

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Таврійський державний агротехнологічний університет**  
**імені Дмитра Моторного**  
**Факультет енергетики і комп'ютерних технологій**

ЗАТВЕРДЖУЮ

Зав. каф. "Комп'ютерні науки"

доц. \_\_\_\_\_ Сергій ШАРОВ  
"17" лютого 2025 р.

**Пояснювальна записка**

до кваліфікаційної роботи здобувача СВО Магістр  
(ступінь вищої освіти)

на тему: «Інтерактивне віртуальне середовище для вдосконалення ІТ-компетентностей здобувачів вищої освіти за допомогою іммерсивних технологій»

**52/4КНД.11260549.000000ПЗ**

Виконав: здобувач вищої освіти 2 курсу, групи 21МБКН  
спеціальності 122 Комп'ютерні науки  
за ОПП Комп'ютерні науки

(шифр і назва спеціальності та ОПП)

\_\_\_\_\_ Олександр МЕЛЕШКО

(підпис)

Керівник доц. \_\_\_\_\_ Дмитро ЛУБКО

(підпис)

Нормоконтроль, ст.викл. \_\_\_\_\_ Ольга ЗІНОВ'ЄВА

(підпис)

Рецензент, ст.викл. \_\_\_\_\_ Юрій НОВІКОВ

Запоріжжя – 2025 рік

# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Таврійський державний агротехнологічний університет імені Дмитра  
Моторного

Факультет: Енергетики і комп'ютерних технологій

Кафедра: Комп'ютерні науки

Ступінь вищої освіти: Магістр

Спеціальність: 122 Комп'ютерні науки

ОПП: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри\_КН

к.пед.н., доц. \_\_\_\_\_ Сергій ШАРОВ

“04” \_жовтня\_ 2024\_ року

## З А В Д А Н Н Я НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) ЗДОБУВАЧУ ВО

МЕЛЕШКО ОЛЕКСАНДР ДМИТРОВИЧ

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи): Інтерактивне віртуальне середовище для вдосконалення ІТ-компетентностей здобувачів вищої освіти за допомогою іммерсивних технологій

керівник проєкту (роботи) Лубко Дмитро Вікторович

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

затверджені наказом університету від “23” вересня\_ 2024\_ року №\_ 442-С\_

2. Строк подання здобувачем ВО проєкту (роботи) 14 лютого 2025

3. Вихідні дані до роботи: аналіз існуючих інтерактивно віртуальних систем та поставлене технічне завдання до виконання даного дипломного проєкту

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1. Характеристика, аналіз та опис предметної області та аналіз існуючих рішень.

2. Аналіз та оцінка ефективності використання VR-технологій у навчанні ІТ-навичкам.

3. Вибір та обґрунтування технологій розробки інтерактивного віртуального середовища.

4. Проектування, дослідна експлуатація та опис тестування розробленого інтерактивного віртуального середовища

5. Перелік графічного матеріалу

Презентація – 10 слайдів

1. Титульний слайд

2. Предмет, об'єкт, мета, завдання
3. Порівняльна характеристика аналогів
4. Діаграма варіантів використання
5. Схема взаємодії модулів (блоків) розробленої системи
6. Схема роботи розробленої VR-системи
7. Інструментальні засоби розробки
8. Проектування інтерактивного віртуального середовища
9. Проектування інтерактивного віртуального середовища
10. Проектування інтерактивного віртуального середовища
11. Тестування VR-системи
12. Висновки

#### 6. Консультанти розділів проєкту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|-------------------------------------------|----------------|------------------|
|        |                                           | завдання видав | завдання прийняв |
|        |                                           |                |                  |

7. Дата видачі завдання 01 жовтня 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                    | Строк виконання етапів проєкту (роботи) | Примітка |
|-------|-------------------------------------------------------------|-----------------------------------------|----------|
| 1     | Збір вихідних даних для виконання дипломного проєкту        | 10.09.2024                              | Виконано |
| 2     | Аналіз предметної області та постановка задачі              | 10.10.2024                              | Виконано |
| 3     | Проектування програмної системи                             | 23.10.2024                              | Виконано |
| 4     | Програмна реалізація інтерактивного віртуального середовища | 02.11.2024                              | Виконано |
| 5     | Аналіз досвідної експлуатації                               | 25.11.2024                              | Виконано |
| 6     | Тестування розробленої системи                              | 27.12.2024                              | Виконано |
| 7     | Оформлення проєкту в цілому                                 | 20.01.2025                              | Виконано |
| 8     | Підготовка до попереднього захисту дипломної роботи         | 03.02.2025                              | Виконано |
| 9     | Підпис керівником роботи                                    | 14.02.2025                              | Виконано |
| 10    | Підпис завідувачем кафедри                                  | 17.02.2025                              | Виконано |
| 11    | Перевірка роботи на плагіат                                 | 18.02.2025                              | Виконано |

Здобувач вищої освіти

( підпис )

Олександр МЕЛЕШКО

(власне ім'я та ПРИЗВИЩЕ)

Керівник проєкту (роботи)

( підпис )

Дмитро ЛУБКО

(власне ім'я та ПРИЗВИЩЕ)

## РЕФЕРАТ

Кваліфікаційна робота магістра: 101 с., 32 рис., 0 табл., 26 посилань, 3 додатки.

### Актуальність

Однією з головних проблем є обмежений доступ до фізичних компонентів ПК через їхню вартість і ризик пошкодження. Це створює потребу у використанні віртуальної реальності (VR), яка забезпечує інтерактивне навчальне середовище без фізичних обмежень.

VR дає змогу студентам детально вивчати будову ПК, моделювати складання та тестування без ризику пошкоджень. Дослідження підтверджують, що VR сприяє кращому засвоєнню матеріалу завдяки інтерактивності та високій залученості студентів.

Основні переваги VR: скорочення витрат на обладнання; гнучкість у навчанні, незалежність від фізичних лабораторій; можливість моделювання несправностей та аварійних ситуацій.

Застосування VR у провідних університетах (MIT, Стенфорд) підтверджує ефективність цього підходу для технічної освіти.

Об'єкт дослідження – процес навчання складанню ПК у VR. Предмет дослідження – розробка VR-середовища для навчання IT-спеціалістів та оцінка його ефективності.

Мета дослідження – створення інтерактивного VR-середовища для навчання складанню ПК.

Завдання дослідження: аналіз існуючих VR-рішень у навчанні; розробка концепції VR-середовища для навчання складанню ПК; створення та тестування VR-програми; оцінка ефективності VR у навчальному процесі.

### Методи дослідження:

1. Аналіз літератури – вивчення наукових джерел щодо VR у навчанні.
2. Розробка ПЗ – використання Unity для створення VR-середовища.
3. Експериментальний метод – тестування VR-системи студентами.

4. Порівняльний метод – оцінка ефективності VR у порівнянні з традиційним навчанням.

Практичне значення. Розроблене VR-середовище дозволяє студентам здобувати ІТ-компетенції у безпечному інтерактивному форматі без потреби у фізичних лабораторіях.

Апробація результатів. Результати дослідження опубліковані у матеріалах Всеукраїнської науково-практичної конференції «Сучасні комп'ютерні та інформаційні системи та технології» (Запоріжжя, 2024).

Ключові слова: ВІРТУАЛЬНА РЕАЛЬНІСТЬ (VR), ІНТЕРАКТИВНЕ НАВЧАННЯ, ІММЕРСИВНІ ТЕХНОЛОГІЇ, META QUEST, VR-СИМУЛЯЦІЯ, ДИСТАНЦІЙНЕ НАВЧАННЯ У VR.

