеси. Інтеграція штучного інтелекту у віртуальні лабораторії відкриває можливості для адаптації складності навчального контенту залежно від індивідуальних потреб студента, що підвищує ефективність засвоєння матеріалу.

Досвід Європи Америки та Азії також підтверджує успішність подібних рішень: ШІ і 3D лабораторій використовуються для автоматизації навчання та створення симуляцій, що дозволяє значно покращити якість освіти.

Японські університети зараз створюють інклюзивне навчальне середовище (рис. 2.1), і студентам з обмеженими можливостями (SwD) [6] доступна різноманітна підтримка. Однак кількість SwD у науці, технологіях, інженерії та математиці (STEM) мінімальна через нестачу доступності.

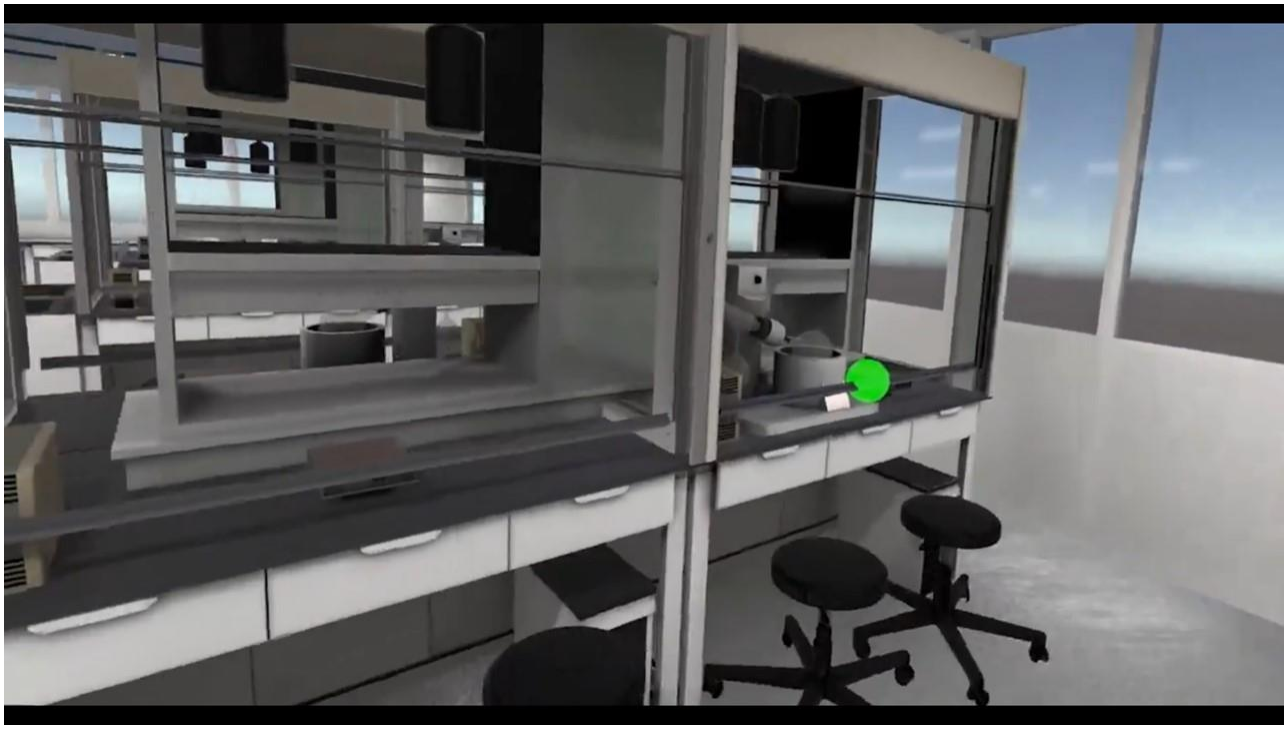

Рисунок 2.1 – Приклад зарубіжної віртуальної лабораторії від Namiki Laboratory

Попит на віртуальні лабораторії в Україні останніми роками демонструє помітне зростання, що зумовлено кількома ключовими факторами. Одним із головних є зростання інтересу до інтерактивних і практично орієнтованих навчальних рішень, особливо в освітніх закладах, де потреба в сучасних інструментах для навчання стає все більш актуальною. Віртуальні лабораторії (рис. 2.2), такі як ROQED [7], які активно впроваджуються в українські школи та університети, забезпечують студентів можливістю безпечно відпрацьовувати практичні навички та експериментувати в умовах, наближених до реальних, але без фізичних ризиків або витрат на матеріали.

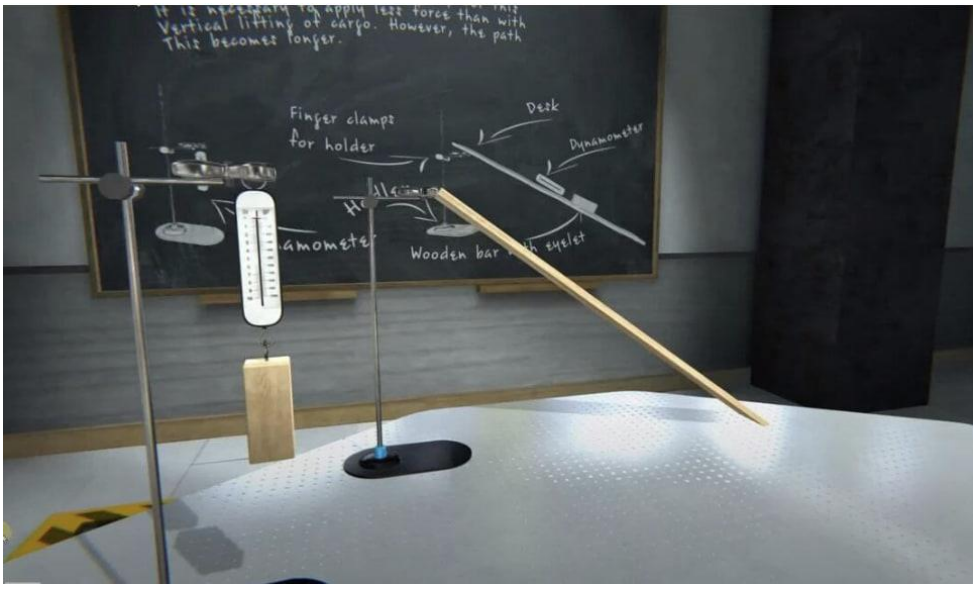

Рисунок 2.2 – Приклад віртуальної лабораторії від ROQED

Також навчально-дослідницьку лабораторію віртуальної та доповненої реальності «Ulab» було створено у Сумському державному університеті

Крім того, Міністерство освіти і науки України підтримує впровадження таких технологій, щоб надати студентам доступ до сучасних методів навчання і покращити рівень підготовки в різних сферах, включаючи STEM-дисципліни. Це також допомагає університетам адаптуватися до нових викликів, особливо у контексті сучасних цифрових технологій, що набувають поширення у світі.

Дистанційне навчання та адаптація до онлайн-формату підштовхнула освітню систему до впровадження дистанційних інструментів, серед яких віртуальні лабораторії посіли значне місце. В умовах віддаленого навчання вони стали незамінними для підтримки практичних занять, які було неможливо провести в звичайних класах і лабораторіях.

Віртуальні лабораторії дозволяють студентам працювати з потенційно небезпечними речовинами або експериментами без ризику для здоров’я. Це особливо важливо для таких дисциплін, як хімія чи фізика, де реальні експерименти можуть мати високі вимоги до безпеки.

Адаптація до індивідуальних потреб, використання віртуальних лабораторій дозволяє вчителям налаштовувати контент відповідно до рівня підготовки учнів, що підвищує ефективність навчання. Штучний інтелект, інтегрований у багато сучасних платформ, дозволяє персоналізувати навчальний процес і пропонувати завдання відповідного рівня складності для кожного студента.

Зростаючий інтерес у вищих навчальних закладах та професійній підготовці. В Україні зростає попит на підготовку фахівців у галузях природничих наук, технологій, інженерії та математики (STEM) [8]. Віртуальні лабораторії дозволяють забезпечити підготовку майбутніх спеціалістів навіть в умовах обмеженого доступу до фізичних лабораторій, що є критично важливим для розв'язання завдань професійного розвитку в інженерії, медицині та наукових дослідженнях.

Отже, аналіз показує, що впровадження віртуальних лабораторій значно сприяє модернізації освіти в Україні та відповідає глобальним трендам у цифровізації навчання. Використання 3D-технологій, таких як Blender, Maya та Unreal Engine 5, дає змогу створювати віртуальні лабораторії, які забезпечують високу деталізацію, інтерактивність і реалістичність симуляцій. Це дозволяє не лише підвищити ефективність навчання, а й залучити студентів до більш активної взаємодії з матеріалом.

Використання роботизованих систем у сфері екскурсійного обслуговування та забезпечення безпеки (рис. 2.3) відвідувачів набуває дедалі більшого поширення. Сучасні роботи здатні не лише проводити екскурсії, озвучуючи інформацію, але й попереджати про небезпечні зони, тим самим підвищуючи загальний рівень безпеки та комфорту під час відвідувань різноманітних об'єктів.

Одним із прикладів таких роботів є робот-гід, розроблений для музеїв та виставкових центрів. Цей робот оснащений системою навігації, що дозволяє йому самостійно пересуватися залами, слідуючи за заздалегідь визначеним маршрутом. Він здатен розпізнавати відвідувачів, вітати їх та надавати

інформацію про експонати. Завдяки вбудованим сенсорам та камерам, робот може визначати місцезнаходження відвідувачів та адаптувати свою поведінку відповідно до їхніх дій.

Крім того, такі роботи можуть бути обладнані системами розпізнавання мовлення, що дозволяє їм відповідати на запитання відвідувачів у режимі реального часу. Це забезпечує інтерактивність екскурсії та підвищує зацікавленість аудиторії. Додатково, роботи-гіди можуть використовувати мультимедійні засоби, такі як екрани або проектори, для демонстрації додаткових матеріалів, що доповнюють розповідь.

У контексті забезпечення безпеки, роботи можуть бути оснащені датчиками, що виявляють небезпечні зони або об'єкти. Наприклад, у промислових підприємствах роботи-помічники можуть попереджати працівників про наближення до небезпечних машин або зон з підвищеним ризиком. Вони можуть озвучувати попередження або використовувати візуальні сигнали для привернення уваги. Такі системи сприяють зниженню кількості нещасних випадків на виробництві та підвищують загальний рівень безпеки.

Іншим прикладом є роботи, що використовуються в аеропортах для інформування пасажирів про правила безпеки та попередження про небезпечні зони. Вони можуть надавати інформацію кількома мовами, що є особливо корисним у міжнародних терміналах. Крім того, такі роботи можуть допомагати пасажирам знаходити потрібні виходи або інші об'єкти інфраструктури, тим самим покращуючи загальний досвід перебування в аеропорту.

Варто зазначити, що розробка та впровадження таких роботизованих систем [9] вимагає врахування етичних та соціальних аспектів. Зокрема, необхідно забезпечити конфіденційність даних відвідувачів, які можуть збиратися роботами під час їхньої роботи. Також важливо враховувати можливі психологічні реакції людей на взаємодію з роботами та адаптувати дизайн і поведінку роботів відповідно до культурних та соціальних норм.

У підсумку, використання роботів для проведення екскурсій, озвучування інформації та попередження про небезпечні зони є перспективним напрямом

розвитку технологій. Такі системи підвищують ефективність надання інформації, забезпечують інтерактивність та сприяють підвищенню рівня безпеки в різноманітних середовищах. Однак, їхнє впровадження вимагає ретельного планування та врахування низки технічних, етичних та соціальних аспектів

Рисунок 2.3 – Приклад роботу

2.2 Постановка задач

Після ретельного аналізу можливостей доцільно розробити план виконання майбутнього проекту відповідно до вимог, визначених у технічному завданні (ТЗ).

Розробка віртуальної лабораторії для навчальних закладів ставить перед собою завдання створення інтерактивного навчального середовища, яке забезпечить студентам можливість проводити наукові експерименти у безпечному віртуальному просторі, що імітує реальне лабораторне обладнання та процеси. Необхідно створити систему, яка дозволить учням відпрацьовувати практичні навички, розуміти та закріплювати теоретичні знання, а також виконувати навчальні завдання незалежно від місця і часу. Задача включає:

– розробку 3D-моделей лабораторного обладнання;

– налаштування їхньої анімації для відтворення реальних процесів;

– інтеграцію штучного інтелекту для адаптації навчального контенту під індивідуальні потреби студентів;

– інтегрувати засоби збору даних для отримання статистики.

Середовище має відповідати вимогам сучасних технологій і забезпечувати високу продуктивність та інтерактивність за допомогою ігрових рушіїв і 3D-технологій, таких як Unreal Engine 5.

2.3 Актуальність

Зростання попиту на дистанційне навчання та сучасні технології в освіті зумовлює необхідність у нових інструментах для забезпечення якісного навчального процесу. Віртуальні лабораторії стають важливою складовою інтерактивного навчання, дозволяючи студентам отримувати практичні знання без ризику та значних витрат на матеріали й обладнання. У сучасних умовах, особливо в контексті України, де не всі освітні заклади мають належне матеріально-технічне оснащення, віртуальні лабораторії допомагають компенсувати брак фізичних ресурсів. Актуальність посилюється також глобальними освітніми тенденціями, де штучний інтелект і 3D-симуляції активно впроваджуються у навчальні програми для підвищення залученості та ефективності засвоєння матеріалу.

2.4 Завдання дослідження

Завданням проєкту буде створення віртуальної лабораторії, яка дозволить протестувати створення та проведення прототипу реальної лабораторної роботи.

Лабораторія має забезпечувати максимальну реалістичність і точність візуалізації завдяки високоякісному 3D-моделюванню та передовим технологіям, як-от Nanite і Lumen в Unreal Engine 5. Цей інструмент призначений для використання в освітніх установах різних рівнів та галузей, де навчальний процес включає необхідність виконання лабораторних дослідів та досліджень.

2.5 Вимоги та вхідні дані

Для розробки віртуального середовища потрібно мати відповідні засоби, такі як програмне забезпечення таке як Blender або Maya, ігровий рушій Unreal Engine 5, який забезпечує інтерактивність і продуктивність, а також інструменти для створення текстур і рігів. Вхідними даними є необхідні матеріали щодо технічних вимог, набори 3D-моделей, а також специфікації для обладнання, яке потрібно змоделювати та анімувати.

2.6 Розробка алгоритму дій

Був здійснений детальний аналіз великої кількості даних та джерел. Серед цих джерел можна виділити інтернет-ресурси, наукові статті та журнали, а також звіти.

Відповідно до технічного завдання, було розроблено алгоритм для майбутнього проєкту, який передбачає наступне:

– створити whitebox-прототип приміщення віртуальної лабораторії, орієнтуючись на матеріали, надані викладачем, щоб визначити базову структуру, розташування зон та оцінити ергономічність простору до етапу деталізації;

– розробити ТЗ на високоякісні 3D-моделі лабораторного обладнання з акцентом на деталізацію, створивши хайполь і лоуполь моделі, додавши текстури високої якості та коректно налаштовані UV-розгортки для досягнення реалістичного вигляду;

– підключити створені об'єкти до середовища Unreal Engine 5, розташувати їх відповідно до плану, забезпечивши підтримку технологій Nanite для оптимізації складних геометричних структур і Lumen для реалістичного освітлення та високої продуктивності;

– забезпечити інтерактивність моделі, створивши можливості реалістичної взаємодії з лабораторним обладнанням у реальному часі, включаючи активацію пристроїв, зміну параметрів і проведення віртуальних експериментів;

– розробити логіку роботи моделі робота для віртуальної лабораторії, протестувати її на різних пристроях, у тому числі в умовах віддаленого доступу, забезпечивши високу продуктивність і сумісність із навчальними платформами.

– створити програму для розпізнавання QR-кодів у реальному часі завдяки Open CV;

– розробити інструменти для збору інформації та статистики, які допоможуть відстежувати ефективність навчального процесу, аналізувати взаємодію користувачів із системою та покращувати її функціональність;

– провести тестування лабораторної установки у віртуальній лабораторії, оцінюючи її функціональність, зручність використання та відповідність навчальним цілям, враховуючи відгуки користувачів для оптимізації системи.

2.7 Висновок до другого розділу

У другому розділі було проведено детальний аналіз методів моделювання структури та плану реалізації проєкту, що включало вивчення наукових джерел, визначення ключових вимог та розробку алгоритму дій. Було розглянуто сучасні технології 3D-моделювання та програмування, які дозволяють створювати

реалістичні інтерактивні лабораторії, а також визначено переваги використання Unreal Engine 5 для реалізації проєкту, зокрема його можливості щодо високополігонального рендерингу (Nanite), глобального освітлення (Lumen) та інтеграції штучного інтелекту.

Розроблено концепцію інтерактивної віртуальної лабораторії, що підтримує 3D-моделювання, штучний інтелект і бази даних, дозволяючи створювати адаптивні навчальні середовища, що реагують на дії користувачів у режимі реального часу. Було визначено основні етапи створення лабораторії – від проєктування та програмування до тестування, що включає розробку детальних 3D-моделей обладнання, налаштування анімації, інтеграцію фізичних симуляцій та механізмів взаємодії з об’єктами.

Було сформовано алгоритм реалізації, що включає створення whitebox-прототипу, що дозволяє визначити розташування ключових елементів лабораторії та ергономіку віртуального простору. Далі здійснювалася розробка 3D-моделей з урахуванням оптимізації продуктивності та якості графіки, їх інтеграція в Unreal Engine 5, налаштування колізійних параметрів і фізичних властивостей об'єктів. Окремо було опрацьовано механізми взаємодії користувачів з об'єктами лабораторії через Blueprint-скрипти та C++, що дозволило створити повноцінне інтерактивне середовище.

Також розглянуто можливість використання комп’ютерного зору та алгоритмів штучного інтелекту для розпізнавання користувацьких дій і створення адаптивного навчального процесу. Для цього було реалізовано технології OpenCV та YOLOv5, що дозволяють здійснювати аналіз зображень, ідентифікацію об’єктів та автоматичний контроль виконання лабораторних завдань.

3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ ВІРТУАЛЬНОГО СЕРЕДОВИЩА

3.1 Збір інформації

Перший етап дослідження був присвячений збору та аналізу вихідних даних, необхідних для подальшого тестування та оптимізації роботи експериментального робота (рис. 3.1).

Основну увагу було приділено визначенню простору, в якому проводитимуться експерименти, а також його основним технічним параметрам. Місцем проведення тестування визначено лабораторію 160-2 кафедри комп’ютерно-інтегрованих технологій автоматизації та робототехніки (КІТАР) (рис. 3.2), що забезпечує оптимальні умови для експериментальної діяльності.

У ході підготовки до тестування було здійснено аналіз можливих варіантів маневру робота та виявлення ключових об’єктів, взаємодія з якими є критичною для коректного виконання експериментів. Варто зазначити, що розташування цих об’єктів може змінюватися залежно від результатів попередніх випробувань, що дає змогу ефективніше використовувати доступний простір та підвищити якість проведення тестувань.

Як зовнішні параметри середовища були взяті до уваги підсумковий план облаштування приміщення та план кафедри, який включає точне розташування основних вузлів і обладнання. До аналізу також були залучені фотографії та відеоматеріали, зроблені в межах кафедри, що дозволило створити віртуальну модель приміщення.

Додатково було отримано детальні зображення експериментального робота, його технічні характеристики, список робочих модулів та їх призначення.

Рисунок 3.1 – Фото експериментального робота «Хнурешник-1»

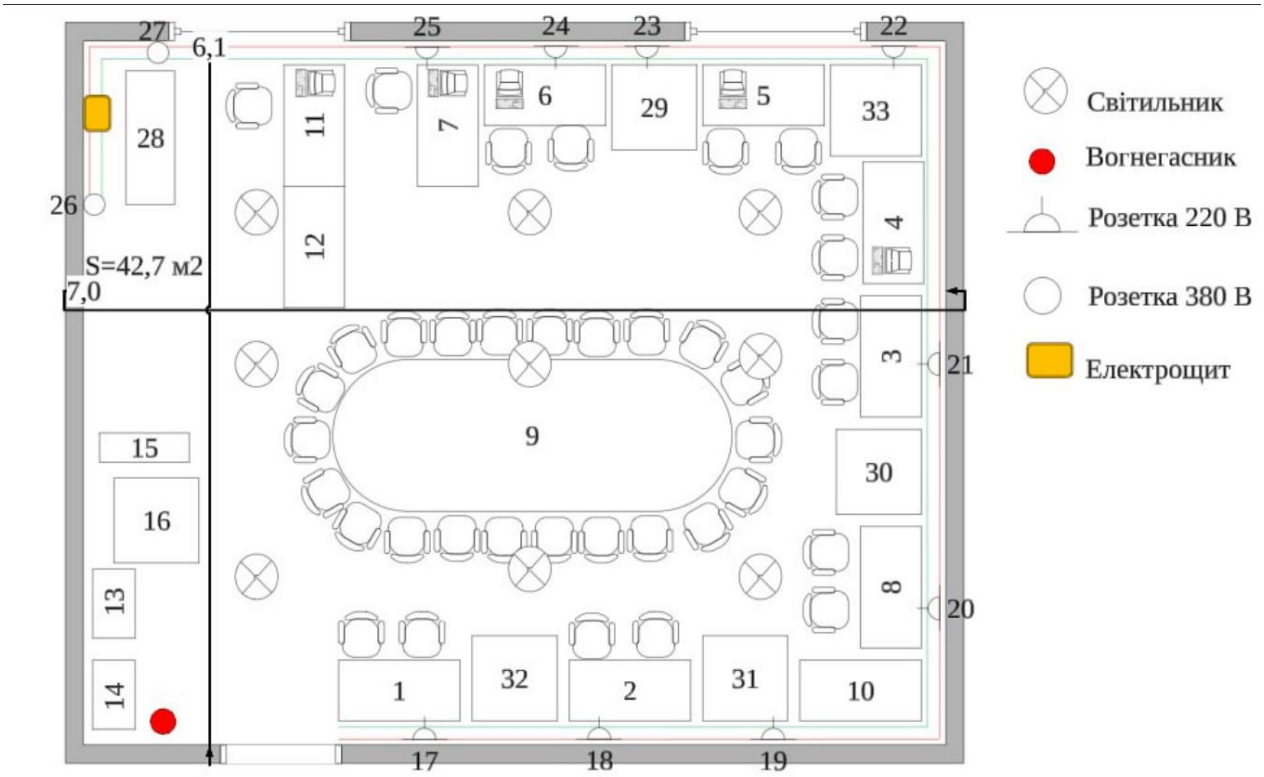

Рисунок 3.2 – План лабораторії кафедри

На основі аналізу зібраних даних у програмі для 3D-моделювання Blender була розроблена лоуполі-модель робота, що відображає його ключові геометричні та функціональні характеристики. Фото та план кафедри стали основою для створення віртуального блок-ауту середовища в Unreal Engine 5, що дозволило імітувати умови тестування у віртуальній реальності. Зазначений підхід є логічним продовженням попереднього дипломного проєкту, спрямованого на впровадження новітніх технологій у процес навчання та досліджень. Таким чином, перший етап роботи став базисом для подальшого створення повноцінного віртуального середовища, яке забезпечить високий рівень деталізації та інтерактивності, необхідний для ефективного тестування і вдосконалення робототехнічних систем.

3.2 Аналіз засобів розробка та вибір технології

Проєкт, що розглядається, базується на використанні сучасної технологічної платформи Unreal Engine 5 [10]. Цей ігровий двигун був обраний завдяки своїй універсальності, потужним можливостям візуалізації та інтеграції, а також попередньому досвіду роботи з ним у межах аналогічних проєктів. під час попередньої роботи, за допомогою Unreal Engine 5 було створено віртуальне середовище, яке тестувало здатність відтворювати візуальну різноманітність приладів, точність моделювання приміщень та показуючи високий рівень оптимізації на машинах різного покоління, завдяки чому студенти могли ознайомитися з функціоналом кафедри.

Результати попередньої роботи стали основою для подальшого розвитку проєкту (рис. 3.3 – 3.4) з акцентом на вирішення конкретного завдання – створення платформи для віртуального тестування можливостей експериментального робота ХНУРЕШНИК-1 та збору статистики для подальшого покращення робота та підготовки реального приміщення для тестування в реальному житті.

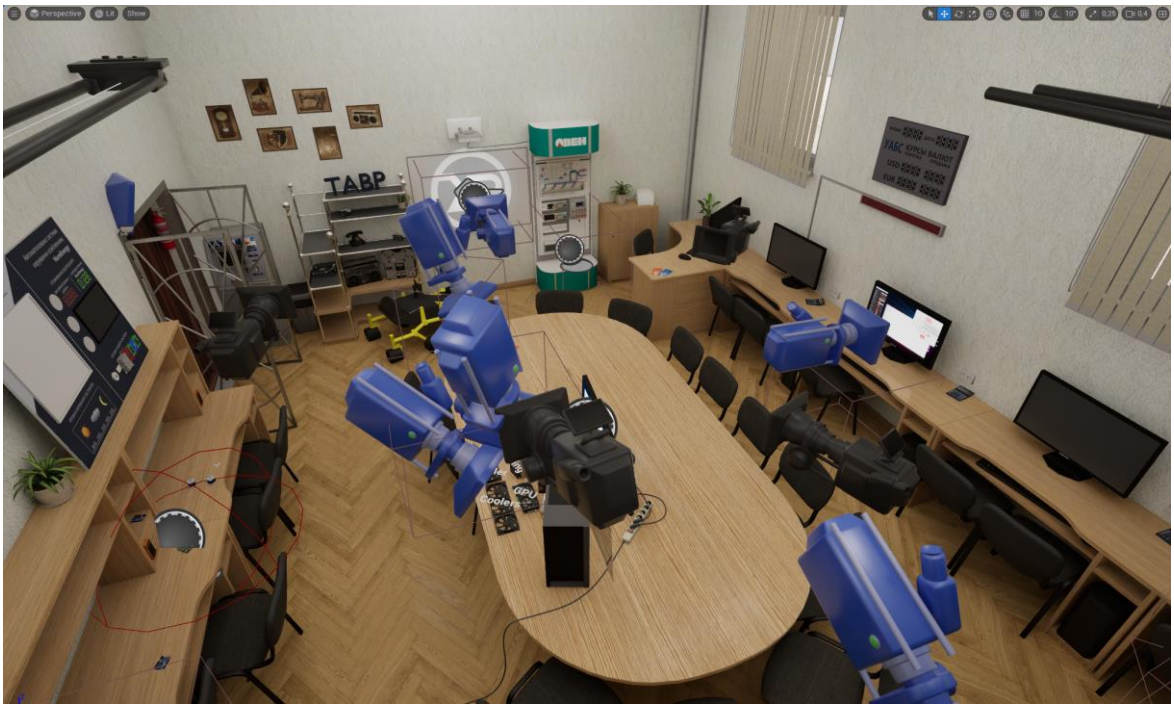

Рисунок 3.3 – Віртуальна версія лабораторії 162

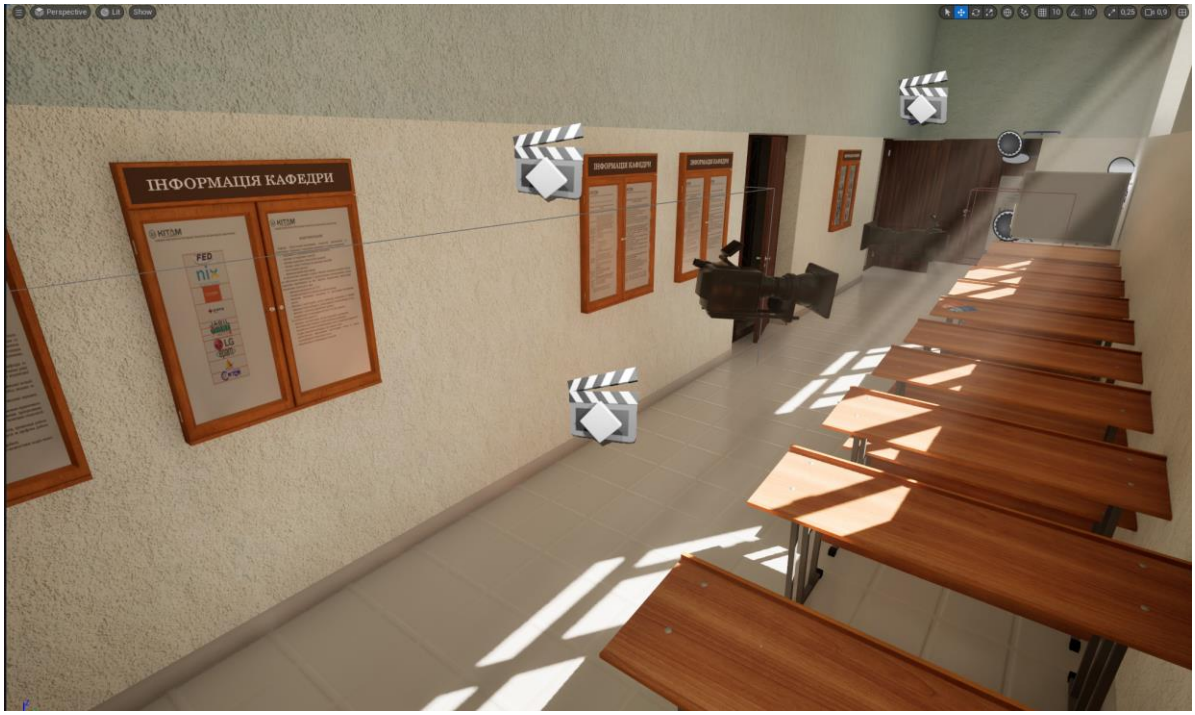

Рисунок 3.4 – Віртуальна версія коридору кафедри КІТАР

Unreal Engine 5, який вперше став доступним у квітні 2022 року, на момент початку роботи проєкту вже отримав значну кількість оновлень. Остання доступна версія – Unreal Engine 5.5 [11] – містить численні покращення, які значно розширили можливості цього інструменту.

Серед ключових нововведень варто відзначити оптимізацію системи Nanite що дозволяє ефективніше працювати з високополігональними об'єктами; покращення технології Lumen [12] для більш точної глобальної імітації освітлення; інтеграцію більшої кількості інструментів для розробки анімації; а також розширену підтримку плагінів, таких як OpenCV [13] та YOLOv5 [14].

Ці функції суттєво вплинули на ефективність і якість розробки проєкту, особливо у частині симуляції поведінки робототехнічних систем.

Однією з основних причин вибору Unreal Engine 5 для даного проєкту стало те, що він надає змогу створювати високоякісні візуалізації в реальному часі. Це особливо важливо для віртуального тестування роботів, де точність відтворення деталей і динамічність середовища мають вирішальне значення.

Крім того, платформа забезпечує високий рівень продуктивності та підтримує адаптивну оптимізацію, що дозволяє ефективно використовувати апаратні ресурси. Це стало критично важливим для створення інтерактивного середовища, яке може відтворювати поведінку експериментального робота в різних умовах.

У ході роботи над проєктом використання нових можливостей Unreal Engine 5 дозволило значно підвищити якість моделювання та взаємодії. Одним із ключових удосконалень стало спрощення інтеграції анімацій. Раніше цей процес вимагав значних витрат часу та ресурсів, однак у версіях починаючи з Unreal Engine 5.3 були впроваджені нові інструменти, що дозволяють автоматизувати значну частину роботи. Завдяки цьому стало можливим більш точно налаштовувати анімацію руху робота ХНУРЕшник-1, що критично важливо для тестування його функціональності.

Додатково, вбудована підтримка плагінів, таких як OpenCV і YOLOv5, відкрила нові горизонти для аналізу даних і машинного навчання у межах Unreal Engine 5. Зокрема, інтеграція OpenCV дозволила використовувати обчислювальні можливості комп'ютерного зору для аналізу навколишнього середовища та взаємодії робота з об'єктами. Використання YOLOv5, в свою чергу, надало змогу швидко й точно розпізнавати об'єкти в реальному часі, що значно підвищило ефективність симуляцій.

Однією з унікальних переваг Unreal Engine 5 є технології Nanite та Lumen, які стали революційними для віртуальної розробки. Nanite дозволяє використовувати геометрію об'єктів із надвисокою деталізацією без значних втрат продуктивності, що забезпечує високу якість візуалізації навіть на середньопродуктивних пристроях. Lumen, у свою чергу, дає можливість створювати реалістичні сцени завдяки динамічному освітленню та глобальній імітації світла, що є критично важливим для симуляції умов роботи робота.

Серед інших переваг Unreal Engine 5 варто відзначити його мультиплатформенність і доступність інструментів для командної роботи. Ця платформа підтримує інтеграцію з Git [15], що дозволяє ефективно організувати розробку у командному середовищі та легко впроваджувати зміни у проєкт. Також важливою перевагою є зручний інтерфейс користувача, що дозволяє швидко адаптуватися до нових функцій і інструментів навіть розробникам із базовим рівнем підготовки.

На етапі розробки було враховано всі зазначені особливості та вдосконалення Unreal Engine 5. Це дозволило створити віртуальне середовище, яке максимально відповідає завданням проєкту. Зокрема, для реалізації тестувань були впроваджені моделі високої точності, що відтворюють геометрію робота ХНУРЕшник-1, його функціональні вузли та сценарії взаємодії з навколишнім середовищем. У результаті вдалося досягти високого рівня деталізації, що дозволяє ефективно імітувати поведінку системи та тестувати її можливості у віртуальних умовах.

Слід також зазначити, що розвиток цього проєкту став можливим завдяки гнучкості Unreal Engine 5 у реалізації складних технічних завдань. Додаткові оновлення платформи надали змогу швидше впроваджувати нові ідеї, зокрема адаптивну взаємодію системи з різними сценаріями тестування. Наприклад, вбудовані інструменти для створення процедурних об'єктів значно спростили розробку варіативного середовища, що дозволяє моделювати різні конфігурації приміщень і перешкод. Ці можливості є вирішальними для тестування експериментального робота в умовах, наближених до реальних.

Загальні покращення Unreal Engine 5 також включають більш зручний інтерфейс користувача, розширену підтримку багатокористувацького режиму та вдосконалення системи управління проєктами. Це дозволяє ефективніше організовувати командну роботу, спрощуючи інтеграцію змін і тестування нових функцій. Усе це сприяє не лише підвищенню продуктивності розробників, але й забезпечує високу гнучкість під час роботи з проєктом.

Таким чином, удосконалення Unreal Engine 5, впроваджені в останніх версіях, стали важливими для реалізації поставлених завдань проєкту. Завдяки цим нововведенням вдалося створити більш реалістичне, точне та динамічне віртуальне середовище для тестування можливостей експериментального робота ХНУРЕшник-1.

Однією з ключових задач проєкту стало розпізнавання QR-кодів [16], яке потребувало вибору відповідного інструмента. OpenCV був обраний з-поміж інших рішень завдяки своїй високій продуктивності, гнучкості та доступності.