## SUMMARY

Master's qualification work: 101 pages, 32 figures, 0 tables, 26 references, 3 appendices.

Relevance. One of the main challenges is the limited access to physical PC components due to their cost and the risk of damage. This creates a demand for the use of virtual reality (VR), which provides an interactive learning environment without physical constraints.

Use of VR in Education. VR enables students to study PC components in detail, simulate assembly and testing without the risk of damage. Research confirms that VR enhances material retention through interactivity and high student engagement.

Key benefits of VR: reduction in equipment costs; flexibility in learning, independent of physical laboratories; ability to simulate malfunctions and emergency situations.

The implementation of VR in leading universities (MIT, Stanford) confirms the effectiveness of this approach in technical education.

Object and Subject of Research. The object of research is the learning process of PC assembly in VR.

The subject of research is the development of a VR environment for training IT specialists and evaluating its effectiveness.

#### Research Goals and Objectives

Goal – to create an interactive VR environment for learning PC assembly.

Research objectives:

Analyze existing VR solutions in education.

Develop a concept for a VR environment for PC assembly training.

Create and test a VR application.

Evaluate the effectiveness of VR in the learning process.

Research Methods:

Literature Analysis – studying scientific sources on VR in education.

Software Development – using Unity to create the VR environment.

Experimental Method – testing the VR system with students.

Comparative Method – evaluating VR effectiveness compared to traditional learning.

#### Practical Significance

The developed VR environment allows students to acquire IT competencies in a safe, interactive format without the need for physical laboratories.

#### Testing of Results

The research findings have been published in the proceedings of the All-Ukrainian Scientific and Practical Conference "Modern Computer and Information Systems and Technologies" (Zaporizhzhia, 2024).

Keywords: VIRTUAL REALITY (VR), INTERACTIVE LEARNING, IMMERSIVE TECHNOLOGIES, META QUEST, VR SIMULATION, REMOTE LEARNING IN VR.

## ЗМІСТ

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| ВСТУП .....                                                                                      | 9  |
| РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ                                              |    |
| РІШЕНЬ.....                                                                                      | 12 |
| 1.1 Поняття віртуальної реальності та її роль у сучасній освіті.....                             | 12 |
| 1.2 Технічні та психологічні аспекти використання VR в навчанні .....                            | 13 |
| 1.3 . Вплив технологій на підготовку ІТ-спеціалістів та роль VR у цьому процесі .....            | 14 |
| 1.4 Основні напрямки розвитку VR у навчанні ІТ-навичок .....                                     | 15 |
| 1.5 Аналіз існуючих рішень .....                                                                 | 16 |
| 1.6 Технічне завдання на розробку інтерактивного віртуального середовища.....                    | 19 |
| РОЗДІЛ 2 АНАЛІЗ ТА ОЦІНКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ VR-ТЕХНОЛОГІЙ У НАВЧАННІ ІТ-НАВИЧКАМ.....    |    |
| 2.1 Порівняння VR-технологій з традиційними методами навчання .....                              | 24 |
| 2.2 Аналіз переваг та викликів при використанні VR у навчанні.....                               | 26 |
| 2.3 Перспективи розвитку VR для ІТ-освіти.....                                                   | 28 |
| 2.4 Концепція та мета розробки VR-системи для навчання складанню комп'ютерів .....               | 29 |
| 2.5 Рекомендації щодо подальшого використання VR-технологій у навчанні ІТ-навичок.....           | 30 |
| 2.6 Перспективи розвитку VR-технологій у вищій освіті.....                                       | 31 |
| 2.7 Проектування діаграми варіантів використання розробленої системи .                           | 32 |
| 2.8 Схема взаємодії модулів (блоків) системи .....                                               | 34 |
| 2.9 Схема роботи розробленої VR-системи.....                                                     | 37 |
| РОЗДІЛ 3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА ..... |    |
| 3.1 Аналіз програмних пакетів для проектування системи .....                                     | 38 |
| 3.2 Аналіз VR-середовищ для проектування програмної системи .....                                | 42 |

|                                                                                                                                   |    |
|-----------------------------------------------------------------------------------------------------------------------------------|----|
| 3.3 Обґрунтування вибору інструментарію для проектування системи .....                                                            | 46 |
| РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ                                                                                      |    |
| ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА .....                                                                                      | 48 |
| 4.1 Опис підготовчого етапу проектування системи.....                                                                             | 48 |
| 4.2 Проектування інтерактивного віртуального середовища.....                                                                      | 48 |
| 4.3 Завантаження та встановлення ассетів у Unity .....                                                                            | 53 |
| 4.4 Завантаження Meta Quest Link.....                                                                                             | 55 |
| 4.5 Налаштування проєкту для роботи з VR .....                                                                                    | 57 |
| 4.6 Тестування проєкту у гарнітурі Oculus.....                                                                                    | 59 |
| 4.7 Налаштування плагіна Auto Hand .....                                                                                          | 60 |
| 4.8 Інтеграція користувацького інтерфейсу. Опис процесу тестування .....                                                          | 63 |
| 4.9 Налаштування компонента PlacePoint для взаємодії з об'єктами.....                                                             | 67 |
| 4.10 Налаштування PlacePoint із материнською платою .....                                                                         | 69 |
| 4.11 Налаштування інших компонентів комп'ютера .....                                                                              | 71 |
| 4.12 Опис методики тестування. Результати застосування VR<br>у навчальному процесі.....                                           | 72 |
| 4.13 Опис виконаної роботи (практичного кодування) для реалізації<br>та впровадження інтерактивного віртуального середовища ..... | 74 |
| 4.14 Опис етапів при проектуванні програмної системи .....                                                                        | 75 |
| 4.15 Інтеграція компонентів та рекомендації до розробленої програмної<br>системи .....                                            | 75 |
| 4.16 Візуалізація розробки - приклади роботи програми .....                                                                       | 76 |
| ВИСНОВКИ.....                                                                                                                     | 81 |
| СПИСОК ЛІТЕРАТУРИ.....                                                                                                            | 84 |
| Додаток А.....                                                                                                                    | 86 |
| Додаток Б .....                                                                                                                   | 92 |
| Додаток В.....                                                                                                                    | 98 |

## ВСТУП

Сучасна освіта переживає стрімкі зміни, зумовлені розвитком цифрових технологій. Одним із найбільш інноваційних методів навчання є використання віртуальної реальності (VR), яка відкриває нові можливості для взаємодії з навчальним матеріалом. VR дозволяє студентам працювати в інтерактивному середовищі, моделювати процеси, які складно відтворити в реальності, та освоювати практичні навички без ризиків, пов'язаних із фізичними експериментами. Однак, поряд із перевагами, впровадження VR у навчальний процес пов'язане з фінансовими та організаційними викликами, які потребують детального аналізу.

### Переваги використання VR в освіті

Візуалізація складних концепцій. VR дозволяє студентам детально вивчати об'єкти та процеси, які неможливо побачити в реальному світі. Наприклад, учні можуть досліджувати будову комп'ютера на молекулярному рівні або збирати його у віртуальному просторі без необхідності працювати з фізичним обладнанням.

Підвищення залученості та мотивації. Імерсивний характер VR створює ефект присутності, роблячи навчання захопливим. Дослідження показують, що використання VR значно підвищує інтерес студентів до предмета, сприяючи кращому засвоєнню матеріалу.

Безпека та економія ресурсів. Віртуальне середовище дозволяє уникнути ризиків, пов'язаних із неправильною збіркою обладнання або пошкодженням дорогої техніки. Це особливо актуально у навчанні IT-фахівців, яким потрібна практика роботи з комп'ютерними компонентами.