Основними конкурентами OpenCV у цій сфері є бібліотеки, такі як ZBar[17], ZXing [18] та ML Kit [19]. Однак OpenCV має кілька суттєвих переваг. По-перше, він підтримує широкий спектр мов програмування, що забезпечує інтеграцію з різними платформами. По-друге, бібліотека OpenCV має високу швидкість обробки зображень завдяки використанню оптимізованих алгоритмів і підтримці апаратного прискорення. Це дозволяє досягти високої точності та швидкості розпізнавання навіть у реальному часі.

Порівнюючи з конкурентами, ZBar забезпечує швидке розпізнавання кодів, але має обмежену підтримку форматів і не настільки гнучкий у налаштуваннях. ZXing, у свою чергу, добре працює з різними форматами штрих-кодів, але поступається OpenCV у продуктивності при обробці великої кількості даних. ML Kit, розроблений Google, також пропонує якісне розпізнавання QR-кодів, але його інтеграція у середовище Unreal Engine є складнішою та менш документованою.

OpenCV також надає розширені можливості, які виходять за межі розпізнавання QR-кодів. Зокрема, бібліотека підтримує операції з обробки зображень, такі як фільтрація, сегментація, детекція контурів та інші. У контексті Unreal Engine 5 це дозволяє створювати складні алгоритми аналізу зображень, які можуть бути використані для покращення симуляцій. Наприклад, OpenCV може бути застосований для аналізу текстур об'єктів або обчислення характеристик навколишнього середовища в реальному часі.

У рамках нашого проєкту OpenCV дозволив реалізувати ефективне розпізнавання QR-кодів, забезпечивши точність і стабільність навіть у складних сценах з низьким рівнем освітлення або значними спотвореннями. Це стало ключовим фактором для симуляції поведінки робота ХНУРЕшник-1 у віртуальному середовищі.

Однією з ключових задач проєкту стало розпізнавання QR-кодів, яке потребувало вибору відповідного інструмента. OpenCV був обраний з-поміж інших рішень завдяки своїй високій продуктивності, гнучкості та доступності. Основними конкурентами OpenCV у цій сфері є бібліотеки, такі як ZBar, ZXing та ML Kit. Однак OpenCV має кілька суттєвих переваг. По-перше, він підтримує широкий спектр мов програмування, що забезпечує інтеграцію з різними платформами. По-друге, бібліотека OpenCV має високу швидкість обробки зображень завдяки використанню оптимізованих алгоритмів і підтримці апаратного прискорення. Це дозволяє досягти високої точності та швидкості розпізнавання навіть у реальному часі.

Порівнюючи з конкурентами, ZBar забезпечує швидке розпізнавання кодів, але має обмежену підтримку форматів і не настільки гнучкий у налаштуваннях. ZXing, у свою чергу, добре працює з різними форматами штрих-кодів, але поступається OpenCV у продуктивності при обробці великої кількості даних. ML Kit, розроблений Google, також пропонує якісне розпізнавання QR-кодів, але його інтеграція у середовище Unreal Engine є складнішою та менш документованою.

OpenCV також надає розширені можливості, які виходять за межі розпізнавання QR-кодів. Зокрема, бібліотека підтримує операції з обробки зображень, такі як фільтрація, сегментація, детекція контурів та інші. У контексті Unreal Engine 5 це дозволяє створювати складні алгоритми аналізу зображень, які можуть бути використані для покращення симуляцій. Наприклад, OpenCV може бути застосований для аналізу текстур об'єктів або обчислення характеристик навколишнього середовища в реальному часі.

Інтеграція OpenCV з Unreal Engine 5 має свої особливості. Одним із викликів є необхідність компіляції бібліотеки для використання у середовищі UE5. Цей процес може бути складним для недосвідчених розробників, оскільки вимагає врахування сумісності версій і залежностей. Також виникають складнощі з налаштуванням шляхів до бібліотек і обробкою специфічних для UE5 форматів даних. Однак ці труднощі компенсуються можливістю створення ефективних і точних алгоритмів розпізнавання та аналізу, які відповідають потребам проєкту.

У рамках нашого проєкту OpenCV дозволив реалізувати ефективне розпізнавання QR-кодів, забезпечивши точність і стабільність навіть у складних сценах з низьким рівнем освітлення або значними спотвореннями. Це стало ключовим фактором для симуляції поведінки робота ХНУРЕшник-1 у віртуальному середовищі. Завдяки використанню цієї бібліотеки вдалося підвищити продуктивність і якість розробки, що дозволяє забезпечити виконання поставлених завдань на найвищому рівні.

Для реалізації проєкту, де беруть участь двоє розробників, а обсяг даних становить близько 4 ГБ, важливим етапом було вибрати відповідну систему контролю версій. Основними вимогами були зручність у використанні, швидкість обробки великих файлів, інтеграція з існуючими інструментами розробки та підтримка командної роботи. Після аналізу різних варіантів було обрано GitHub [20], який відповідає усім цим критеріям.

GitHub є однією з найпопулярніших платформ для хостингу репозиторіїв, яка забезпечує зручний веб-інтерфейс і потужний набір інструментів для командної роботи. Однією з основних причин вибору цієї платформи є її інтеграція з іншими інструментами, такими як Visual Studio [21] та Unreal Engine. GitHub підтримує роботу з великими файлами через систему Git Large File Storage (LFS) [22], що дозволяє зручно керувати об'ємними ресурсами, такими як 3D-моделі та текстури, які є важливою частиною проєкту.

У порівнянні з іншими платформами, такими як GitLab та Bitbucket, GitHub має кілька ключових переваг. GitLab пропонує широкі можливості для налаштування внутрішньої інфраструктури, але це вимагає додаткових ресурсів для розгортання та підтримки. Bitbucket, у свою чергу, добре інтегрується з іншими продуктами Atlassian, але поступається GitHub у популярності та кількості доступних плагінів. GitHub забезпечує найбільшу кількість інтеграцій і має широку спільноту, що дозволяє швидко знаходити рішення для виникаючих проблем.

Для нашого проєкту використання GitHub забезпечило централізоване зберігання коду, простоту в управлінні версіями та зручність у відстеженні змін. Завдяки функціоналу pull requests стало можливим ефективно організувати процес рев’ю коду, що дозволяє покращувати якість розробки. Також платформа надає інструменти для автоматизації процесів, такі як GitHub Actions, які можуть бути використані для автоматичного тестування та збірки проєкту.

Особливо варто відзначити інтеграцію GitHub із системою Unreal Engine, що дозволяє легко синхронізувати зміни в проєкті та забезпечувати стабільну роботу навіть при великій кількості файлів. Підтримка Git LFS забезпечила

швидкий доступ до великих ресурсів, що є критичним для збереження продуктивності під час роботи над проєктом. Загалом, вибір GitHub як системи контролю версій став оптимальним рішенням, яке відповідає потребам проєкту та забезпечує високий рівень зручності й ефективності командної роботи.

Visual Studio було обрано як основне середовище розробки для написання коду мовою C++ у рамках нашого проєкту. Цей вибір обумовлений рядом факторів, що стосуються як сумісності з Unreal Engine 5, так і функціональних можливостей середовища. Visual Studio, як один із провідних інструментів для розробки програмного забезпечення, має численні переваги, які роблять його ідеальним вибором для створення високопродуктивних проєктів у середовищі UE5.

Однією з ключових причин вибору Visual Studio є її тісна інтеграція з Unreal Engine. Epic Games офіційно рекомендує Visual Studio для розробки на C++, що підтверджується підтримкою специфічних для Unreal функцій, таких як створення класів, налаштування модулів і використання інструментів діагностики. Крім того, Visual Studio забезпечує потужну систему автозаповнення коду (IntelliSense), яка значно полегшує роботу з великими кодовими базами. Це особливо важливо для нашого проєкту, оскільки код містить складні алгоритми для взаємодії робота ХНУРЕшник-1 із віртуальним середовищем.

Серед конкурентів Visual Studio можна назвати CLion, Code::Blocks та Qt Creator. CLion, хоч і є сучасним інструментом для розробки на C++, має меншу сумісність з Unreal Engine і обмежену підтримку плагінів, що ускладнює його використання в масштабних проєктах. Code::Blocks, хоча й легкий та зручний, поступається Visual Studio у функціональності, особливо в частині налагодження й інтеграції з іншими інструментами. Qt Creator відомий своєю зручністю для розробки графічних інтерфейсів, але не забезпечує необхідної підтримки специфічних функцій Unreal Engine. Visual Studio, натомість, надає повний набір інструментів для інтеграції, налагодження та оптимізації.

Ще однією перевагою Visual Studio є її розширюваність. Використання плагінів, таких як Visual Assist, значно спрощує навігацію по коду, оптимізує процес написання й аналізу складних програм. Для нашого проєкту це дозволило ефективно працювати з великими класами та об'єктами, характерними для Unreal Engine. Крім того, Visual Studio має широкий спектр засобів для налагодження, включаючи можливість роботи з багатопотоковими додатками, що було важливо для тестування поведінки робота у віртуальному середовищі.

Мова програмування C++ також була обрана через її тісний зв'язок з Unreal Engine. Unreal Engine надає широкий API, написаний саме цією мовою, що дозволяє гнучко й ефективно реалізовувати складні алгоритми й взаємодії. C++ забезпечує високу продуктивність, контроль над пам’яттю і сумісність з іншими мовами та бібліотеками, такими як OpenCV і YOLOv5, що активно використовуються в нашому проєкті.

Для запису озвучення робота ми обрали платформу Narakeet через її унікальні технічні можливості та переваги, які забезпечують високий рівень якості і гнучкості у створенні голосових повідомлень. Насамперед, Narakeet пропонує широкий вибір синтетичних голосів, які максимально наближені до природного звучання, що дозволило нам створити озвучення, яке викликає довіру і забезпечує комфортне сприйняття для користувачів. Платформа підтримує багатомовність, що стало важливим фактором для нас, адже дозволяє інтегрувати різні мовні варіації для майбутнього розширення функціоналу нашого робота.

Ще однією перевагою Narakeet [23] є зручність у використанні: інтуїтивно зрозумілий інтерфейс і підтримка текстових сценаріїв дозволяють легко генерувати озвучення без необхідності володіння професійними навичками звукорежисури. Окрім цього, сервіс підтримує інтеграцію з сучасними технологіями штучного інтелекту, що відкриває можливість налаштування інтонації, ритму і швидкості мовлення, адаптуючи голос під специфіку сценарію. Нарешті, Narakeet забезпечує швидкий рендеринг аудіофайлів у високій якості, що дозволило значно скоротити час на створення озвучення і зосередитися на

інших аспектах розробки проєкту. Усі ці фактори стали вирішальними при виборі платформи для запису голосу робота.

3.3 Налаштування програмного середовища

Для початку оновлення необхідно завантажити нову версію через Epic Games Launcher (рис. 3.5), який автоматично перевіряє доступність останніх версій двигуна. Після цього проводиться інсталяція, під час якої користувач може вибрати необхідні модулі й плагіни. Після завершення встановлення слід відкрити проєкт у новій версії двигуна, де система автоматично пропонує міграцію існуючих даних.

На цьому етапі важливо перевірити сумісність усіх використовуваних плагінів та бібліотек, а також протестувати функціональність проєкту в новому середовищі. У разі виникнення проблем з сумісністю, може знадобитися оновлення або заміна певних компонентів проєкту. Було интегровано нові функції та покращення Unreal Engine без втрат для стабільності й продуктивності проєкту.

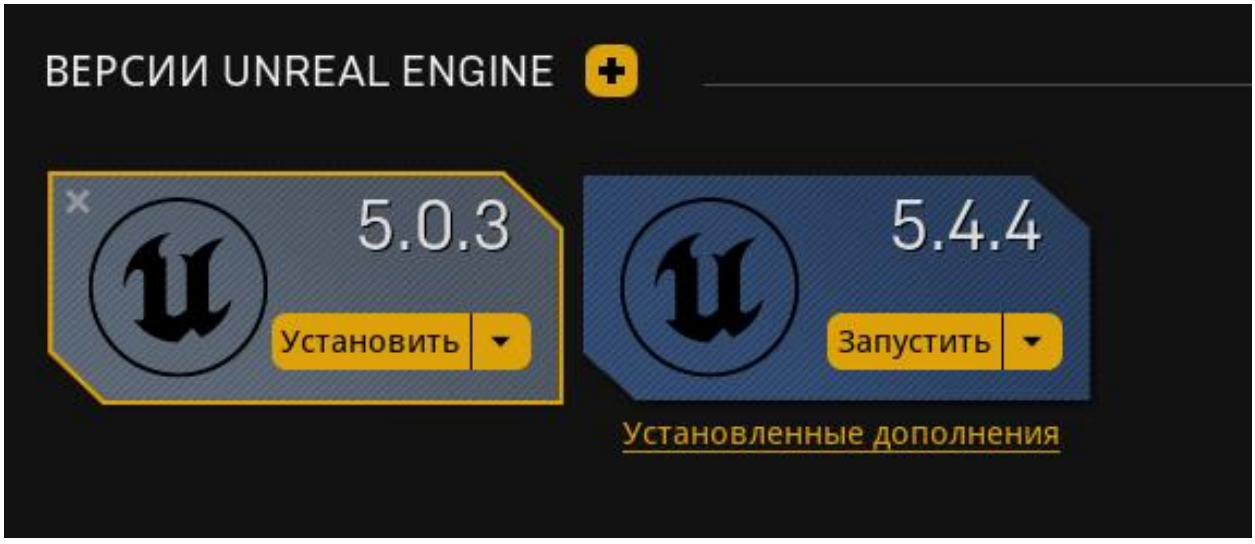

Рисунок 3.5 – Прикад оновлення Unreal Engine 5

Для повноцінного функціонування та ефективної роботи в середовищі Unreal Engine 5 необхідно встановити низку ключових плагінів, які забезпечують розширення базового функціоналу для нашого пректу . Одним із найважливіших плагінів є OpenCV. Цей плагін, що базується на бібліотеці Open Source Computer Vision, дозволяє реалізовувати широкий спектр задач, пов'язаних із комп'ютерним зором, таких як розпізнавання об'єктів, аналіз зображень, трекінг руху та багато іншого. OpenCV є незамінним інструментом для проєктів, що вимагають аналізу та обробки візуальної інформації, оскільки він забезпечує швидкість, точність і зручність у використанні.

Ще одним важливим компонентом є плагін SQLite, який дозволяє інтегрувати бази даних безпосередньо в середовище Unreal Engine. SQLite забезпечує локальне збереження даних, що є критично важливим для додатків, які повинні працювати без постійного доступу до мережі. Цей плагін надає можливість створювати, зберігати та маніпулювати структурованими даними, що дозволяє розробникам ефективно організовувати інформацію та забезпечувати її швидкий доступ. SQLite є особливо корисним у випадках, коли проєкт передбачає інтерактивність та взаємодію з користувачем через бази даних, наприклад, у створенні освітніх середовищ або симуляцій.

Плагін Restart Editor [24] також відіграє важливу роль у розробці проєктів в Unreal Engine. Його основна функція – забезпечення швидкого перезапуску редактора, що особливо актуально в ситуаціях, коли вносяться критичні зміни в конфігурацію проєкту або додаються нові плагіни. Завдяки Restart Editor значно скорочується час, необхідний на перезапуск, що підвищує ефективність роботи розробників і зменшує можливі перерви у процесі розробки.

Плагін Read Local txt [25] (рис. 3.6) дозволяє зчитувати текстові файли безпосередньо з локального сховища, що забезпечує простоту доступу до зовнішніх текстових даних. Цей плагін корисний у випадках, коли необхідно інтегрувати в проєкт динамічні текстові дані, наприклад, сценарії, описові матеріали чи навчальні модулі. Використання ReadLocal txt сприяє автоматизації процесів і мінімізує потребу у ручному внесенні змін до текстових ресурсів.

Blueprint Library є ще одним надзвичайно корисним плагіном, який розширює можливості роботи з Blueprints – візуальними сценаріями, що використовуються для створення ігрової логіки без написання коду. Цей плагін надає доступ до додаткових функцій, які не входять до стандартного набору Unreal Engine, що дозволяє розробникам значно розширити функціонал проєктів. Blueprint Library є незамінним для тих, хто працює з великими та складними проєктами, оскільки вона забезпечує зручність, гнучкість і ефективність у роботі.

Наступним кроком було завантаження та встановлення бібліотеки OpenCV. Було завантажено останню стабільну версію з офіційного сайту OpenCV у форматі .exe, що дозволило спростити процес інсталяції.

Після запуску інсталяційного файлу бібліотека була розпакована у директорію C:\OpenCV, яка містила необхідні підкаталоги, такі як bin, include, lib та інші. Ця структура забезпечила доступ до всіх компонентів, потрібних для роботи OpenCV.

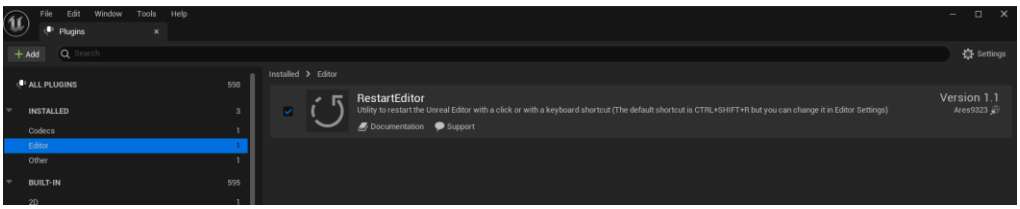

Рисунок 3.6 – Прикад підключсеного плагіну Read Local txt

Наступним важливим етапом стало налаштування системних змінних середовища. Ми додали шлях до виконуваних файлів бібліотеки C:\OpenCV\build\x64\vc16\bin у змінну середовища Path. Це дало змогу операційній системі знаходити потрібні динамічні бібліотеки під час виконання програм. Для цього ми скористалися налаштуваннями системних параметрів, зокрема меню "Environment Variables".

Для реалізації програмування на C++ було завантажено та встановлено Visual Studio, де ми обрали модуль "Desktop Development with C++". У Visual Studio був створений новий консольний проєкт.

Під час налаштування проєкту в "Project Properties" було додано необхідні шляхи: до директорії C:\OpenCV\build\include (рис. 3.7) для включення заголовкових файлів, а також до C:\OpenCV\build\x64\vc16\lib для підключення бібліотек. У секції "Additional Dependencies" до списку були додані файли opencv_world< >d.lib для режиму Debug та opencv_world< >.lib для Release.

Також необхідно встановити всі необхідні інструменти Visual Studio для інтеграції з Unreal Engine 5. Для цього потрібно встановити компоненти: "Розробка ігор на C++", "Розробка класичних програм на C++", а також кілька додаткових компонентів (рис. 3.8).

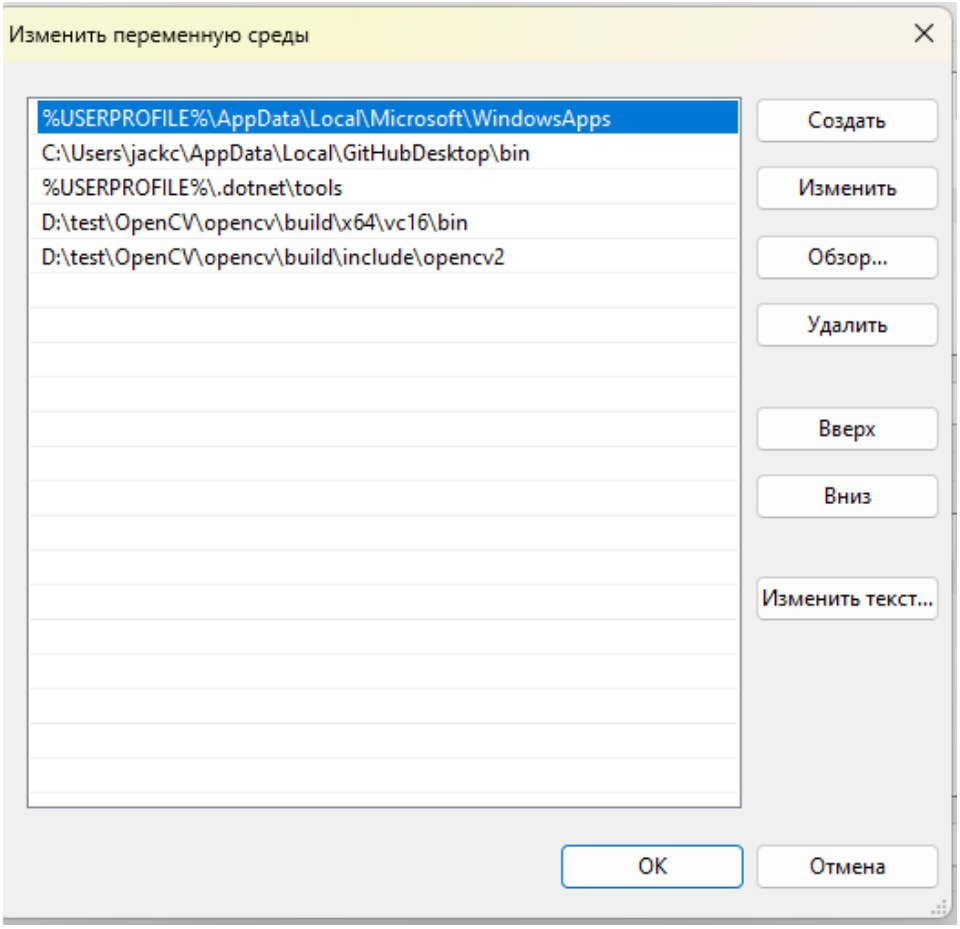

Рисунок 3.7 – Шлях до виконуваних файлів бібліотеки OpenCV

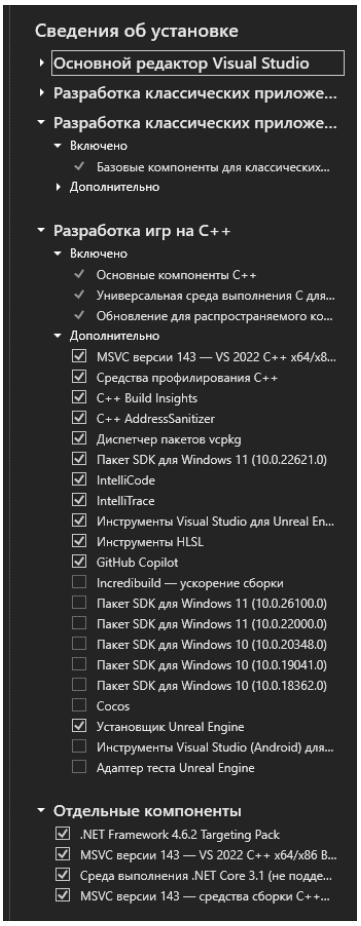

Рисунок 3.8 – Необхідні інструменти Visual Studio для інтеграції з Unreal Engine 5.

Для забезпечення ефективного управління версіями та колективної розробки проєкту в Unreal Engine 5, було виконано інтеграцію з платформою GitHub.

Перш за все, було завантажено й інстальовано систему контролю версій Git. Для цього ми скористалися офіційним вебсайтом Git, де завантажили інсталяційний файл для операційної системи Windows. Після завершення завантаження було розпочато процес інсталяції, де ми обрали стандартні параметри без змін у конфігурації. Після встановлення Git, для налаштування системи було відкрито Git Bash, де виконано конфігурацію користувача за допомогою команд: git config --global user.name "Ваше_ім’я" і git config --global user.email. Ці параметри дали змогу вказати ім’я розробника та адресу

електронної пошти, що використовуватимуться у записах про зміни в репозиторії.

Далі було створено новий репозиторій на платформі GitHub (рис. 3.9). Для цього ми увійшли до свого облікового запису GitHub, перейшли до панелі керування та натиснули кнопку "New Repository". У вікні створення репозиторію вказали його назву, обрали параметри доступу (приватний репозиторій) і натиснули "Create Repository". У результаті було отримано інструкції для підключення локального проєкту до віддаленого репозиторію.

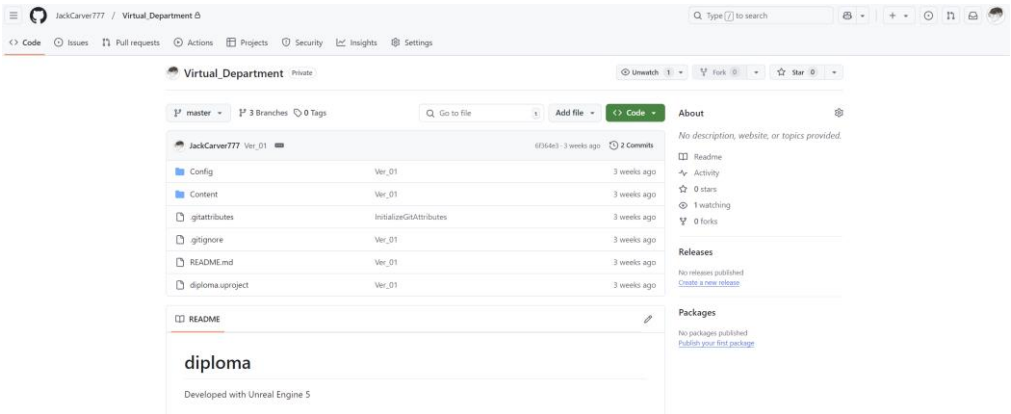

Рисунок 3.9 – Рипозиторій на сторінці GitHub

Наступним етапом стало налаштування Unreal Engine 5 для роботи з системою контролю версій. У середовищі Unreal Engine ми відкрили проєкт і активували плагін Git. Для цього перейдімо до меню "Edit" → "Plugins" і скористалися рядком пошуку, щоб знайти плагін "Source Control". Було активовано "Git Source Control", після чого Unreal Engine запропонував перезавантажити редактор для застосування змін.

Після повторного запуску Unreal Engine ми перейшли до меню "Source Control" у верхній панелі та обрали "Connect to Source Control". У діалоговому вікні, що відкрилося, ми вказали систему контролю версій "Git (beta)", а також надали шлях до виконуваного файлу Git (C:\Git\bin\git.exe) (рис. 3.10) і URL

нашого віддаленого репозиторію, який ми створили на GitHub. Для приватного репозиторію було введено облікові дані, що забезпечило успішну авторизацію.

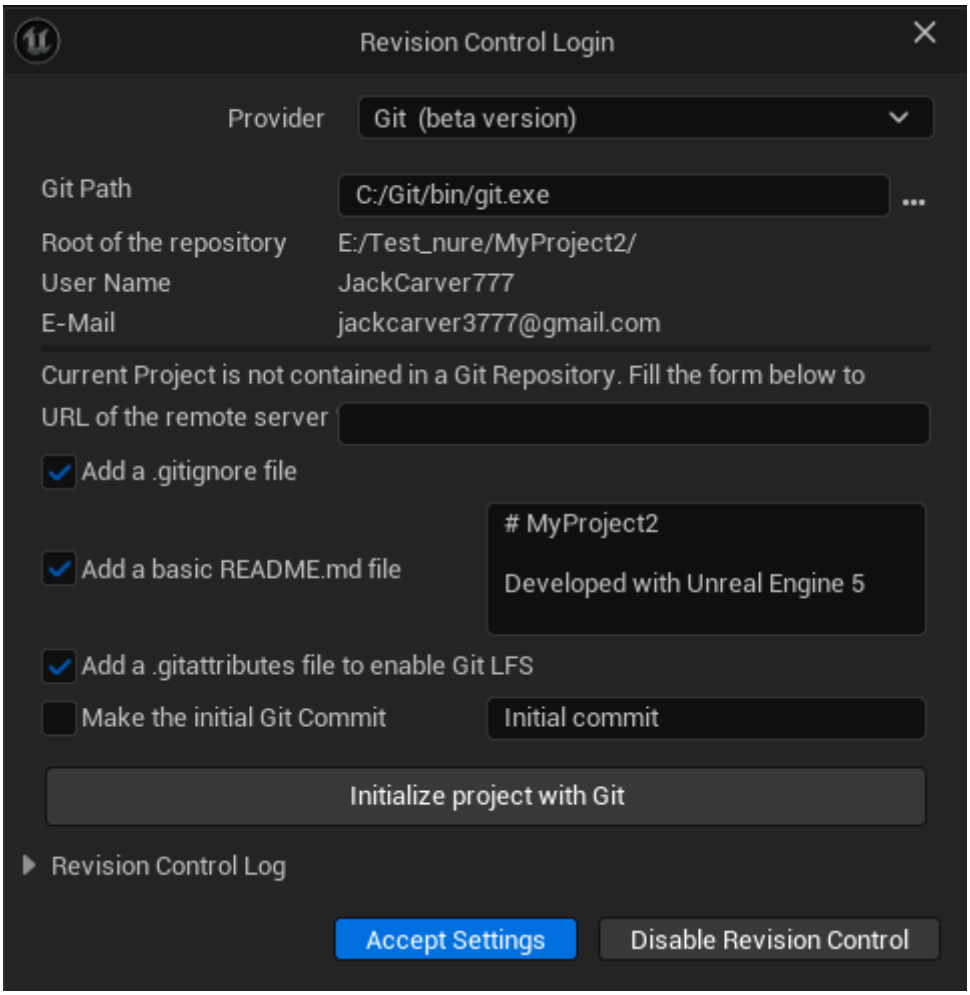

Рисунок 3.10 – Меню підключення Git

Після збереження налаштувань Unreal Engine автоматично розпізнав поточний стан локального проєкту та ініціалізував репозиторій. Було створено початковий коміт, що містив усі необхідні файли проєкту. Для цього ми скористалися меню "Source Control" і обрали пункт "Submit to Source Control". У вікні коміту було зазначено опис початкового стану проєкту, після чого всі файли було завантажено до віддаленого репозиторію на GitHub.

На завершення, ми провели перевірку функціональності інтеграції. Для цього було внесено тестові зміни до проєкту, створено коміт і виконано push до

віддаленого репозиторію. Усі дії було успішно виконано, що підтвердило коректну роботу налаштувань.

Git LFS – це розширення для Git, яке дозволяє зберігати великі файли або файли, що часто змінюються, у спеціалізованому сховищі. Замість збереження великих файлів безпосередньо в репозиторії, Git LFS створює посилання на ці файли та зберігає їх у віддаленому сховищі, що значно зменшує розмір репозиторію та покращує продуктивність роботи з ним. Це особливо актуально для проєктів, які включають мультимедійні файли, графіку, відео або великі набори даних. Використання Git LFS передбачає встановлення відповідного розширення та конфігурування репозиторію для роботи з файлами певних типів, які мають зберігатися через цей механізм. Наприклад, команди git lfs track "*.psd" і git add .gitattributes дозволяють додати правила для відстеження файлів Photoshop. Завдяки цьому розширенню розробники можуть уникнути перевантаження репозиторію та зберегти історію змін у компактному вигляді.

Git Ignore, у свою чергу, є конфігураційним файлом .gitignore, який визначає перелік файлів і директорій, які не повинні бути включені до репозиторію або відстежуватись Git. Це дозволяє уникнути додавання тимчасових, службових чи конфіденційних файлів, наприклад, файлів налаштувань середовища розробки, логів, кешів або об'єктних файлів, що генеруються автоматично. Файл .gitignore базується на простих текстових шаблонах, які визначають правила ігнорування. Наприклад, запис *.log вказує на ігнорування всіх файлів із розширенням .log, а запис build/ – на ігнорування всієї директорії build.

Використання Git LFS (рис. 3.11) і Git Ignore (рис. 3.12) дозволяє забезпечити оптимізоване управління репозиторіями, уникнути перевантаження системи контролю версій та спростити співпрацю в команді, особливо для великих проєктів із значною кількістю даних.

Рисунок 3.11 – Підключення Git LFS

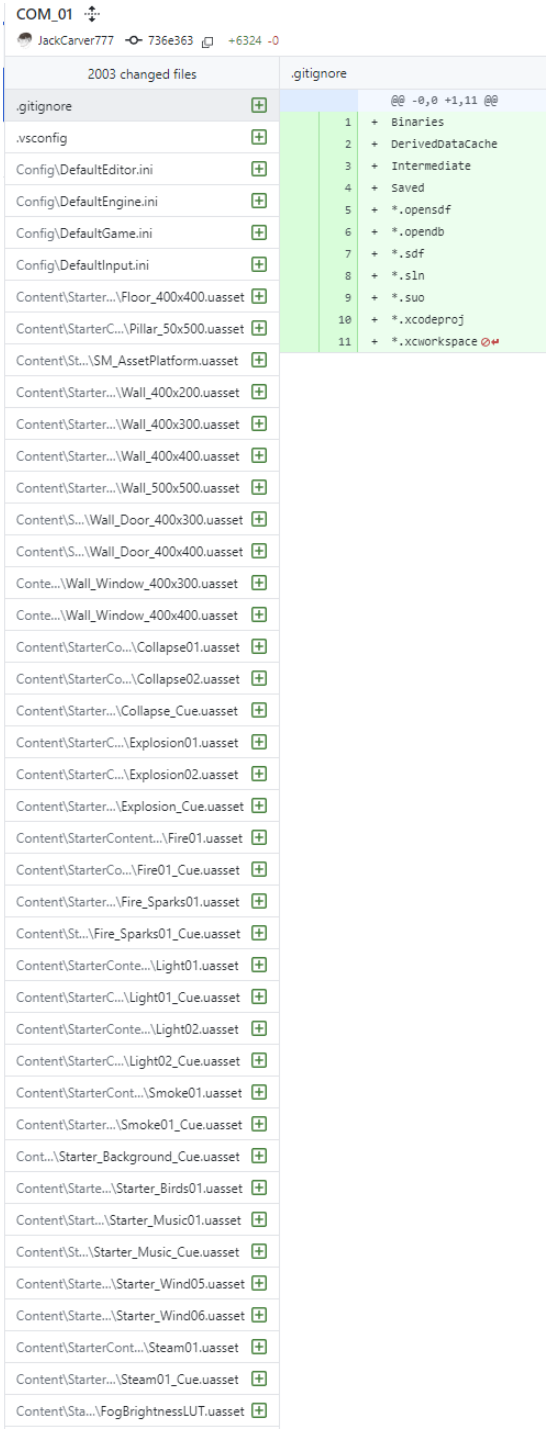

Рисунок 3.12 – Використання Git Ignore

3.4 Налаштування віртуального простору

Друге приміщення віртуальної кафедри, відтворене у середовищі Unreal Engine 5, одним із перших етапів стало створення whitebox-моделі.

Whitebox (рис. 3.13) є базовою тривимірною концепцією приміщення, що виконується на початкових стадіях проєктування і складається з простих геометричних форм, таких як куби, циліндри та площини.

Метою whitebox є попереднє визначення просторової організації кімнати, розміщення ключових об'єктів, перевірка ергономіки і масштабу приміщення, а також створення основи для подальшого заповнення деталями та декораціями.