Гнучкість у навчанні. Завдяки VR студенти можуть навчатися в будь-якому місці без необхідності фізично перебувати в лабораторії або аудиторії. Це особливо важливо для дистанційного навчання та програм підготовки спеціалістів у віддалених регіонах.

Реалістичне середовище для відпрацювання навичок. За допомогою VR можна моделювати різні сценарії, включаючи нестандартні ситуації, які неможливо створити в реальному навчальному процесі.

#### Фінансові аспекти впровадження VR

Попри очевидні переваги, оснащення навчального процесу VR-технологіями потребує значних фінансових вкладень. Розглянемо основні статті витрат при обладнанні класу з 30 осіб:

Придбання VR-шоломів. Одним із найпопулярніших пристроїв для освітніх цілей є Meta Quest 3 з обсягом пам'яті 128 ГБ, вартість якого становить близько 25 299 грн за одиницю. Для повного оснащення класу потрібно 30 пристроїв:

$$25\,299 \text{ грн} \times 30 = 758\,970 \text{ грн.}$$

Гігієнічні накладки. Під час використання VR-шоломів різними учнями необхідно враховувати санітарні норми. Комплект із 100 одноразових масок коштує приблизно 1 000 грн. Враховуючи, що за одне заняття потрібно 30 масок, а протягом навчального року проводиться 30 занять, загальна кількість необхідних масок складає 900 штук (9 комплектів):

$$1\,000 \text{ грн} \times 9 = 9\,000 \text{ грн.}$$

Обслуговування та ремонт. VR-обладнання потребує регулярного технічного обслуговування та ремонту. Згідно з рекомендаціями виробників, на це слід виділяти близько 10% від вартості обладнання щорічно:

$$758\,970 \text{ грн} \times 0,10 = 75\,897 \text{ грн на рік.}$$

Додаткові витрати. Окрім основних витрат, можуть знадобитися оновлення програмного забезпечення, підписки на освітні VR-платформи, навчання викладачів та придбання контролерів для взаємодії.

Заміна та модернізація обладнання. У середньому, термін служби VR-шолома складає 4-5 років, після чого може знадобитися його заміна або оновлення комплектуючих.

Загальні витрати на впровадження VR:

VR-шоломи (30 шт.) – 758 970 грн.

Гігієнічні накладки – 9 000 грн.

Обслуговування та ремонт – 75 897 грн.

Додаткові витрати – приблизно 80 000 грн.

Заміна обладнання через 4-5 років – приблизно 200 000 грн.

Разом: 1 123 867 грн.

VR-технології мають величезний потенціал для трансформації освітнього процесу, роблячи його доступнішим, інтерактивнішим та ефективнішим. Однак їх впровадження потребує значних інвестицій як на початковому етапі, так і під час експлуатації. Тому навчальним закладам необхідно ретельно аналізувати фінансові та організаційні аспекти, щоб оцінити доцільність використання VR у навчальному процесі. При правильному підході такі технології можуть значно підвищити якість освіти та підготувати студентів до реальних умов роботи в IT-сфері. До того ж, із розвитком VR-технологій та зниженням вартості обладнання можна очікувати зростання доступності таких методів навчання.

## РОЗДІЛ 1

### ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

#### 1.1 Поняття віртуальної реальності та її роль у сучасній освіті

Віртуальна реальність (VR) стала однією з найбільш перспективних технологій для освітнього процесу в останні десятиліття. Віртуальна реальність є технологією, що дозволяє користувачам занурюватися в імітовані цифрові середовища за допомогою спеціального обладнання, такого як шоломи, рукавички та інші пристрої для відслідковування рухів. Це створює ілюзію присутності в іншому просторі та часі, що відкриває нові можливості для навчання і досліджень.

Використання VR у навчанні дозволяє значно підвищити ефективність освіти, оскільки студенти не просто спостерігають за навчальним процесом, але активно взаємодіють з контентом. Такий підхід допомагає глибше засвоїти матеріал і покращити практичні навички. Зокрема, для підготовки ІТ-спеціалістів, VR може створювати віртуальні лабораторії, в яких студенти можуть виконувати складні завдання без необхідності мати фізичне обладнання.

Дослідження показують, що віртуальні середовища можуть мати значний вплив на процес навчання. Наприклад, вивчення програмування або адміністрування комп'ютерних мереж в умовах віртуальної реальності дозволяє студентам опановувати практичні навички в безпечному середовищі, яке можна легко адаптувати під потреби кожного студента. Такі віртуальні лабораторії дозволяють моделювати ситуації, які важко або неможливо відтворити в реальному світі, наприклад, аварійні ситуації в мережах або серверних системах.

Використання VR у навчанні має чимало переваг. По-перше, це доступність навчальних матеріалів і інструментів. Студенти можуть навчатися без обмежень щодо часу та місця. По-друге, це можливість взаємодії з

навчальним контентом у реальному часі, що значно підвищує ефективність засвоєння матеріалу. Такий підхід дозволяє навчатися навіть складним процесам, таким як налаштування серверів або складання комп'ютера, без необхідності наявності дорогого обладнання або реальних компонентів.

З кожним роком віртуальна реальність стає доступнішою завдяки розвитку апаратних засобів та програмних платформ. Шоломи VR, такі як Oculus Rift, HTC Vive, Microsoft HoloLens, стали значно доступнішими за ціною, що дозволяє все більшій кількості навчальних закладів впроваджувати VR-технології у навчальний процес. Програмне забезпечення для створення віртуальних середовищ, таке як Unity або Unreal Engine, надає розробникам можливість створювати реалістичні та інтерактивні додатки, які можуть бути використані для навчання в різних галузях.

## 1.2 Технічні та психологічні аспекти використання VR у навчанні

Використання віртуальної реальності у навчанні вимагає певних технічних і психологічних передумов для забезпечення максимального ефекту від навчання.

Технічні аспекти. Для ефективного використання VR у навчальному процесі необхідно забезпечити наявність високопродуктивного комп'ютерного обладнання, яке підтримує роботу з віртуальними середовищами. Це включає комп'ютери з потужними процесорами і відеокартами, а також спеціалізоване обладнання — шоломи для занурення, контролери для взаємодії з об'єктами та датчики руху, що дозволяють відслідковувати рухи користувача. Для розробки VR-додатків використовуються такі програмні платформи, як Unity і Unreal Engine, які дозволяють створювати інтерактивні 3D-середовища, що відтворюють реальні об'єкти і сценарії.

Крім того, важливим аспектом є створення реалістичних 3D-моделей і симуляцій. Для цього використовуються інструменти для 3D-моделювання,

такі як Blender або Autodesk Maya, які дозволяють створювати об'єкти з високим рівнем деталізації. Використання таких інструментів допомагає створювати навчальні програми, які є не лише технічно правильними, а й інтуїтивно зрозумілими для користувачів.

Психологічні аспекти. Віртуальне середовище здатне значно впливати на когнітивні процеси користувачів. Занурення в віртуальну реальність стимулює увагу, підвищує мотивацію і дозволяє краще сприймати і запам'ятовувати матеріал. Однією з переваг VR є можливість створення "персоналізованих" навчальних середовищ, де кожен студент може працювати у своєму темпі, отримуючи миттєвий зворотний зв'язок щодо своїх дій. Студент може легко виправляти помилки, що виникають у процесі навчання, що допомагає йому швидше освоїти необхідні навички.

Віртуальні середовища мають великий потенціал для розвитку інтерактивного навчання. Студенти мають можливість не лише спостерігати, а й активно взаємодіяти з навчальними матеріалами, що дозволяє покращити їх розуміння. Такий підхід стимулює їх до більш глибокого осмислення та застосування набутих знань у реальних ситуаціях.