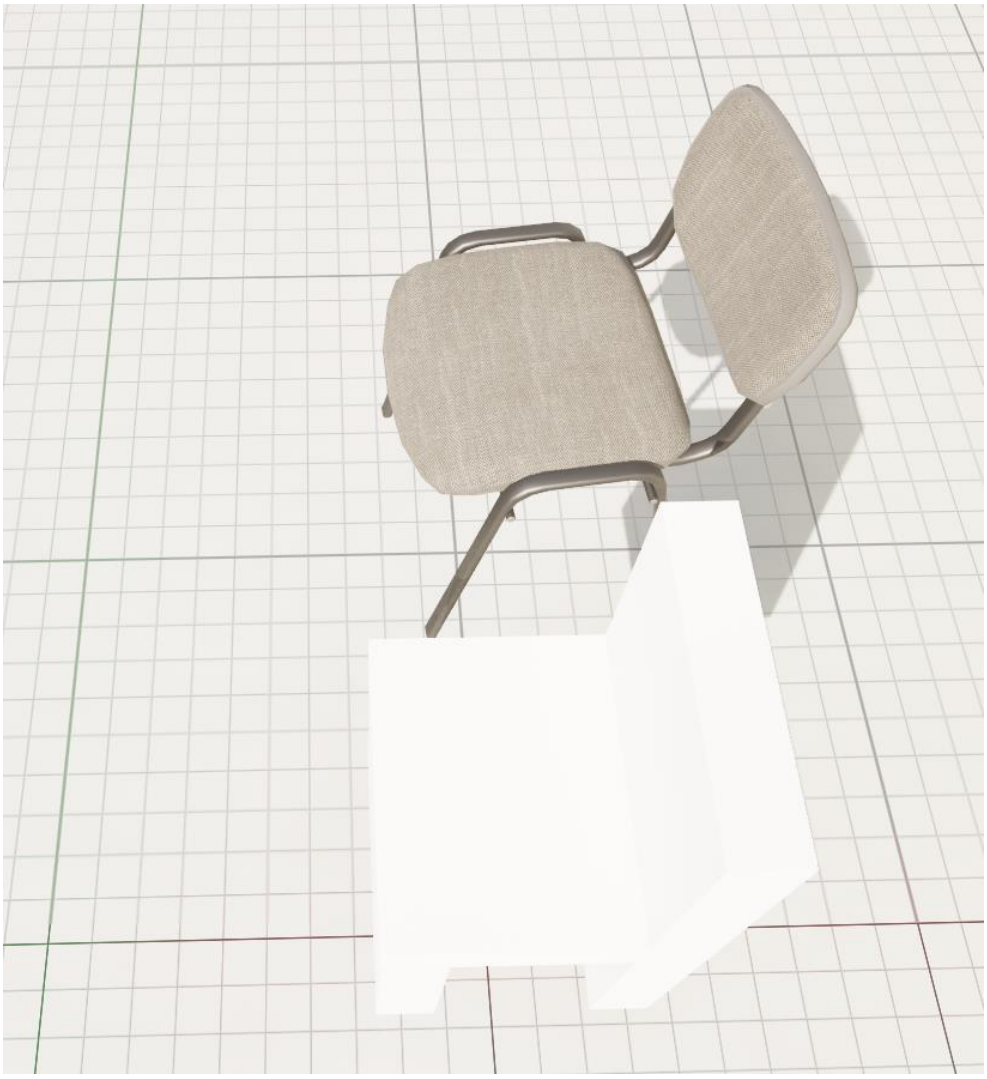

Рисунок 3.13 – Стілець у форматі whitebox в порівнянні з готовим варіантом

Після завершення цього етапу, ми змогли більш точно уявити структуру кімнати і перейти до інтеграції раніше створених елементів, які вже використовувалися в першому приміщенні кафедри.

Частина об’єктів, які були створені для першої кімнати, одразу знайшли своє застосування у другій кімнаті, що значно прискорило процес її наповнення. Такі елементи, як столи, стільці, полиці та базові освітлювальні прилади, не потребували додаткової адаптації, оскільки їхні параметри відповідали загальному стилю та функціональному призначенню нової кімнати. Це дозволило зосередитися на розробці специфічних об’єктів, які були унікальними для другого приміщення. Для створення таких моделей було підготовлено технічне завдання (ТЗ), яке детально описувало вимоги до геометрії, текстур, масштабу та функціональних властивостей кожного нового елемента.

Виконання ТЗ було передане іншому фахівцеві, який реалізував створення тривимірних моделей відповідно до встановлених критеріїв. Після завершення роботи над 3D-моделями до них було підключено текстури, що надавали об’єктам реалістичного вигляду.

Усі об’єкти були ретельно розташовані (рис. 3.15) в приміщенні відповідно до концептуального плану, створеного на основі whitebox-моделі.

Особливу увагу було приділено налаштуванню колізій для всіх тривимірних моделей у приміщенні. Колізії є ключовим елементом інтерактивності, оскільки визначають, як об’єкти взаємодіють із користувачем і між собою в межах віртуального середовища.

Для налаштування (рис. 3.14) використовувалися як автоматично згенеровані, так і вручну створені колізійні моделі, що забезпечило точність та оптимальну продуктивність сцени. Усі об’єкти, які вимагали фізичної взаємодії, отримали колізії, відповідні їхній геометрії, тоді як декоративні елементи мали спрощені форми колізій, щоб зменшити навантаження на обчислювальні ресурси.

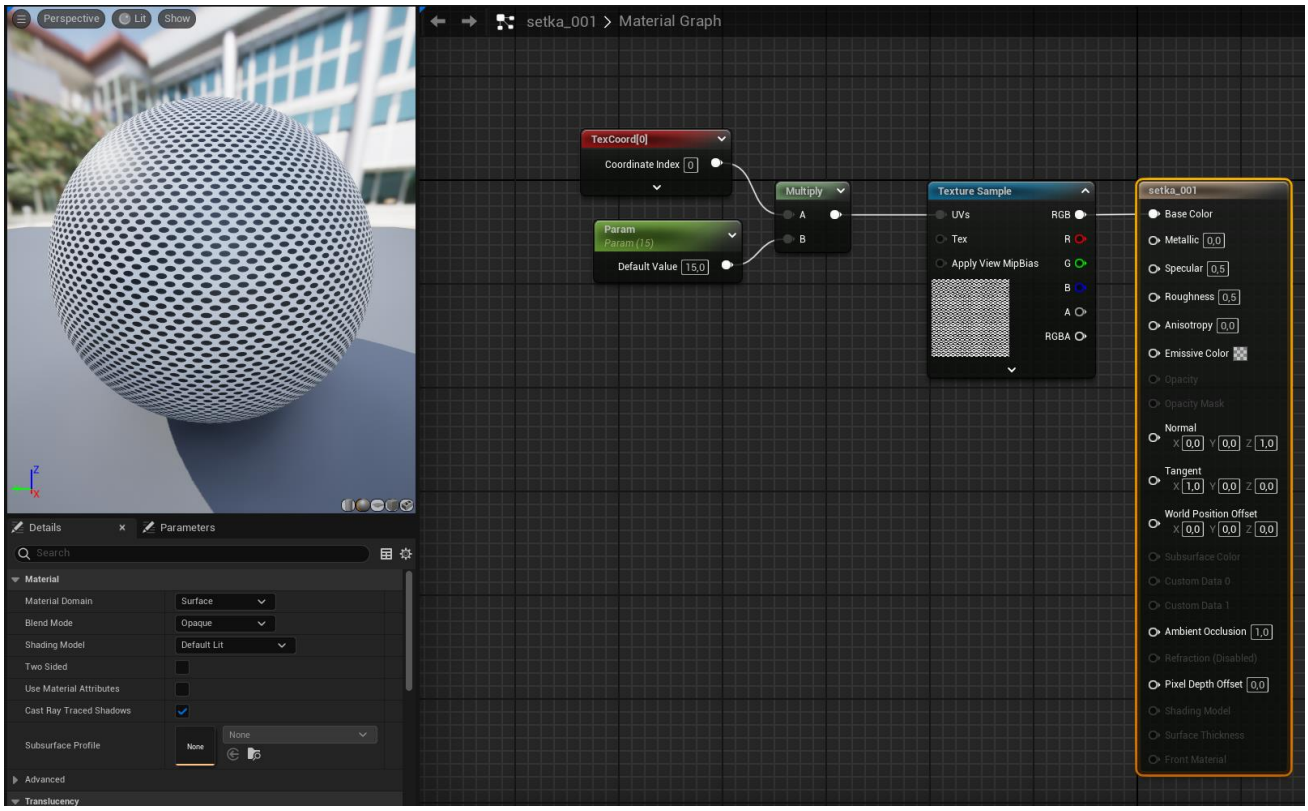

Рисунок 3.14 – Приклад підключення текстур до об'єкту

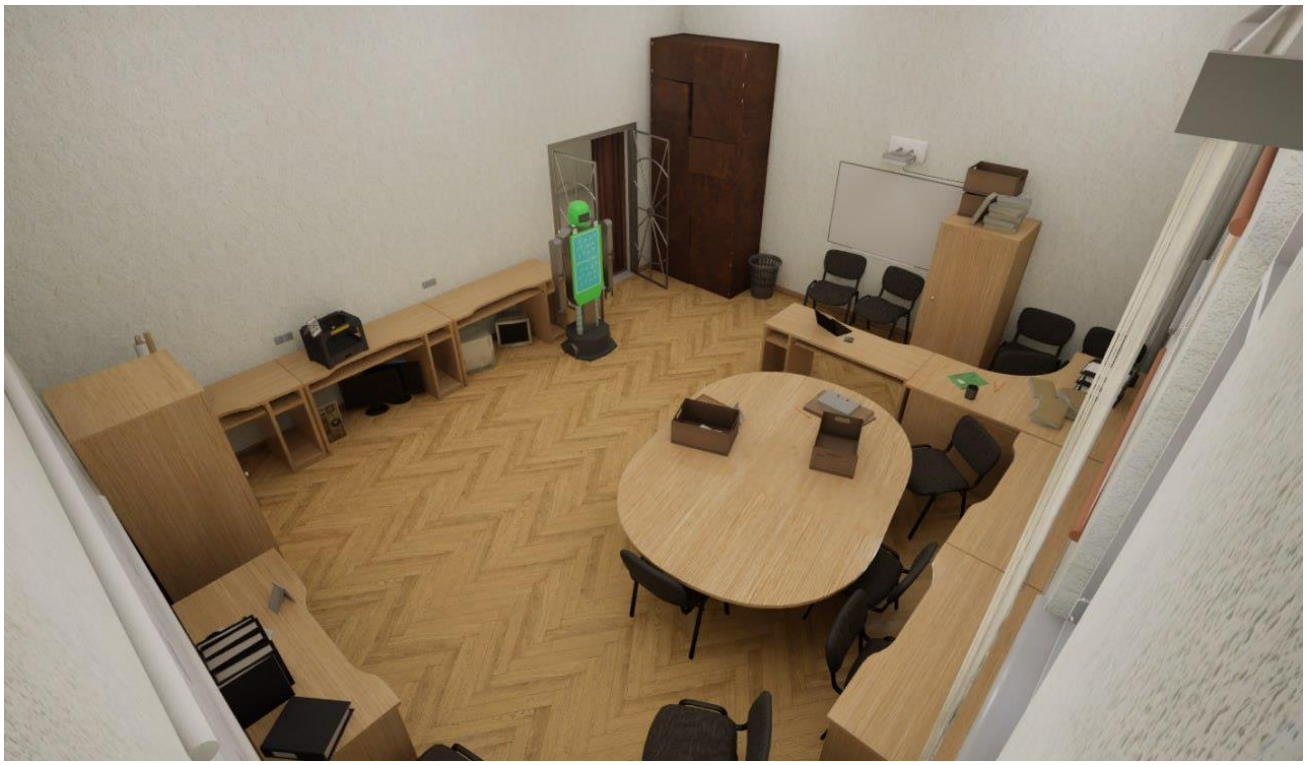

Рисунок 3.15 – Приклад приміщення

3.5 Реалізація робота

Для створення робота NPC, здатного виконувати комплекс завдань у віртуальному середовищі Unreal Engine 5, було реалізовано багатоступеневий підхід, який включав розробку його фізичної моделі, поведінки, інтерактивності та механізмів збору й передачі даних.

Основна мета проєкту полягала у створенні робота, який міг би за натисканням клавіш пересуватися між стендами, автоматично об'їжджаючи перешкоди, довертати голову у напрямку стенда, фотографувати QR-коди, передавати їх у систему розпізнавання OpenCV для декодування, озвучувати текстову інформацію та збирати статистику про виконані завдання. Ця статистика включала дані про кількість підходів до стендів, час, витрачений на досягнення цілей, кількість поворотів голови та інші параметри, які передавалися на сервер для збереження у базі даних.

Процес розробки розпочався зі створення основного каркасу робота у вигляді Blueprint Class [26]. До складу цього класу входили кілька ключових компонентів, які забезпечували функціональність та взаємодію робота з навколишнім середовищем. Одним із базових елементів був Capsule Component, що визначав фізичні властивості робота та забезпечував його взаємодію з іншими об'єктами. Цей компонент відповідав за колізію робота, що дало змогу уникати зіткнень із перешкодами та забезпечити коректну інтеракцію в середовищі. Іншим важливим елементом був Arrow Component, який виконував роль візуального покажчика напрямку руху та слугував орієнтиром для налаштування поведінки.

Для реалізації функції захоплення зображень, необхідних для розпізнавання QR-кодів, у Blueprint Class (рис. 3.16) був інтегрований Screen Capture Component 2D [27]. Цей компонент дозволяв зберігати зображення з віртуальної камери робота, які згодом передавалися в систему OpenCV для аналізу. Задля забезпечення гнучкості та зручності розробки всі компоненти

були ретельно налаштовані, а їхня взаємодія протестована в умовах моделювання різних сценаріїв поведінки робота.

Рисунок 3.16 – Основний Blueprint class робота

Mesh роботу був поміщений в Capsule Component для надання фізичних властивостей роботу.

Screen Capture Component 2D (рис. 3.17) була підключення до render target для запису зображення, в Capture source був обраний оптимальний формат, також були задані інші дрібні параметри.

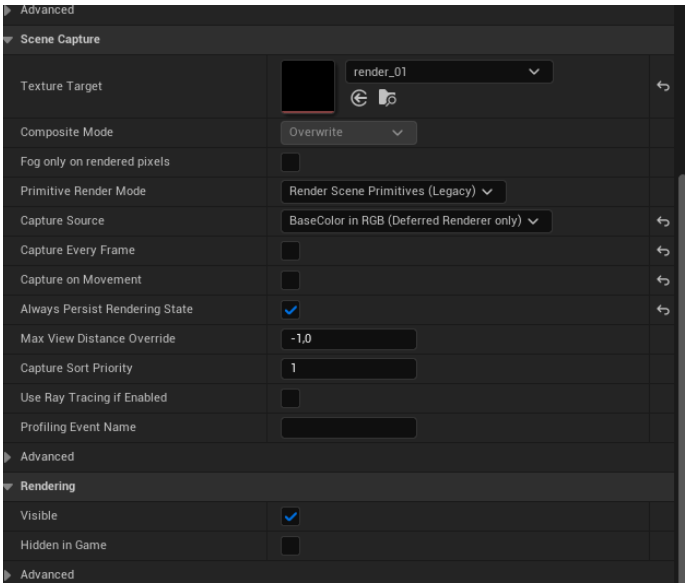

Рисунок 3.17 – Налаштування Screen Capture Component 2D

Наступним кроком було впровадження системи штучного інтелекту для управління поведінкою робота. Для цього був створений AI Component, що дозволив інтегрувати дерево поведінки (Behavior Tree), яке керувало діями робота в залежності від поставлених завдань. У структурі дерева поведінки були реалізовані послідовності та селектори, що забезпечували чітку логіку виконання завдань. Наприклад, для переміщення між стендами був створений таск, який використовував систему навігації (NavMesh) для побудови оптимального маршруту з урахуванням розташування перешкод.

NavMesh [28] (Navigation Mesh) у Unreal Engine 5 є технологією, що використовується для створення і управління шляхами навігації для штучного інтелекту (ШІ) у віртуальному середовищі (рис. 3.18). Це основний інструмент, який дозволяє реалізувати логіку переміщення персонажів у тривимірному просторі, враховуючи складність сцени, перешкоди, зміну рельєфу та інші аспекти.

Навігаційна сітка (NavMesh) є набором полігонів, які генеруються у межах простору сцени, де персонажі ШІ можуть пересуватися. Ця сітка автоматично визначає доступні зони для руху, ігноруючи недоступні області, такі як стіни, об'єкти чи інші фізичні бар'єри. Основна перевага NavMesh полягає в тому, що вона дозволяє ШІ обчислювати найефективніший шлях до заданої точки, зважаючи на всі обмеження середовища.

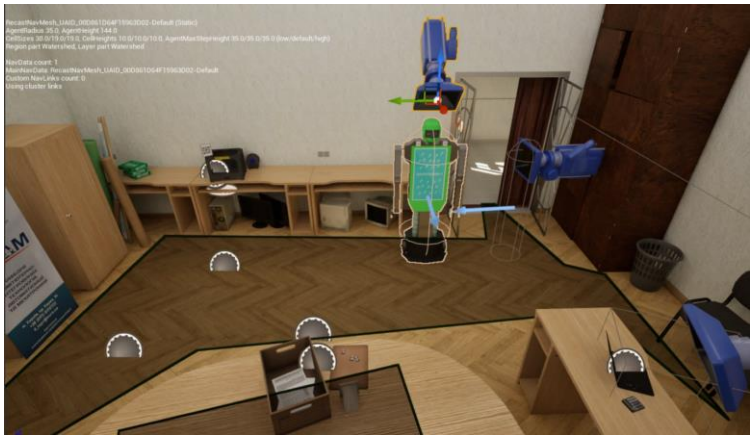

Рисунок 3.18 – Розташування NavMesh

Процес генерації NavMesh в Unreal Engine 5 починається з додавання до сцени об'єкта NavMeshBoundsVolume, який визначає область, у межах якої буде створена навігаційна сітка. Система автоматично аналізує геометрію сцени у межах цього об'єму і створює сітку на основі налаштувань, заданих розробником. Наприклад, такі параметри, як розмір кроку, висота кроку та мінімальний розмір площі для пересування, впливають на те, як буде побудована навігаційна сітка.

Unreal Engine 5 також надає можливість динамічного оновлення NavMesh у реальному часі. Це особливо корисно для сцен, у яких середовище змінюється під час гри, наприклад, у результаті руйнування об'єктів чи створення нових перешкод. Така функціональність забезпечується завдяки адаптивному оновленню сітки, яке дозволяє ШІ оперативно адаптувати свої маршрути до змін.

Ще однією важливою особливістю NavMesh є інтеграція з поведінковими деревами (Behavior Trees) (рис. 3.19) [29] і контролерами ШІ (AI Controllers). Це дозволяє розробникам задавати складну поведінку персонажів, наприклад, уникання перешкод, пошук оптимального маршруту чи динамічну зміну цілей. У поєднанні з системою запитів EQS (Environment Query System) NavMesh дає змогу реалізовувати складні сценарії, що враховують оточення і контекст ситуації.

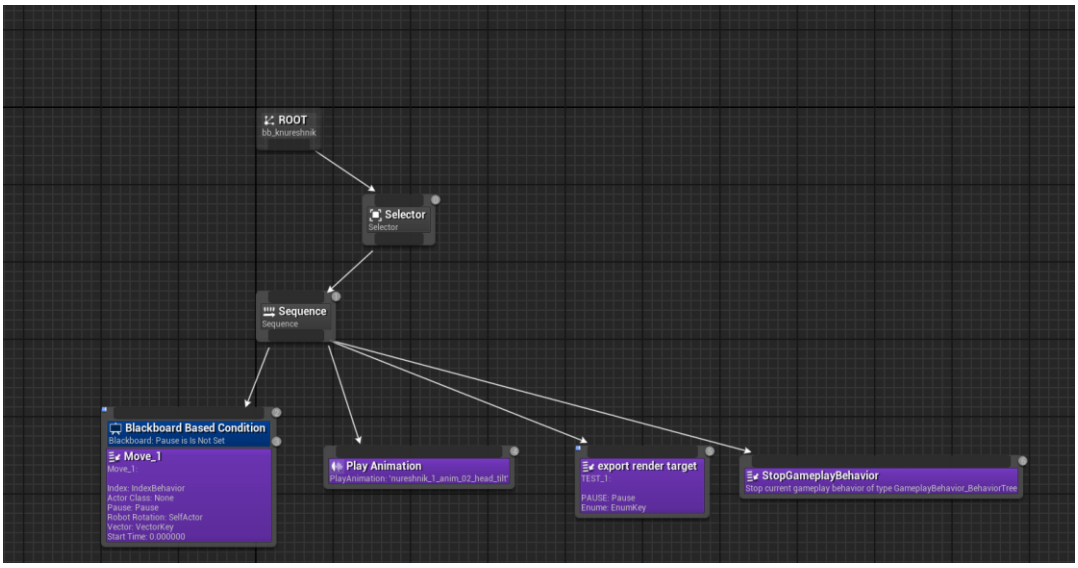

Рисунок 3.19 – Реалізація Behavior tree

Таск для довороту голови був налаштований на обчислення кута повороту у напрямку заданої точки, що дозволяло роботові точно орієнтуватися на стенд. Інший таск відповідав за запуск анімацій і успішного розпізнавання QR-коду.

Окрім цього, для організації роботи дерева поведінки (рис. 3.19) був створений Blackboard (рис. 3.20), що містив ключі для збереження проміжних даних, таких як поточна цільова точка, стан виконання завдання та напрямок руху. Використання чорної дошки дозволило зробити систему більш модульною та легкою для масштабування, оскільки кожен ключ міг бути використаний у кількох тасках, забезпечуючи обмін даними між ними.

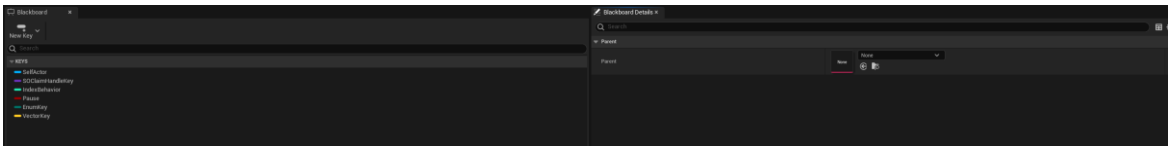

Рисунок 3.20 – Реалізація Blackboard

Для забезпечення інтерактивності робота було створено Widget Blueprint, який слугував інтерфейсом для передачі команд користувача. Цей віджет (рис. 3.21 – 3.22) був частиною головного інтерфейсу (Main Widget) і забезпечував інтуїтивно зрозумілий спосіб управління роботом. Користувач міг обрати цільовий стенд за допомогою кнопок, після чого інформація передавалася через інтерфейс (рис. 3.23) до робота, що дозволяло йому почати виконання.

Widget Blueprint був створений як частина окремого класу Blueprint, до якого було додано статичну сітку ноутбука. Для створення та налаштування самого віджета використовувався спеціалізований редактор, що є невід'ємною складовою Unreal Engine 5.

У цьому редакторі віджет було детально налаштовано для інтеграції з іншими елементами проєкту, включаючи адаптивність до змінних параметрів середовища. Завдяки широким можливостям редактора, вдалося реалізувати

оптимальну взаємодію між графічним інтерфейсом і функціональними елементами віртуального простору.

Рисунок 3.21 – Widget Blueprint для керування роботом

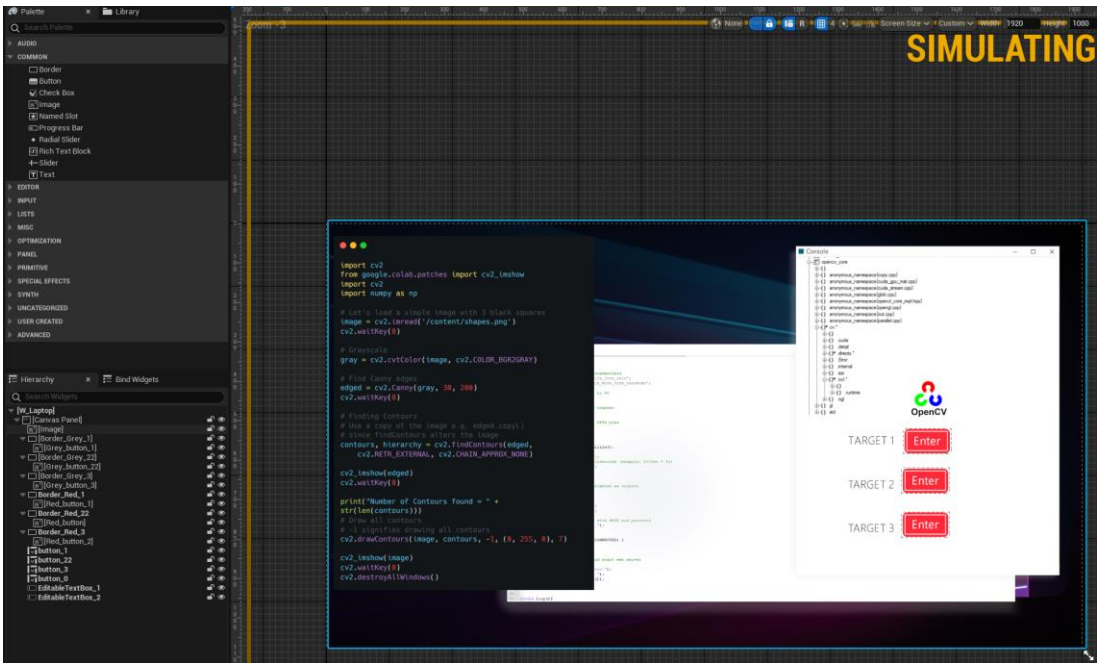

Рисунок 3.22 – Widget Blueprint для керування

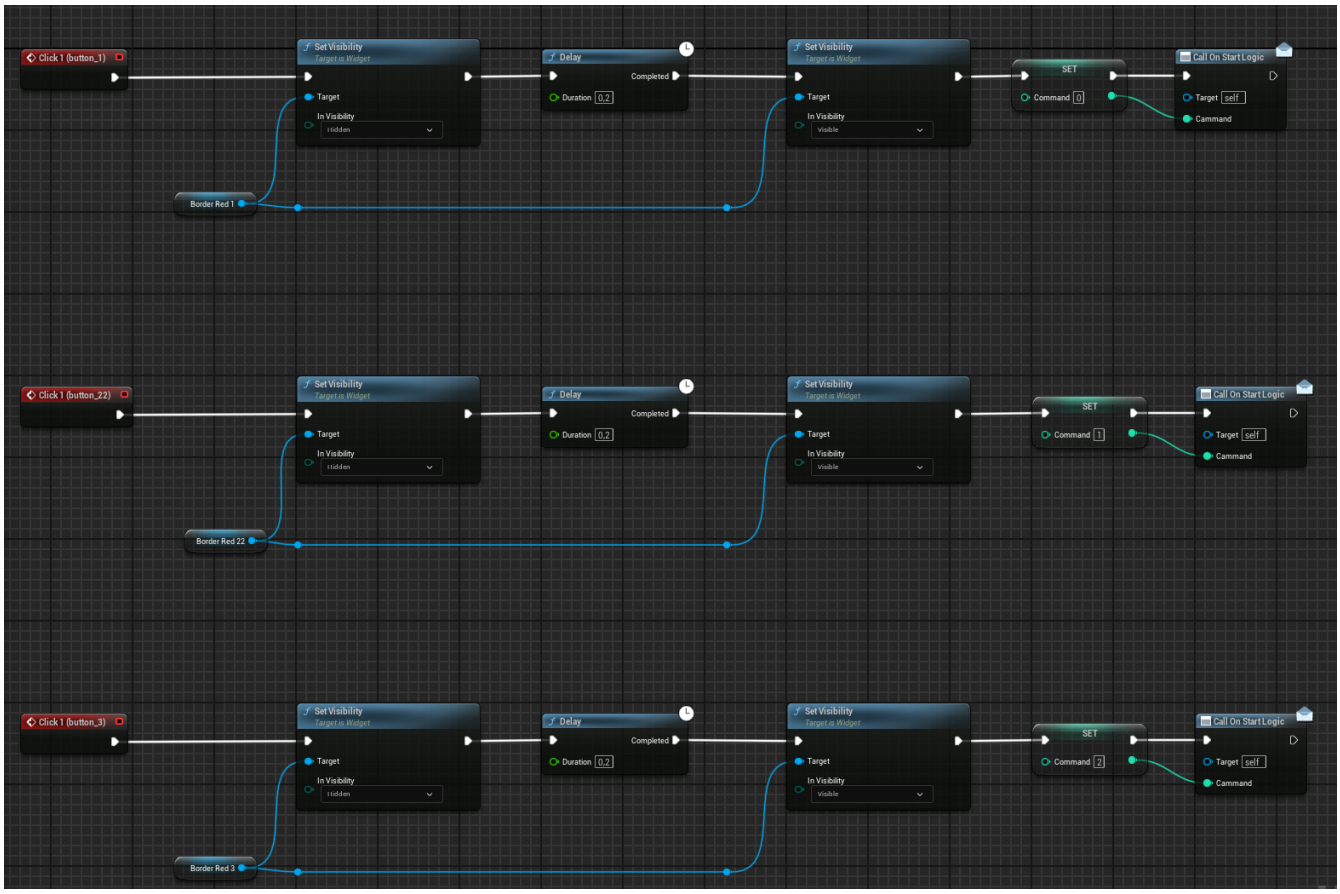

Рисунок 3.23 – Реалізація кнопок Widget Blueprint

Особливу увагу було приділено реалізації механізму сканування QR-кодів. Для цього було налаштовано збереження зображення через Render Target (рис. 3.24), який працював у зв’язці з Screen Capture Component 2D. Захоплене зображення передавалося до модуля OpenCV, інтегрованого в Unreal Engine, де здійснювалося його розпізнавання. Використання OpenCV дозволило з високою точністю аналізувати зображення, розпізнавати QR-коди та отримувати текстову інформацію.

Після декодування текст передавався у систему озвучення, де за допомогою технології Text-to-Speech синтезувався голос. Озвучений текст відтворювався через динаміки робота, що додавало інтерактивності та полегшувало сприйняття інформації.

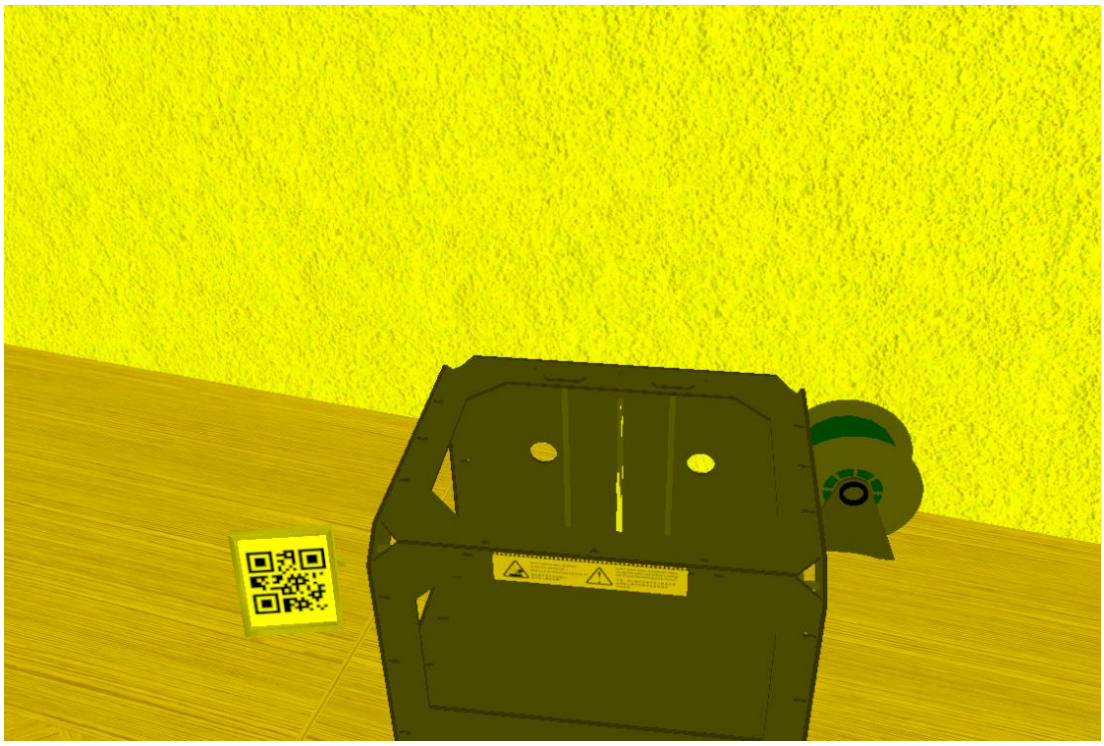

Рисунок 3.24 – Приклад записаної інформації у Render Target

Щоб забезпечити зберігання й аналіз даних про роботу NPC, була розроблена система збору статистики. У процесі виконання завдань робот збирав дані про кількість відвіданих стендів, час, витрачений на досягнення кожної точки, кількість поворотів голови та інші параметри (рис. 3.25). Ця інформація передавалася на сервер через HTTP-запити, де зберігалася у базі даних для подальшого аналізу. Завдяки цьому з’явилася можливість оцінювати ефективність роботи робота та адаптувати його поведінку до нових вимог.

У процесі розробки також були налаштовані колізії для всіх компонентів робота, що забезпечило його стабільну роботу в середовищі з перешкодами. Для цього використовувалися як автоматично згенеровані, так і вручну створені колізійні моделі, що дозволило оптимізувати продуктивність системи. Колізії налаштовувалися таким чином, щоб уникати непередбачуваних зіткнень і забезпечувати плавний об’їзд перешкод, що було досягнуто завдяки точному налаштуванню Capsule Component та використанню системи навігації.

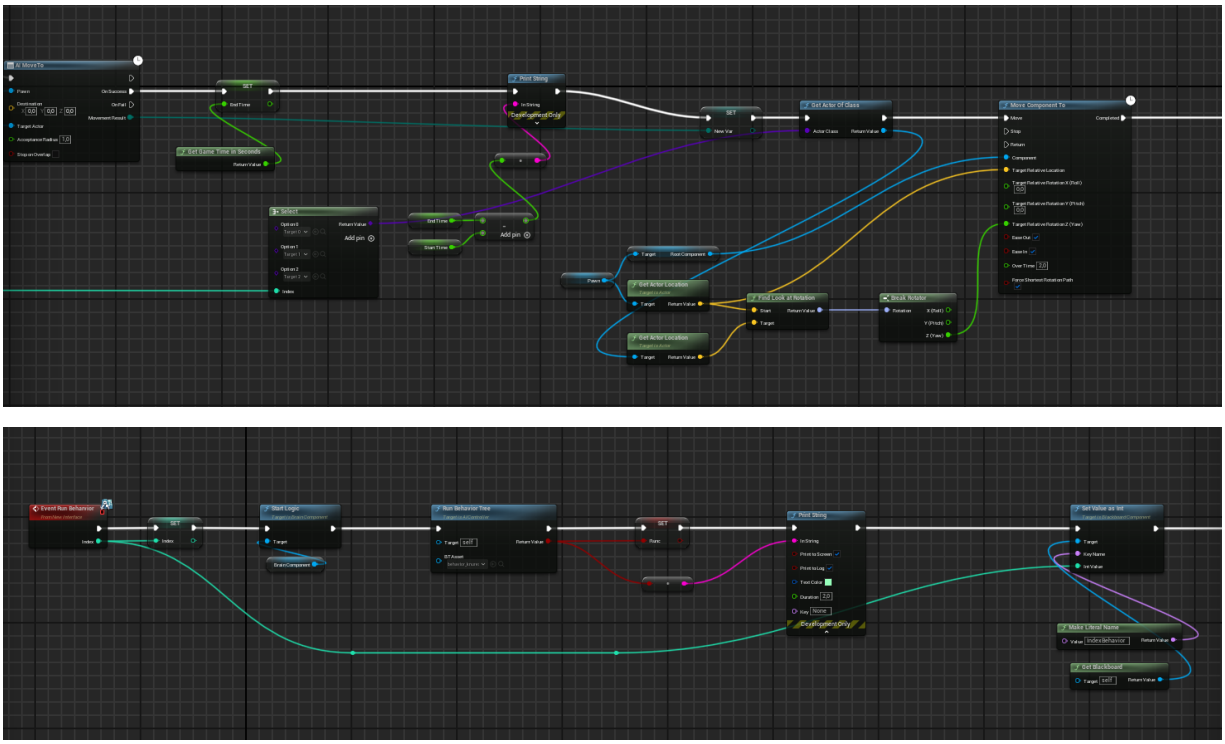

Рисунок 3.25 – Приклад нод, що забезпечують роботу тасків та AI component

3.6 Реалізація OpenCV

Окремо слід виділити створення програми для розпізнання qr-кодів на основі бібліотеки OpenCV та інтеграція функціоналу з Unreal Engine 5 за допомогою плагіна Read Local TXT.