### 1.3. Вплив технологій на підготовку ІТ-спеціалістів та роль VR у цьому процесі

ІТ-освіта є однією з найбільш динамічно розвиваючих сфер, оскільки вимоги до кваліфікації спеціалістів постійно зростають. Підготовка ІТ-фахівців потребує не лише теоретичних знань, а й практичних навичок, які можна набувати через вирішення реальних завдань. Традиційні методи навчання часто не забезпечують достатньої практичної підготовки, тому важливо впроваджувати інноваційні технології, які можуть значно покращити процес навчання.

Віртуальна реальність є ідеальним інструментом для навчання складним технічним навичкам, таким як програмування, налаштування комп'ютерних

систем або адміністрування мереж. За допомогою VR можна створити віртуальні лабораторії, де студенти можуть працювати з реальними інструментами, не боячись пошкодити дорогі компоненти або обладнання.

Завдяки VR, студенти можуть моделювати реальні ситуації, що значно покращує їх практичну підготовку. Наприклад, при навчанні складанню комп'ютерів студенти можуть взаємодіяти з компонентами, підключати їх, налаштовувати операційну систему та перевіряти працездатність без необхідності мати фізичний доступ до реальних частин комп'ютера. Це дає можливість студентам працювати з різними типами конфігурацій і операційних систем, що значно підвищує їх рівень підготовки.

#### 1.4 Основні напрямки розвитку VR у навчанні ІТ-навичок

Розглянемо основні напрямки розвитку VR у навчанні ІТ-навичок:

1. Розширення функціональності VR-систем. Подальше вдосконалення VR-освіти передбачає інтеграцію складніших завдань, таких як налаштування серверних систем, управління віртуальними мережами та програмування у VR-середовищах. Віртуальні платформи можуть включати модулі для навчання кібербезпеки, роботи з базами даних та тестування програмного забезпечення в інтерактивному форматі.

2. Інтеграція VR з штучним інтелектом. Використання алгоритмів штучного інтелекту у VR-системах дозволить адаптувати навчальні програми під рівень знань кожного студента. Наприклад, AI може аналізувати успішність виконання завдань, визначати слабкі місця та пропонувати додаткові вправи для покращення навичок. Це сприятиме персоналізації навчального процесу, що підвищить його ефективність.

3. Розвиток дистанційного навчання з VR. Віртуальна реальність сприяє створенню повноцінних дистанційних освітніх програм, що дозволяє студентам взаємодіяти з викладачами та іншими учасниками курсу незалежно від географічного розташування. Це відкриває нові можливості для

міжнародної співпраці, доступу до провідних навчальних платформ та реалізації масштабних освітніх ініціатив.

4. Імітація робочих процесів у VR. Віртуальні середовища можуть точно відтворювати робочі процеси у сфері IT, дозволяючи студентам відпрацьовувати завдання в умовах, наближених до реальних. Наприклад, симуляція DevOps-процесів, управління хмарними обчисленнями або тестування програмного забезпечення у VR допоможе підготувати спеціалістів до реальної роботи у галузі.

5. Інтеграція VR у корпоративне навчання. Багато IT-компаній вже впроваджують VR для внутрішнього навчання персоналу, і ця тенденція лише посилюватиметься. Використання VR-симуляцій дозволяє ефективно навчати нових співробітників, тестувати їхні навички та моделювати складні робочі сценарії без ризиків для бізнесу.

Зважаючи на стрімкий розвиток VR-технологій, можна прогнозувати, що їхня роль у навчанні IT-спеціалістів лише зростатиме. Навчальні заклади, які інтегрують VR у свої програми, зможуть підготувати конкурентоспроможних фахівців, здатних швидко адаптуватися до сучасного технологічного середовища. Успішна реалізація VR-освіти залежатиме від подальшого розвитку апаратного забезпечення, вдосконалення методик навчання та розширення доступності цих технологій для студентів у всьому світі.

### 1.5 Аналіз існуючих рішень

У сучасній освіті та професійній підготовці все частіше використовуються віртуальні симулятори, які забезпечують безпечне та ефективне середовище для навчання. Розглянемо три існуючі системи: Forklift Simulator VR, PC Building Simulator та Mursion, їхні переваги, недоліки та вартість.

#### 1. Forklift Simulator VR

Опис: Forklift Simulator VR — це віртуальний симулятор керування навантажувачем, який надає користувачам можливість відчувати реалістичний досвід роботи з цим обладнанням у безпечному середовищі.

Переваги:

- безпека: дозволяє користувачам навчатися без ризику для здоров'я та безпеки, що особливо важливо при роботі з важкою технікою;
- реалістичність: використання VR-технологій забезпечує високий рівень занурення та реалістичності, що сприяє кращому засвоєнню навичок;
- ефективність: можливість багаторазового повторення сценаріїв дозволяє користувачам відпрацьовувати навички до автоматизму;

Недоліки:

- обмежена сфера застосування: Симулятор зосереджений лише на навичках керування навантажувачем, що обмежує його використання в інших сферах.

Вартість: Forklift Simulator VR доступний на платформах, таких як Steam, за ціною близько \$7.99.

## 2. PC Building Simulator

Опис: PC Building Simulator — це симулятор складання комп'ютерів, який дозволяє користувачам навчитися збирати, налаштовувати та ремонтувати комп'ютери у віртуальному середовищі.

Переваги:

- освітній потенціал: дозволяє користувачам отримати практичні навички складання та налаштування комп'ютерів без необхідності в реальних компонентах;
- економія ресурсів: використання симулятора дозволяє зекономити на придбанні дорогих компонентів для навчання;
- доступність: програма доступна для широкого кола користувачів і не вимагає спеціалізованого обладнання.
- Недоліки:

- обмежена реалістичність: хоча симулятор надає детальну інформацію про компоненти, він не може повністю відтворити фізичні аспекти складання комп'ютера;
- відсутність VR-підтримки: програма не підтримує VR, що знижує рівень занурення;
- вартість: PC Building Simulator доступний на платформах, таких як Steam, за ціною близько \$20.

### 3. Mursion

Опис: Mursion — це платформа для підвищення кваліфікації, яка поєднує штучний інтелект та людський фактор для створення імерсивних симуляцій, спрямованих на розвиток професійних навичок.

#### Переваги:

- універсальність: платформа підходить для тренінгів у різних галузях, включаючи освіту, охорону здоров'я та бізнес;
- індивідуалізація: можливість налаштування сценаріїв під конкретні потреби організації або користувача;
- реалістичність: поєднання AI та живих акторів забезпечує високий рівень реалістичності взаємодії.

#### Недоліки:

- вартість: висока вартість підписки може бути недоступною для невеликих організацій або окремих користувачів;
- залежність від інтернету: для роботи платформи потрібне стабільне інтернет-з'єднання;
- вартість: Mursion працює за моделлю підписки, вартість якої залежить від обсягу та тривалості використання.

Проаналізувавши існуючі аналоги, такі як Forklift Simulator VR, PC Building Simulator та Mursion, можна зробити висновок, що кожна з цих систем має свої переваги та недоліки.

Однак жодна з них не забезпечує повного спектру можливостей, необхідних для нашої специфічної сфери застосування. Тому доцільно розробити власну систему, яка буде враховувати особливості нашої галузі, забезпечуючи більш високу ефективність та адаптивність до потреб користувачів.