У даній частині було розроблено застосунок для розпізнавання QR-кодів, який також забезпечує запис розпізнаного тексту у файл і подальшу інтеграцію отриманих даних в Unreal Engine 5. Програма для розпізнавання QR-кодів реалізована на мові програмування C++ з використанням потужної бібліотеки OpenCV, яка є стандартом для обробки зображень і комп'ютерного зору. Основний функціонал програми включає кілька етапів обробки зображень і взаємодії з користувачем.

На першому етапі програма завантажує зображення з вказаного шляху на локальному диску, що дозволяє обробляти різноманітні формати зображень, що

містять QR-коди. Завантажене зображення проходить через етап попередньої обробки, де воно перетворюється в градації сірого. Цей крок важливий, оскільки дозволяє зменшити кількість кольорової інформації, зосереджуючи увагу на контрасті, що покращує точність розпізнавання. Додатково проводиться вирівнювання гістограми, яке підвищує контраст між фоном і самим QR-кодом, що також сприяє покращенню точності детекції.

Для безпосереднього розпізнавання QR-кодів використовується клас QRCodeDetector з бібліотеки OpenCV. Цей клас забезпечує ефективне виявлення та декодування QR-кодів у зображенні, що дозволяє отримати текстові дані, закодовані в QR-коді. Якщо розпізнавання успішне, програма виводить розпізнану інформацію у консоль для зручності користувача. Додатково, на зображенні створюється індикатор успіху – зелений маркер, що вказує на те, що QR-код був успішно розпізнаний.

Крім цього, для забезпечення подальшого використання отриманих даних, вони записуються у текстовий файл result.txt. Це дозволяє зберігати розпізнаний текст для подальшої обробки або інтеграції з іншими системами, зокрема з Unreal Engine 5. У разі невдачі розпізнавання QR-коду, програма надає користувачу відповідне повідомлення про помилку, що дозволяє своєчасно виявляти та виправляти проблеми з якістю зображення або кодом.

Таким чином, програма забезпечує ефективне і точне розпізнавання QR-кодів з високим рівнем обробки зображень, що дозволяє інтегрувати дані у більш складні системи, зокрема віртуальні лабораторії або інтерактивні проєкти на базі Unreal Engine 5. У разі невдачі програма відображає червоний індикатор та повідомляє про відсутність даних.

Окрім цього, код передбачає інтерактивний вивід зображення з відповідним повідомленням для користувача. Завдяки використанню OpenCV забезпечується гнучкість в обробці зображень, а також висока продуктивність у виконанні алгоритмів розпізнавання.

Код програми:

```
#include <opencv2/objdetect.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <iostream>
#include <fstream>

using namespace cv;

int main(int argc, char* argv[]) {

    std::string folderPath = R"(D:\test\)";
    std::string fileName = "QR.png";
    std::string imagePath = folderPath + fileName;


    Mat getImage = imread(imagePath);

    if (getImage.empty()) {
        std::cerr << "Error: Unable to open image file: " << imagePath << std::endl;
        return -1;
    }


    Mat grayImage;
    cvtColor(getImage, grayImage, COLOR_BGR2GRAY);
    equalizeHist(grayImage, grayImage);


    QRCodeDetector qrDet;


    std::string data = qrDet.detectAndDecode(grayImage);


    Scalar indicatorColor;
    std::string statusMessage;

    if (!data.empty()) {
        std::cout << "Decoded URL: " << data << std::endl;
        indicatorColor = Scalar(0, 255, 0);
        statusMessage = "Link Found: " + data;


        std::ofstream outputFile(folderPath + "result.txt");
        if (outputFile.is_open()) {
            outputFile << data;
            outputFile.close();
            std::cout << "Data has been written to result.txt" << std::endl;
        }
        else {
            std::cerr << "Error: Unable to open result.txt for writing" << std::endl;
        }
    }
    else {
        std::cout << "QR Code not detected or failed to decode" << std::endl;
        indicatorColor = Scalar(0, 0, 255);
        statusMessage = "No Link Found";
    }
```

```
        putText(getImage, statusMessage, Point(10, 30), FONT_HERSHEY_SIMPLEX, 1,
indicatorColor, 2);


    imshow("QR Code Scanner", getImage);


    waitKey(1);
    destroyAllWindows();

    return 0;
}
```

Для інтеграції результатів у віртуальне середовище на Unreal Engine 5 було використано плагін Read Local TXT (рис. 3.26). Цей плагін дозволяє зчитувати текстові файли з локальної файлової системи безпосередньо з проєкту Unreal Engine, що є зручним рішенням для передачі результатів з інших додатків.

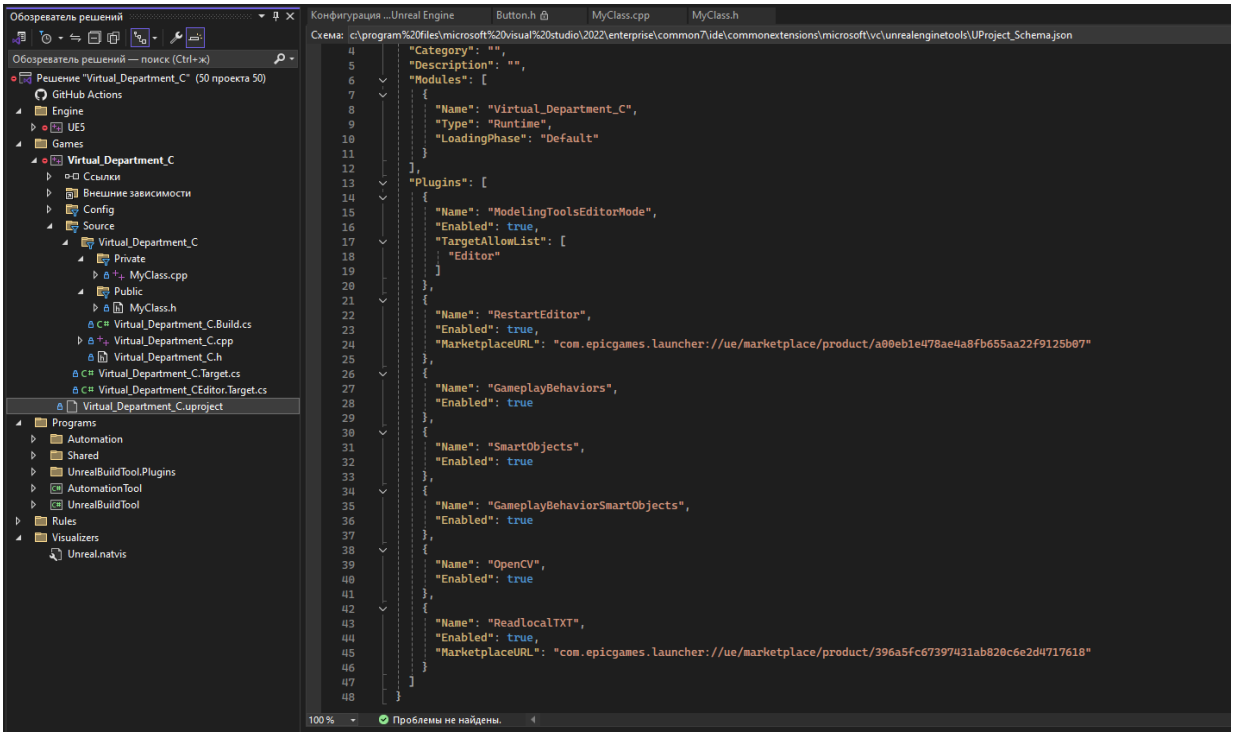

Рисунок 3.26 – Код підключення плагінів

Функціонал плагіна включає:

– читання вмісту текстового файлу та передача цього вмісту у змінні, які можуть бути використані у Blueprint або C++ класах. Це дозволяє легко

інтегрувати збережені текстові дані в логіку проєкту, наприклад, для завантаження параметрів конфігурації, сценаріїв чи текстових ресурсів.

– видалення текстового файлу після зчитування даних, що забезпечує безпеку та унеможливлює повторне використання старих результатів. Ця функція особливо важлива для збереження актуальності даних і запобігання потенційним помилкам через використання застарілої інформації.

– створення текстового файлу з вмістом, який може бути динамічно згенерований у грі або іншому проєкті. Це дозволяє зберігати користувацькі дані, наприклад, налаштування гри, лог-файли, результати тестування чи важливу службову інформацію. Процес створення файлів автоматизований, що забезпечує точність та уникнення конфліктів з іменами файлів, завдяки використанню унікальних ідентифікаторів або стандартних шляхів збереження.

– оцінювання вмісту текстового файлу для перевірки його коректності перед подальшою обробкою. Ця функція дозволяє переконатися, що файл відповідає очікуваним параметрам, наприклад, має правильну структуру або формат даних. Оцінювання дає змогу уникнути помилок у роботі системи, наприклад, якщо файл був пошкоджений або містить неправильні дані. Це важливий крок для забезпечення стабільності і передбачуваності роботи проєкту.

– запис даних у текстовий файл, що дозволяє зберігати результат роботи програми або дані, отримані під час її виконання. Завдяки цій функції можна записувати результати обчислень, прогрес користувача в грі, дані про стан системи чи іншу інформацію, яка може бути корисною для подальшого аналізу. Процес запису є оптимізованим для збереження великих обсягів даних і підтримує роботу з різними кодуваннями, що дозволяє працювати з файлами, сумісними з іншими системами чи програмами.

Налаштування плагіна в проєкті Unreal Engine 5, забезпечення доступу до локальних текстових файлів.

Результатом виконаної роботи стало створення системи, що автоматизує обробку QR-кодів із використанням OpenCV та забезпечує передачу отриманих даних у Unreal Engine для подальшого використання у віртуальній кафедрі.

3.7 Структура проєкту

Проєкт, створений у середовищі Unreal Engine 5, має мати структуровану і чітко визначену ієрархію папок [30], що забезпечує зручність у роботі, легкість навігації та ефективність командної розробки. Основна частина проекту зберігається у папці Content, яка є кореневим каталогом для всіх ресурсів та файлів проекту. Усередині цієї папки реалізовано три основні підрозділи: папка з назвою проекту, External та Dev.

Папка з назвою проекту слугує основним робочим простором, де розташовані всі ресурси та файли, безпосередньо пов’язані з функціоналом ігрового процесу. Ця папка поділена на кілька підпапок:

Character – у цій папці зберігаються всі ресурси, пов’язані з ігровими персонажами. Це анімації, моделі персонажів, скелети, текстури, а також весь код, що відповідає за логіку персонажів, включно з деревами поведінки (Behavior Trees) та штучним інтелектом (AI Controllers).

Sound – у папці Sound розміщено звукові ефекти, музичні файли та аудіо-ресурси, що використовуються у проекті. Вони класифіковані за типами: звуки оточення, дії персонажів та інші.

FX – ця папка містить спеціальні ефекти, такі як частинки, візуальні ефекти, матеріали для симуляції фізичних явищ та інші графічні елементи.

Blueprints – у Blueprint зберігаються всі сценарії, створені у візуальному редакторі Blueprints. Це логіка ігрового процесу, взаємодії між об’єктами, механіки тощо.

Levels – у папці Levels розташовані всі ігрові рівні, сцени та їхні конфігурації. Тут організовано зберігання карт для різних режимів гри.

Props – ця папка використовується для зберігання статичних і динамічних об’єктів сцени, таких як меблі, будівлі, предмети інтер’єру.

PostProcess – у PostProcess знаходяться налаштування постобробки, включаючи корекцію кольорів, ефекти розмиття, глибини різкості тощо.

UI – ця папка містить усі елементи користувацького інтерфейсу, включаючи віджети, іконки, шрифти та матеріали, що використовуються для відображення інформації на екрані.

Animations – у папці Animations зберігаються всі файли, пов’язані з анімаціями, включаючи секвенції, синхронізації рухів і контролери анімації.

Cinematics – у Cinematics знаходяться усі відеовставки, катсцени, анімовані переходи та файли, пов’язані з їхньою реалізацією.

Core – папка Core є однією з найважливіших у структурі проекту. Тут знаходяться файли, що відповідають за базову функціональність проекту. Це Gamemode (режими гри), система збереження прогресу (Save System), а також Ability System, яка забезпечує реалізацію та керування здібностями персонажів.

Окрім того, у проекті є папка External, яка використовується для зовнішніх ресурсів, таких як сторонні плагіни, бібліотеки або файли, що імпортуються для допоміжного використання. Ця папка дозволяє ізолювати зовнішні залежності від основних файлів проекту.

Папка Dev слугує місцем для зберігання тимчасових файлів, персональних розробок, тестових сцен або експериментальних функцій. Важливо зазначити, що папка Dev за замовчуванням не пересилається в систему контролю версій (наприклад, Git), оскільки її вміст зазвичай не є частиною фінального продукту. Для того щоб включити папку Dev у систему контролю версій, необхідно змінити файл .gitignore, додавши відповідні виключення для цієї папки.

У проекті також дотримується певна ієрархія для файлів, написаних на мові програмування C++. Вони організовані за функціональним призначенням:

SaveGameSystems – файли, що відповідають за систему збереження прогресу, включно з механіками автозбереження та відновлення стану гри.

Core – базові класи та системи, такі як основні компоненти ігрового процесу та допоміжні функції.

Player – класи, що реалізують функціональність ігрових персонажів, включаючи управління, взаємодію з об’єктами та здібності.

GameService – сервіси, які підтримують ігровий процес, включаючи мережеві функції, обробку подій і керування ресурсами.

Settings – налаштування складності гри, параметри графіки, аудіо, управління та інші конфігураційні файли.

Структурований підхід до організації файлів (рис. 3.27) у проекті дозволяє спростити підтримку, розширення та масштабування проекту, забезпечуючи при цьому високий рівень узгодженості та прозорості у командній роботі.

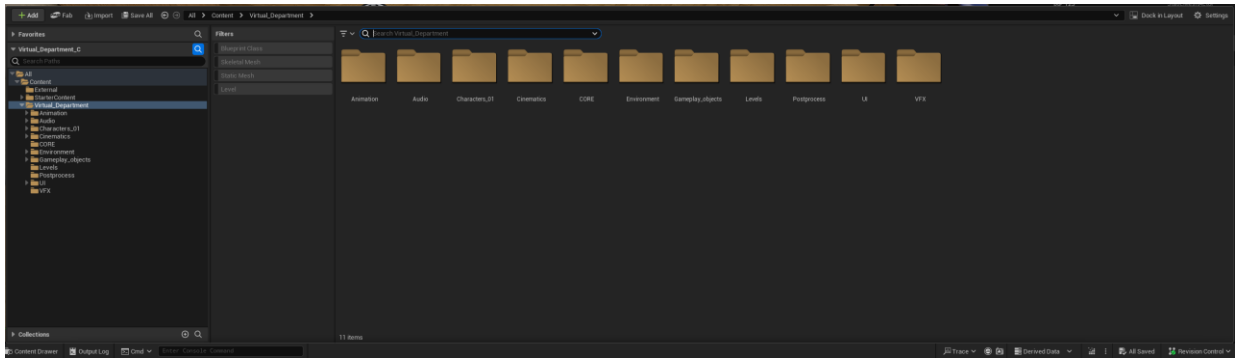

Рисунок 3.27 – Приклад організації папок в середовищі UE5

3.8 Тестування програмної частини

Після завершення всіх ключових етапів розробки та підготовки середовища для тестування, було проведено розміщення трьох експериментальних приладів у трьох різних місцях території лабораторії 162-2 кафедри КІТАР. Кожен прилад було оснащено відповідною ідентифікаційною табличкою, на якій містився QR-код. У QR-кодах було зашифровано детальний опис функціональних можливостей і технічних характеристик кожного приладу. Додатково, для ускладнення умов тестування, між першим і другим приладом була розміщена штучна перешкода, яка імітувала фізичний бар’єр, що вимагав від робота обхідного маневру.

Процес тестування проводився у кілька етапів. На кожному етапі від оператора вимагалося послідовно натискати на три кнопки, які відповідали

трьом встановленим приладам. Завдання робота полягало в тому, щоб правильно ідентифікувати натиснуту кнопку, знайти відповідний прилад у просторі, наблизитися до нього, виконати сканування таблички з QR-кодом та надати оператору коректну інформацію про прилад, зашифровану в коді. Особлива увага приділялася тому, як робот справляється із виявленням та обходом перешкоди, розміщеної на його шляху.