## 1.6 Технічне завдання на розробку інтерактивного віртуального середовища

Будь-яка програмна розробка (довідково-інформаційна система, програмно-педагогічний засіб, електронний ресурс тощо), на початковому етапі повинна передбачати створення технічного завдання (ТЗ). Цей документ регламентує функціональність програмного засобу та визначає вимоги до процесу впровадження та звітної документації. При цьому виконавець робіт повинен зрозуміти сутність завдання, показати замовнику приблизний інтерфейс майбутнього програмного продукту, а замовнику зрозуміти, підтвердити та розписатися в тому, що саме йому потрібно у цьому програмному засобі.

При написанні ТЗ ми орієнтувалися на ГОСТ 34.602-89. Цей стандарт розповсюджується на автоматизовані системи для автоматизації різних видів діяльності (управління, проектування, дослідження тощо), включаючи їх поєднання, і встановлює склад, зміст, правила оформлення документа «Технічне завдання на створення (розвиток або модернізацію) системи».

Вимоги до введення програмного засобу в дію

### 1. Вступ

#### 1.1 Загальні визначення

Повна назва програмного продукту (тема дипломної роботи): «Проектування інтерактивного віртуального середовища для поглиблення ІТ-навичок та вмінь студентів ЗВО за допомогою іммерсивних технологій віртуальної реальності».

Замовник – організація, яке замовляє створення програмного засобу.

Виконавець – програміст, який розробляє програмний засіб та встановлює його на комп'ютер Замовника або передає Замовнику готовий програмний продукт, в даному випадку це студент М2 курсу очного відділення Мелешко О.Д.

## 1.2 Призначення документу

У цьому документі приводиться повний перелік вимог до програмної реалізації та впровадження програмного засобу. У цьому документі Замовник і Виконавець підтверджують їх згоду з нижченаведеними фактами і умовами:

1. Виконавець підготував і розробив цей документ, що іменується Технічне Завдання, який містить перелік вимог до виконуваних робіт.

2. Замовник згоден зі всіма положеннями цього Технічного Завдання.

3. Замовник не має права вимагати від Виконавця в рамках поточного Договору виконання робіт або надання послуг, прямо не описаних у справжньому Технічному Завданні.

4. Виконавець зобов'язується виконати роботи в об'ємі, вказаному в справжньому Технічному Завданні.

5. Замовник не має права вимагати від Виконавця дотримання яких-небудь форматів і стандартів, якщо це не вказано в справжньому Технічному Завданні.

6. Все неоднозначності, виявлені в справжньому Технічному завданні після його підписання, підлягають двосторонньому узгодженню між Сторонами. В процесі узгодження можуть бути розроблені додаткові вимоги, які оформляються додатковою угодою до Договору і відповідним чином оцінюються.

## 2. Призначення розробки

### 2.1 Призначення і мета розробки програмного засобу

Предметом розробки є інтерактивне віртуальне середовище для поглиблення ІТ-навичок та вмінь студентів ЗВО за допомогою іммерсивних технологій віртуальної реальності.

Мета – це розробка інтерактивного віртуального середовища для навчання студентів технічних спеціальностей складанню комп'ютерів за допомогою технологій віртуальної реальності. Розроблене VR-середовище дозволить студентам на практиці освоїти основи складання комп'ютерних систем, що є важливою частиною підготовки майбутніх ІТ-фахівців. Крім того, використання VR дасть можливість створити більш доступний і ефективний навчальний процес, який буде орієнтований на практичні навички студентів, а також стимулюватиме їх мотивацію.

### 3. Вимоги до програмного засобу

#### 3.1 Вимоги до оформлення інтерфейсу

Графічний інтерфейс програмної системи повинен бути легким (розрахованим на користувача з початковим рівнем інформаційної культури та навичок роботи за комп'ютером), але в той же час функціональним.

#### 3.2 Вимоги до програмного та технічного забезпечення

Для функціонування програмного засобу необхідне наступне програмне та мінімальне апаратне забезпечення:

- Операційна система – бажано Windows 7 та більш пізні версії, краще Windows 10 або 11;
- Framework 4.5;
- процесор – Intel Pentium IV 1 Ghz або більш потужний;
- оперативна пам'ять – 4 Гб RAM або більше;
- жорсткий диск – 500 Гб або більше.

### 4. Стадії і етапи розробки програмного засобу

#### 4.1 Склад і зміст робіт із розробки програмного засобу

Розробка програмного засобу (інтерактивного віртуального середовища) передбачає виконання наступних етапів:

- аналіз існуючих підходів до використання віртуальної реальності в навчальному процесі;
- розробка концепції VR-середовища для навчання складанню ПК;

- створення та тестування VR-програми, яка дозволяє студентам виконувати практичні завдання по складанню комп'ютера у віртуальному середовищі;
- оцінка ефективності застосування VR у навчанні ІТ-навичок та порівняння результатів з традиційними методами навчання;
- тестування і перевірка працездатності програмного продукту в реальних умовах;
- виправлення помилок роботи програми і проведення завершального показу Замовнику.

5. Перелік обов'язкових етапів при виконанні практичного проектування системи (основні етапи кодування):

1. Інтеграція плагіна Auto Hand:

- реалізація базового налаштування плагіна для створення інтерактивного середовища, включаючи запуск Setup Wizard і налаштування параметрів фізики.

2. Налаштування взаємодії з об'єктами:

- використання компонент PlacePoint для забезпечення можливості точного розміщення об'єктів, таких як оперативна пам'ять, на сцені;
- налаштування функції Parent on Place, Force Hand Release і Disable RigidBody on Place для оптимізації фізичної взаємодії об'єктів.

3. Створення інтерактивних точок:

- встановлення PlacePoint для всіх ключових компонентів: CPU, RAM, GPU, PSU, материнської плати, охолодження процесора та корпусу системного блоку;
- використання функції StartPlaced для забезпечення автоматичного розміщення об'єктів у початкових позиціях.

4. Тестування інтерактивності:

- проведення перевірки коректної роботи Grabbable та PlacePoint для кожного об'єкта у режимі Play Mode;

- виконання налаштування точного позиціонування об'єктів у паузі Play Mode із подальшим збереженням параметрів.

#### 5. Інтеграція фізики об'єктів:

- додавання Rigidbody та Box Collider до кожного компонента для забезпечення реалістичної фізичної взаємодії.

#### 6. Оптимізація взаємодії:

- забезпечення плавності інтерактивності шляхом уникнення зайвих фізичних розрахунків завдяки функції Disable Rigidbody on Place.

## РОЗДІЛ 2

### АНАЛІЗ ТА ОЦІНКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ VR-ТЕХНОЛОГІЙ У НАВЧАННІ ІТ-НАВИЧКАМ

#### 2.1 Порівняння VR-технологій з традиційними методами навчання

Традиційні методи навчання складання персонального комп'ютера зазвичай передбачають використання фізичних комплектуючих, друкованих посібників, лекційного викладу та практичних занять у лабораторних умовах. Однак ці методи мають ряд обмежень, які можуть впливати на якість засвоєння матеріалу та ефективність навчального процесу.

Одним із головних викликів є *доступність апаратного забезпечення*. Багато навчальних закладів не можуть забезпечити кожного студента окремими комплектами для складання комп'ютера через високу вартість обладнання. У такому випадку студенти змушені працювати в групах, що обмежує їхню практичну взаємодію та індивідуальну підготовку. Крім того, через часте використання комплектуючі можуть зношуватися або виходити з ладу, що ускладнює проведення занять.

VR-технології дозволяють усунути ці недоліки, забезпечуючи доступ до необмеженої кількості комплектуючих у віртуальному середовищі. Студенти можуть практикувати процес складання без ризику пошкодження деталей та необхідності чекати черги для доступу до лабораторного обладнання. Завдяки інтерактивності вони отримують можливість багаторазово повторювати процеси, закріплюючи свої навички у безпечному середовищі.