Під час тестування було також перевірено функціональність передачі даних між програмним забезпеченням робота та сервером. Це включало перевірку стабільності зв’язку, точності переданих даних, а також швидкості їх обробки. Зокрема, особливо важливим було переконатися, що всі результати сканування QR-кодів правильно відображаються у центральній базі даних, доступній для подальшого аналізу.

За результатами тестування було виявлено кілька критичних помилок у роботі системи. Однією з них стало некоректне розпізнавання QR-коду у складних умовах освітлення. Крім того, спостерігалися затримки у передачі даних на сервер. Всі виявлені недоліки були ретельно проаналізовані, і на їх основі було проведено доопрацювання системи. Зокрема, алгоритми обробки зображень були оптимізовані для роботи в умовах змінного освітлення, а також були оновлені механізми передачі даних, що дозволило суттєво зменшити затримки та підвищити стабільність системи в цілому.

Таким чином, проведені тести дали змогу не лише оцінити загальну ефективність розробленого програмного забезпечення, але й визначити його слабкі місця, що стало основою для подальшого покращення якості проєкту.

3.9 Розрахункова частина

Розрахунок пропускної здатності GPU для текстур дозволяє оцінити, чи зможе ваша відеокарта ефективно обробляти задану кількість текстур у сцені без втрати продуктивності (FPS) або інших проблем, таких як пропуски кадрів чи затримки. Текстури займають значну частину відеопам'яті (VRAM) GPU. Якщо

обсяг текстур перевищує доступний VRAM, GPU змушений використовувати оперативну пам’ять (RAM) для їх обробки, що значно уповільнює роботу та знижує FPS.

У роботі використовуються текстури, розподільна здатність яких, не перевищує 2048 х 2048 пікселів (2к). Розмір однієї текстури (в байтах) для формату ARGB (4 байти на піксель).

Середній розмір текстури (3.1):

$$T_c = S_x * S_Y * N_b \quad (3.1)$$

де S_x – кількість пікселів по горизонталі

S_Y – кількість пікселів по вертикалі

N_b – кількість байтів

$$T_c = 2048 * 2048 * 4 = 16{,}777{,}216 \text{ байт} \approx 17 \text{ МБ}$$

Отже, середній розмір текстури $\approx$ 17 МБ

Але у сцені присутня 21 текстура

Загальний обсяг текстур (3.2):

$$T_з = T_c * N_t \quad (3.2)$$

де

N_t– загальна кількість текстур

$$T_з = 17 * 21 = 357 \text{ МБ}$$

Обсяг текстур займає 357 МБ, що значно менше, ніж доступні 12 ГБ VRAM.

Розрахуємо загальний обсяг RAM, необхідний для моделей, текстур та фізики (3.3)

Враховуємо текстури і моделі. 3D-моделі займають 286 МБ, фізика і скрипти займають 742 МБ.

$$R_з = M_з * T_з * P_з \quad (3.3)$$

де $M_з$ – загальний обсяг пам'яті, яку займають моделі

$P_з$ – память яка видаляється на симуляцію фізики

$$R_з = 0.286 + 0.357 + 0.742 = 1.385 \text{ ГБ}$$

Отже, загальний обсяг RAM, необхідний для моделей, текстур та фізики $\approx$ 1.385 ГБ, що значно менше, ніж доступні 32 ГБ RAM.

Також необхідно враховувати вплив роздільної здатності монітора на продуктивність (FPS).

Роздільна здатність монітора безпосередньо впливає на продуктивність GPU, оскільки вона визначає кількість пікселів, які потрібно обробляти для кожного кадру. Більша роздільна здатність означає більше пікселів, що вимагає більше обчислювальної потужності графічного процесора.

Кількість пікселів на кадр визначається за (3.4):

$$N_з = S_ш * S_в \quad (3.4)$$

де $S_ш$ – кількість пікселів на моніторі по горизонталі

$S_в$– кількість пікселів на моніторі по вертикалі

Розрахунки для Full HD (1920 × 1080):

$N_з$= 1920 * 1080 = 2,073,600 пікселів (≈2 млн)

Розрахунки для 2K (2560 ∗ 1440):

$$N_з = 2560 * 1440 = 3{,}686{,}400 \text{ пікселів } (\approx 3.7 \text{ млн})$$

Розрахунки для 4K (3840 ∗ 2160):

$$N_з = 3840 * 2160 = 8{,}294{,}400 \text{ пікселів } (\approx 8.3 \text{ млн})$$

Продуктивність GPU в залежності від роздільної здатності монітора

FPS обернено пропорційний до кількості пікселів, які потрібно обробити. Для цього можна використати співвідношення продуктивності (3.5):

$$\text{FPS}_{\text{нова}} = \text{FPS}_{\text{базова}} \times \frac{N_{\text{пікселів базова}}}{N_{\text{пікселів нова}}} \tag{3.5}$$

При тестуванні на при розподільній здатності 2K отримали в середньому 90 FPS, приймаємо це значення за FPS базова

Розрахунки FPS для Full HD:

$$FPS_{fullHD} = 90 * \frac{3686400}{2073600} = 90 * 1.777 = 160 \text{ FPS}$$

Розрахунки FPS для 4K:

$$FPS_{4к} = 90 * \frac{3686400}{8294400} = 90 * 0.444 = 40 \text{ FPS}$$

Якщо базова продуктивність у 90 FPS була досягнута при роздільній здатності 2K (2560 × 1440), то можна очікувати, що для Full HD (1920 × 1080)

FPS значно зросте, оскільки кількість пікселів у цій роздільній здатності приблизно в 1.777 раза менша. У цьому випадку продуктивність становитиме близько 160 FPS. З іншого боку, при переході до роздільної здатності 4K (3840 × 2160), кількість пікселів зростає в 2.25 раза порівняно з 2K, що призводить до суттєвого зниження FPS до приблизно 40. Це пов’язано з тим, що графічний процесор повинен обробляти більше пікселів у кожному кадрі, що значно збільшує обчислювальне навантаження.

3.10 Подальші перспективи

Після створення та успішного налагодження інструменту віртуального тестування відкриваються перспективи для ускладнення задач, збору даних та проведення відповідних аналітичних розрахунків.

Такий підхід дозволить ефективно підготувати кафедру та самого робота до виконання лабораторної роботи у реальних умовах, що сприятиме суттєвій економії ресурсів та часу.

Зібрана статистика, яка зберігається на сервері, може стати цінним джерелом інформації для подальшого аналізу. Зокрема, ці дані допоможуть визначити оптимальний набір модулів, які будуть використовуватися для роботи робота, оцінити можливі обмеження, а також розкрити потенційні шляхи вдосконалення його функціоналу.

Окрім цього, зібрані аналітичні дані можуть слугувати основою для розширення сфери застосування робота. Наприклад, накопичена статистика дозволить адаптувати його до виконання інших лабораторних робіт, підвищивши тим самим універсальність та ефективність використання технології у навчальному процесі.

Усе це сприятиме вдосконаленню як технічного оснащення кафедри, так і загальної якості освітнього процесу, інтегруючи сучасні інноваційні технології у практичну підготовку студентів.

3.11 Охорона праці

Однією з фундаментальних людських потреб є відчуття безпеки. Забезпечення охорони здоров’я, збереження життя, захист від різноманітних ризиків і створення комфортних умов для існування є ключовими складовими цієї потреби [31]. Діяльність людини є особливою формою активної взаємодії зі світом, що дозволяє впливати на нього, змінювати його та адаптувати до власних потреб. Вона є одночасно метою, засобом, процесом і результатом людської активності.

Таким чином, безпека життєдіяльності (БЖД) полягає у створенні безпечних і комфортних умов взаємодії людини з навколишнім середовищем, що враховує природні, техногенні й соціально-політичні фактори.

Потрібно слідкувати за поставою: сидіти прямо, тримаючи спину рівною, плечі розслабленими, а ноги розташованими на підлозі або на спеціальній підставці. Протягом тривалої роботи робіть перерви, під час яких потрібно вставати, виконувати легкі фізичні вправи для покращення кровообігу та зняття м’язової напруги.

Забезпечте належне освітлення робочого місця. Уникайте яскравих відблисків на екрані, розташовуйте його під кутом до джерела світла або використовуйте жалюзі чи штори.

Відрегулюйте яскравість і контрастність монітора, щоб мінімізувати навантаження на очі. Також оптимізуйте розмір шрифту для зручності читання.

За можливості використовуйте природне світло, розміщуючи робоче місце поруч із вікном. Якщо природного світла недостатньо, використовуйте додаткові світильники з м’яким і рівномірним освітленням. Застосовуйте лампи з можливістю регулювання яскравості, щоб налаштувати освітлення під свої потреби.

3.12 Висновок до третього розділу

У третьому розділі було здійснено розробку програмної частини віртуального середовища, що включало створення інтерактивної віртуальної лабораторії з використанням Unreal Engine 5. Проведено збір вихідних даних, визначено технічні параметри середовища та розроблено структуру програмного забезпечення, що забезпечує високу продуктивність і гнучкість у реалізації навчальних сценаріїв.

Було створено деталізовані 3D-моделі лабораторного обладнання з використанням Blender та Maya, налаштовано колізії для коректної взаємодії з об’єктами та інтегровано фізично коректні матеріали та текстури, що значно підвищило реалістичність віртуального простору. Використання технологій Nanite та Lumen у Unreal Engine 5 дозволило досягти високої деталізації моделей і реалістичного освітлення, зберігаючи стабільну продуктивність навіть у складних сценах.

Для підвищення рівня інтерактивності було реалізовано логіку роботи моделі робота, що виконує певні навчальні завдання, а також розроблено систему розпізнавання QR-кодів на базі OpenCV, яка дозволяє студентам отримувати миттєвий доступ до додаткових матеріалів або інструкцій у віртуальному середовищі. Додатково впроваджено механізм збору інформації та статистики, що дозволяє аналізувати взаємодію користувачів із системою, відстежувати їхню активність та рівень успішності, а також коригувати навчальні сценарії відповідно до отриманих даних.

Важливим етапом стало тестування програмної частини лабораторії, під час якого було оцінено продуктивність середовища, стабільність його роботи та відповідність навчальним цілям. Здійснено аналіз можливих оптимізацій, зокрема покращення інтерактивності елементів, доопрацювання алгоритмів штучного інтелекту для персоналізації навчального досвіду та підвищення ефективності візуальних обчислень.

ВИСНОВКИ

Під час виконання проєкту було вивчено методи створення віртуальних лабораторій, їх роль у сучасній освіті та потенціал для інтеграції технологій штучного інтелекту. Здійснено аналіз доступних програмних платформ, таких як Unreal Engine 5, для розробки інтерактивних симуляцій та навчальних середовищ. Було виконано наступні завдання:

– розроблено концепцію інтерактивної віртуальної лабораторії, що підтримує 3D-моделювання, штучний інтелект та бази даних.;

– розроблено алгоритм дій для створення віртуальної лабораторії, включаючи етапи проєктування, програмування, тестування;

– визначено вимоги до платформи, а саме використання Unreal Engine 5 з C++ для створення десктопного застосунку;

– створено прототип віртуальної лабораторії;

– розроблено логіку роботи віртуальної моделі робота;

– розроблено програму на базі Open CVдля сканування qr-кодів;

– проведено тестування функціональності лабораторії та вдосконалення системи;

Розроблено інтерактивну віртуальну лабораторію з інтеграцією баз даних і штучного інтелекту для підтримки навчального процесу. Розробка віртуальної лабораторії є важливим кроком у модернізації освіти, що сприяє доступу до передових технологій та інтерактивного навчання для студентів і дослідників в умовах обмежених ресурсів, а також може стати зручним інструментом для планування складних лабораторних робіт, перед їх виконанням у реальному житті, та витратами грошових ресурсів на їх реалізацію.

Дана робота тісно пов’язана з досягненням Цілі сталого розвитку ООН «Якісна освіта». Розробка інтерактивної віртуальної лабораторії за допомогою OpenCV та Unreal Engine 5 сприяє впровадженню інноваційних освітніх технологій, забезпечуючи доступ до них для студентів і викладачів незалежно

від місця проживання чи фінансових можливостей. OpenCV дозволяє здійснювати аналіз зображень і відео, що надає студентам можливість працювати з реальними даними в контексті комп'ютерного зору, а Unreal Engine 5 забезпечує створення високоякісних візуальних середовищ для віртуальних лабораторій, надаючи можливість моделювати складні фізичні процеси та інтерактивні навчальні сценарії.

Завдяки віртуальним лабораторіям учасники навчального процесу отримують можливість здобувати якісні знання, вдосконалювати практичні навички та проводити наукові дослідження у середовищі, максимально наближеному до реальних умов.

Проєкт сприяє скороченню нерівності в доступі до освітніх ресурсів і забезпечує рівні можливості для всіх у здобутті освіти. Віртуальне середовище також дозволяє значно зменшити витрати на обладнання та матеріали, що є важливим для закладів освіти з обмеженим фінансуванням.

Інтеграція технологій, таких як OpenCV для обробки зображень, Unreal Engine 5 для створення інтерактивних 3D-середовищ та розробки реалістичних візуалізацій, дозволяє створити персоналізоване навчальне середовище, яке адаптується до індивідуальних потреб і темпів навчання кожного студента. Це сприяє формуванню сучасної, гнучкої, інтерактивної та доступної освітньої системи, яка відповідає викликам глобалізації та цифровізації.

Реалізація таких проєктів також стимулює розвиток цифрової грамотності, критичного мислення та інноваційного підходу до вирішення завдань, що є ключовими елементами 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 Филипенко, Олександр Цимбал. Харків: ХНУРЕ, 2023. 56 с.

3. Аналіз можливостей впровадження AI та 3D-технологій у різні галузі навчання / М.В. Скляров, 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. Unreal Engine 5.5 Documentation [Електронний ресурс]: Building Virtual Worlds – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/building-virtual-worlds-in-unreal-engine

5. Microsoft Ignite [Електронний ресурс]: Установка Инструменты Visual Studio для Unreal Engine – Режим доступу: https://learn.microsoft.com/en-en/visualstudio/gamedev/unreal/get-started/vs-tools-unreal-install

6. Research Center for Advanced Science and Technology The University of Tokyo [Електронний ресурс]: Inclusive design laboratory Режим доступу: https://www.rcast.u-tokyo.ac.jp/en/research/namiki_lab.html

7. ROQED [Електронний ресурс]: Learn real experimental physics – Режим доступу: https://roqed.com/product-physics/

8. Чорноморський національний університет імені Петра Могили [Електронний ресурс]: НАУКА ОНЛАЙН: ОНЛАЙН-ЛАБОРАТОРІЇ З ХІМІЇ ТА ФІЗИКИ – Режим доступу: https://chmnu.edu.ua/nauka-onlajn-onlajn-laboratoriyi-z-himiyi-ta-fiziki/

9. Ditc-contact [Електронний ресурс]: Роботизована техніка та промислова електроніка – Режим доступу: https://ditc-contact.ua/product-category/robotizovana-texnika/

10. Unreal Engine [Електронний ресурс]: Unreal Engine 5 – Режим доступу: https://www.unrealengine.com/en-US/unreal-engine-5

11. Unreal Engine [Електронний ресурс]: Unreal Engine 5.5 – Режим доступу: https://www.unrealengine.com/en-US/blog/unreal-engine-5-5-is-now-available

12. Unreal Engine [Електронний ресурс]: Lumen Global Illumination and Reflections – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine

13. Opencv [Електронний ресурс]: Opencv – Режим доступу: https://opencv.org

14. Github [Електронний ресурс]: yolov5 – Режим доступу: https://github.com/ultralytics/yolov5

15. Git [Електронний ресурс]: Git community – Режим доступу: https://git-scm.com

16. Ipvid [Електронний ресурс]: Використання QR-кодів у сучасних підручниках – Режим доступу: https://ipvid.org.ua/index.php/psp/article/view/189/190

17. Zbar [Електронний ресурс]: ZBar bar code reader – Режим доступу: https://zbar.sourceforge.net

18. Zxing [Електронний ресурс]: ZXing Decoder Online – Режим доступу: https://zxing.org/w/decode.jspx

19. Google [Електронний ресурс]: API-інтерфейси візуалізації – Режим доступу: https://developers.google.com/ml-kit?hl=en

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

21. Microsoft [Електронний ресурс]: Спільна побудова майбутнього розробки програмного забезпечення – Режим доступу: https://visualstudio.microsoft.com/

22. Git [Електронний ресурс]: Large File Storage (LFS) – Режим доступу: https://git-lfs.com

23. Narakeet [Електронний ресурс]: Текст в голос онлайн українською – Режим доступу: https://www.narakeet.com/languages/ukrainian-text-to-speech-ua/

24. Fab [Електронний ресурс]: Restart Unreal Editor – Режим доступу: https://www.fab.com/listings/2e260da0-4a02-4f7b-8b97-3f783dc025f1

25. Fab [Електронний ресурс]: ReadlocalTXT – Режим доступу: https://www.fab.com/listings/d3dd43fa-99a9-4ce8-9242-78023113bec5

26. Epicgames [Електронний ресурс]: Blueprint Class – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprint-class-assets-in-unreal-engine

27. Epicgames [Електронний ресурс]: Scene Capture 2D – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/1.7---scene-capture-2d?application_version=4.27

28. Epicgames [Електронний ресурс]: Modifying the Navigation Mesh – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/overview-of-how-to-modify-the-navigation-mesh-in-unreal-engine

29. Epicgames [Електронний ресурс]: Behavior Tree Quick Start Guide – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/behavior-tree-in-unreal-engine---quick-start-guide

30. Git [Електронний ресурс]: UE4 Style Guide() – Режим доступу: https://github.com/Allar/ue5-style-guide?tab=readme-ov-file

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