Аналіз швидкості та ефективності навчання.

Порівняльний аналіз результатів навчання показав, що використання VR значно скорочує час, необхідний для освоєння складних технічних завдань. Студенти, які використовували VR-додаток, виконували вправи на 30-40% швидше, ніж їхні колеги, які працювали з фізичними комплектуючими. Вони

могли негайно отримати зворотний зв'язок про свої помилки та швидко їх виправити, що сприяло кращому засвоєнню матеріалу.

Крім того, можливість багаторазового повторення практичних завдань без фізичних обмежень дозволяє більш глибоко опрацювати матеріал та відточити навички. У традиційних умовах студенти мають лише кілька спроб на складання або розбирання ПК, оскільки надмірне використання обладнання може спричинити його зношення. У VR таких обмежень немає, що дає змогу кожному студенту тренуватися стільки, скільки потрібно для досягнення впевненості у своїх діях.

Гнучкість навчального процесу.

Ще однією суттєвою перевагою VR є можливість навчання без прив'язки до розкладу лабораторних занять та доступності фізичного обладнання. Віртуальне середовище доступне в будь-який час, що дозволяє студентам проходити навчальні модулі у зручний для них момент. Це особливо важливо для дистанційного навчання, де обмежений доступ до фізичних лабораторій може ускладнювати засвоєння матеріалу.

Завдяки інтерактивному формату подачі інформації навчання стає більш захопливим, що позитивно впливає на рівень концентрації та зацікавленості студентів. Багато досліджень підтверджують, що активна взаємодія з навчальними об'єктами у VR сприяє кращому запам'ятовуванню матеріалу та розвиває просторове мислення.

Економічна ефективність VR-навчання.

З економічної точки зору VR-навчання є вигідною альтернативою традиційним лабораторним заняттям. Навчальні заклади можуть значно скоротити витрати на закупівлю комплектуючих, їхнє оновлення та технічне обслуговування. Віртуальні моделі не зношуються, не ламаються та не потребують регулярного оновлення, що робить навчальний процес більш доступним і менш затратним.

Окрім цього, VR-технології відкривають можливість створення сценаріїв, які неможливо реалізувати у фізичних лабораторіях. Наприклад,

можна моделювати рідкісні несправності обладнання, проводити тренування на діагностику та ремонт у реалістичних умовах, що значно підвищує рівень підготовки студентів до реальних робочих ситуацій.

Таким чином, аналіз показує, що впровадження VR у навчальний процес не лише підвищує ефективність навчання, але й забезпечує економічні переваги, доступність та гнучкість. Студенти отримують унікальну можливість практикуватися в будь-який час, виправляти помилки без наслідків та ефективніше засвоювати матеріал. VR-навчання поступово стає важливим елементом освітніх програм, адаптуючи підготовку спеціалістів до сучасних технологічних викликів.

## 2.2 Аналіз переваг та викликів при використанні VR у навчанні

Використання віртуальної реальності у навчальному процесі відкриває нові можливості для підготовки студентів, однак водночас створює низку викликів, які потребують вирішення для досягнення максимальної ефективності. Нижче розглянуто ключові переваги та обмеження впровадження VR у навчальні програми.

Переваги VR у навчанні:

1. Інтерактивність та занурення Віртуальна реальність дозволяє створювати високореалістичні навчальні середовища, у яких студенти можуть взаємодіяти з навчальним матеріалом так, як це неможливо у традиційному навчанні. Маніпуляція об'єктами, їхній візуальний аналіз та відтворення реальних сценаріїв у VR підсилюють процес навчання та сприяють кращому засвоєнню інформації.

2. Необмежена доступність до навчального контенту VR-навчання не потребує фізичних компонентів, що особливо важливо для студентів, які не мають доступу до реального обладнання. Віртуальні лабораторії дають можливість навчатися у будь-який зручний час та не залежать від розкладу аудиторних занять чи наявності лабораторного інвентаря.

3. Мотивація та зацікавленість студентів Завдяки ігровим механікам та інтерактивним можливостям VR-середовище робить навчальний процес більш захопливим. Дослідження показують, що студенти, які навчаються за допомогою VR, демонструють вищий рівень зацікавленості та активного залучення в процес навчання у порівнянні з традиційними методами.

4. Безпека навчання Одним із головних ризиків практичного навчання є можливість пошкодження обладнання або отримання студентами травм при роботі з технічними засобами. У VR-середовищі ці ризики відсутні: студенти можуть експериментувати, допускати помилки та виправляти їх без жодних негативних наслідків.

Виклики впровадження VR у навчальний процес:

1. Фінансова доступність та обладнання Головним бар'єром для широкого впровадження VR є висока вартість шоломів віртуальної реальності, потужних комп'ютерів та необхідного програмного забезпечення. Для навчальних закладів, особливо бюджетних, ці витрати можуть стати серйозною перешкодою для інтеграції VR у навчальні курси.

2. Технічні вимоги та інфраструктура VR-програми потребують сучасного обладнання, яке здатне підтримувати високу частоту кадрів і забезпечувати стабільну роботу без затримок. Не всі навчальні заклади мають відповідну інфраструктуру, що обмежує можливості використання VR-технологій у широкому масштабі.

3. Адаптація викладачів до нових технологій Викладачі, які звикли до традиційних методик, можуть стикатися з труднощами при впровадженні VR у навчальний процес. Для успішної інтеграції VR-навчання необхідна підготовка педагогів, розробка нових методичних матеріалів та адаптація навчальних планів.

4. Психофізіологічні обмеження Тривале перебування у VR-середовищі може викликати втому, запаморочення або інші фізіологічні реакції. Це створює певні обмеження щодо тривалості використання VR у

навчальному процесі, що потребує розробки спеціальних рекомендацій для оптимізації навчальних сесій.

Таким чином, хоча VR-освіта стикається з певними технічними та фінансовими викликами, її переваги значно переважають можливі недоліки. Підвищена інтерактивність, доступність, безпечність та залученість студентів свідчать про перспективність використання віртуальної реальності у навчальному процесі. Водночас, розвиток технологій та поступове зниження вартості обладнання сприятимуть ширшому впровадженню VR у навчальні заклади. Успішна інтеграція VR-навчання залежить від ефективної адаптації освітніх програм, підготовки викладачів та подальшого вдосконалення апаратної частини. Зважаючи на всі ці аспекти, можна очікувати, що VR-технології стануть важливим інструментом у підготовці майбутніх спеціалістів.

### 2.3 Перспективи розвитку VR у навчанні ІТ-навичок

Використання VR у підготовці ІТ-спеціалістів відкриває широкі можливості для вдосконалення навчального процесу, надаючи студентам доступ до інтерактивних симуляцій, практичних завдань та реальних сценаріїв роботи з технологіями.

З розвитком обчислювальних потужностей та зниженням вартості VR-обладнання ці технології стають дедалі доступнішими для навчальних закладів. У перспективі VR може змінити підхід до навчання ІТ-спеціальностей, зробивши його більш ефективним та адаптивним до сучасних викликів.

## 2.4 Концепція та мета розробки VR-системи для навчання складанню комп'ютерів

Основною метою розробки VR-системи є створення інструменту для покращення навчання студентів ІТ-спеціальностей, який дозволяє їм на практиці освоювати процес складання комп'ютерів.

Складання комп'ютерів є важливою складовою підготовки сучасних ІТ-спеціалістів, однак традиційні методи навчання не завжди забезпечують достатню гнучкість і доступність для кожного студента.

Більше того, для деяких студентів реальна робота з фізичними компонентами може бути складною через обмежений доступ до обладнання або ресурсів.

Використання віртуальної реальності дозволяє усунути ці обмеження, оскільки віртуальні середовища можна створювати без необхідності фізичного доступу до реальних компонентів комп'ютера.

Віртуальні компоненти комп'ютера (материнські плати, процесори, оперативна пам'ять, відеокарти та інше обладнання) можуть бути точно відтворені у VR, дозволяючи студентам вивчати їх взаємодію в інтерактивному форматі.

Для забезпечення ефективності навчального процесу важливо, щоб програма була інтуїтивно зрозумілою і доступною для студентів різних рівнів підготовки.

У цьому контексті, створення віртуальної лабораторії, де студенти можуть практикувати складання комп'ютера, тестування і налагодження системи без необхідності використовувати реальні компоненти, є важливим етапом у розробці програми.

Процес розробки має включати кілька етапів: створення віртуальних моделей компонентів, розробка інтерактивного інтерфейсу, програмування механіки взаємодії користувача з об'єктами віртуального середовища та

тестування програми для забезпечення її ефективності та зручності використання.

## 2.5 Рекомендації щодо подальшого використання VR-технологій у навчанні ІТ-навичок

З огляду на успішність впровадження VR-технологій у навчання складанню комп'ютерів, можна зробити кілька рекомендацій для подальшого розвитку цього напрямку:

1) Розширення функціональності VR-системи. Для покращення навчального процесу в майбутньому можна додати більше варіантів практичних завдань, зокрема симуляції складніших ситуацій, таких як діагностика несправностей комп'ютерних систем або вирішення проблем з програмним забезпеченням. Це дозволить студентам отримати більш комплексне розуміння процесів, що відбуваються в комп'ютерних системах.

2) Інтеграція штучного інтелекту (AI). Впровадження AI в систему може дозволити адаптувати навчальний процес до індивідуальних потреб кожного студента, надаючи персоналізовані підказки та рекомендації з покращення навичок. AI може також бути використаний для аналізу результатів студентів і пропонувати конкретні завдання для подолання слабких місць.

3) Зниження вартості обладнання. Одним з основних викликів є висока вартість VR-шоломів та комп'ютерного обладнання, що може обмежувати доступ до технологій для навчальних закладів, особливо з обмеженим бюджетом. Зниження вартості обладнання дозволить більшій кількості навчальних закладів впроваджувати VR як основний інструмент для навчання.

4) Розвиток дистанційного навчання. З урахуванням зростання популярності дистанційної освіти, необхідно розробити модулі для навчання ІТ-спеціалістів, які дозволяють здійснювати навчання з будь-якої точки світу. Дистанційне навчання у VR дозволить студентам з різних країн проходити

однакові навчальні курси та практичні завдання, забезпечуючи рівні можливості для всіх.

## 2.6 Перспективи розвитку VR-технологій у вищій освіті

Віртуальна реальність має величезний потенціал для подальшого розвитку у вищій освіті, зокрема для підготовки ІТ-спеціалістів. Перспективи використання VR включають створення нових форм навчальних середовищ та інтеграцію з іншими сучасними технологіями:

1) Створення віртуальних лабораторій. Завдяки розвитку VR можна створювати віртуальні лабораторії для різних дисциплін, де студенти зможуть не лише взаємодіяти з навчальним контентом, а й проводити дослідження та експерименти, що неможливо здійснити в реальному світі через обмеження простору або ресурсів. Це дозволить створити більш інклюзивні і доступні можливості для студентів з усього світу.

2) Застосування VR для міждисциплінарних курсів. Використання VR дозволяє об'єднувати різні дисципліни, створюючи навчальні програми, що поєднують ІТ, інженерію, медицину та інші галузі. Це дозволить студентам отримувати комплексні знання, які є необхідними для вирішення реальних проблем в різних сферах.

3) Інтеграція штучного інтелекту та машинного навчання. Інтеграція AI в VR-системи дозволить створювати адаптивні навчальні середовища, які будуть враховувати індивідуальні потреби кожного студента. Це забезпечить більш ефективне навчання і дозволить студентам краще освоювати матеріал, орієнтуючись на їхні сильні та слабкі сторони.

4) Масштабування VR-систем для великих аудиторій. З розвитком технологій VR можна буде створювати масові онлайн-курси, де студенти з різних куточків світу можуть навчатися разом у віртуальних класах. Це дозволить знизити бар'єри для навчання і забезпечити рівні можливості для всіх студентів, незалежно від їхнього місця знаходження.

Таким чином, VR-технології мають величезний потенціал не лише для покращення навчання IT-спеціалістів, а й для розвитку вищої освіти в цілому.

Віртуальна реальність дозволяє створити нові можливості для навчання, що можуть змінити традиційні підходи до освіти та підвищити її доступність та ефективність.

## 2.7 Проєктування діаграми варіантів використання розробленої системи

Діаграма варіантів використання (Use Case Diagram) є важливим інструментом проєктування програмних систем, який дозволяє візуалізувати функціональність системи з точки зору користувача. Вона допомагає визначити основні сценарії взаємодії, що відображаються у вигляді акторів (користувачів або зовнішніх систем) та варіантів використання (дій, які вони можуть виконувати в системі). Ця діаграма сприяє розумінню функціональних можливостей системи та її ключових процесів.

Основні елементи діаграми прецедентів:

Актори – користувачі системи або зовнішні компоненти, що взаємодіють із нею.

Прецеденти – функції або сценарії, доступні для акторів.

Зв'язки – асоціації між акторами та прецедентами, що показують можливі взаємодії.

Діаграма варіантів використання для VR-системи

У розробленій VR-системі для навчання складанню комп'ютерів основними акторами є:

Студент – користувач, який взаємодіє із віртуальним середовищем для навчання.

Викладач – контролює процес навчання, переглядає прогрес студентів.

Система VR – відповідає за обробку дій користувача, візуалізацію та фізичні взаємодії.

Основні прецеденти:

Запуск навчального модуля – студент починає роботу у VR-середовищі.

Взаємодія з компонентами ПК – користувач бере, встановлює або змінює компоненти.

Отримання підказок – система надає допомогу у виборі або встановленні компонентів.

Оцінювання прогресу – викладач аналізує результати навчання.

Завершення симуляції – студент закінчує заняття, система підбиває підсумки.

Діаграма варіантів використання (Use Case Diagram) є важливим інструментом проєктування програмних систем, який дозволяє візуалізувати функціональність системи з точки зору користувача. Вона допомагає визначити основні сценарії взаємодії, що відображаються у вигляді акторів (користувачів або зовнішніх систем) та варіантів використання (дій, які вони можуть виконувати в системі). Ця діаграма сприяє розумінню функціональних можливостей системи та її ключових процесів (рис. 2.1).

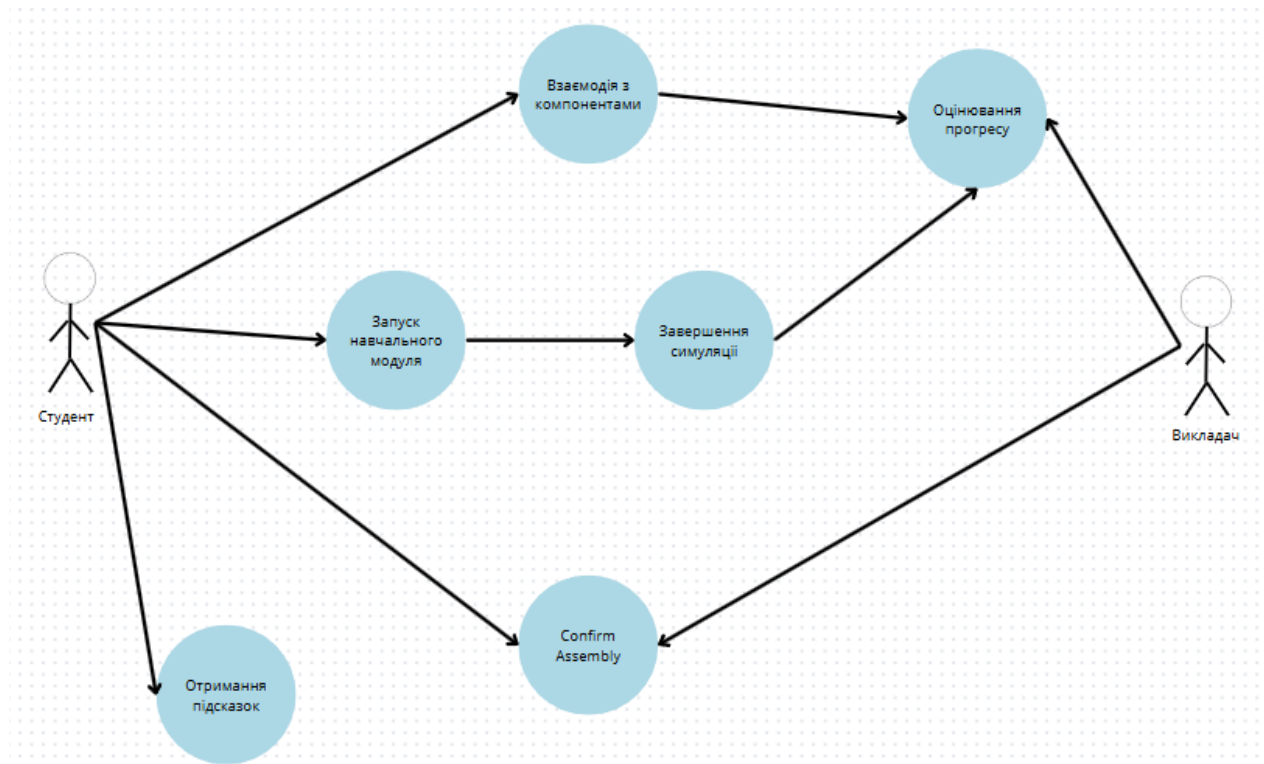


Рисунок 2.1 - Діаграма варіантів використання (діаграма прецедентів) розробленої систем

Вище наведена діаграма варіантів використання (діаграма прецедентів) для VR-системи навчання складанню комп'ютерів (рис. 2.1). Вона містить:

Акторів:

Студент – взаємодіє з навчальним середовищем.

Викладач – оцінює прогрес студентів.

Основні сценарії використання:

Запуск навчального модуля (студент розпочинає роботу).

Взаємодія з компонентами ПК (використання VR для збирання комп'ютера).

Отримання підказок (допомога при складанні).

Оцінювання прогресу (викладач аналізує результати).

Завершення симуляції (система підбиває підсумки).

## 2.8 Схема взаємодії модулів (блоків) системи

Розроблена VR-система для навчання складанню комп'ютера складається з кількох основних модулів, які взаємодіють між собою, забезпечуючи коректний навчальний процес. У цьому розділі представлена схема взаємодії цих модулів та описано їх функціональну роль у системі.

Основними модулями системи є (рис. 2.2):

- VR-клієнт – головний компонент, що відповідає за візуалізацію та взаємодію користувача із середовищем;
- модуль фізичної взаємодії – забезпечує відстеження рухів рук та їхню передачу в систему.
- модуль взаємодії з об'єктами – керує тим, як користувач може брати та розміщувати об'єкти;
- система обмежень розміщення об'єктів – контролює правильність встановлення компонентів, не дозволяючи розміщувати їх у неправильних місцях.

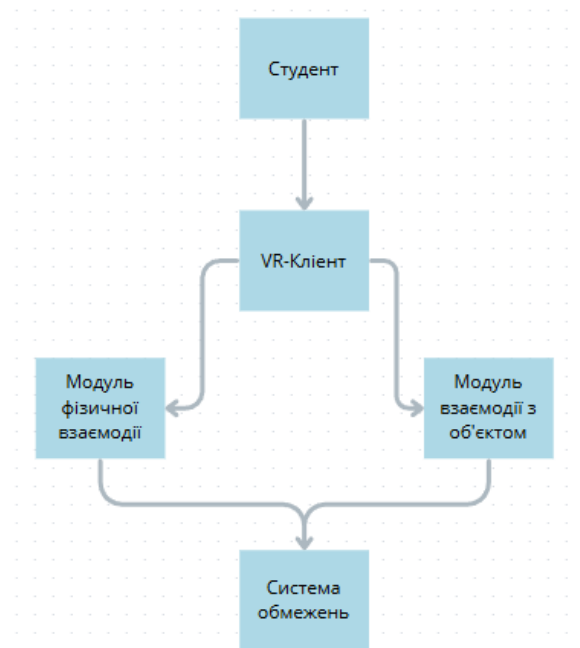


Рисунок 2.2 - Схема взаємодії модулів (блоків) розробленої VR-системи

#### Опис взаємодії модулів

Процес роботи VR-системи включає наступні основні етапи:

1. Студент входить у VR-клієнт, де отримує доступ до робочого середовища.
2. Модуль фізичної взаємодії відстежує рухи рук студента та передає їх у VR-клієнт для подальшої обробки.
3. Модуль взаємодії з об'єктами дозволяє студенту брати компоненти та переміщувати їх у просторі.
4. Система обмежень розміщення об'єктів контролює правильність дій студента:
  - якщо компонент знаходиться у правильному місці, він фіксується у відповідному слоті;
  - якщо компонент розміщується неправильно, система не дозволяє його встановити.

#### Функціональна логіка системи:

- якщо студент бере компонент, VR-клієнт передає цю дію в модуль фізичної взаємодії;

- якщо компонент переміщується, система перевіряє його відповідність правильному розташуванню;
- якщо студент намагається встановити компонент у неправильному місці, система блокує цю дію без додаткових підказок;
- якщо студент встановлює компонент у правильному місці, він автоматично закріплюється.

Запропонована система дозволяє студентам освоювати процес складання ПК у правильній послідовності без можливості допущення помилок. Відсутність додаткових підказок змушує користувача аналізувати можливі варіанти самостійно, що сприяє кращому розумінню принципів роботи з комплектуючими.

## 2.9 Схема роботи розробленої VR-системи

Для наочного представлення принципу роботи VR-системи навчання складанню комп'ютерів використовується блокова схема. Вона демонструє основні етапи роботи системи у вигляді послідовності взаємопов'язаних блоків, що відображають ключові процеси.

### Основні блоки системи

Запуск системи – ініціалізація VR-середовища, завантаження навчального модуля.

Трекінг рухів користувача – система використовує Unity XR Toolkit для обробки рухів контролерів та рук у VR.

Фізична взаємодія з об'єктами – за допомогою AutoHand користувач може брати, переміщувати та фіксувати компоненти.

Перевірка правильності складання – система аналізує, чи правильно встановлені компоненти, надаючи зворотний зв'язок.

Завершення симуляції – оцінювання результатів, завершення навчального процесу.

### Послідовність роботи системи

Блоки розташовані у логічному порядку, з чіткими переходами між ними. Кожен етап веде до наступного (рис. 2.3):

Запуск системи → Трекінг рухів користувача → Фізична взаємодія з об'єктами → Перевірка правильності складання → Завершення симуляції.

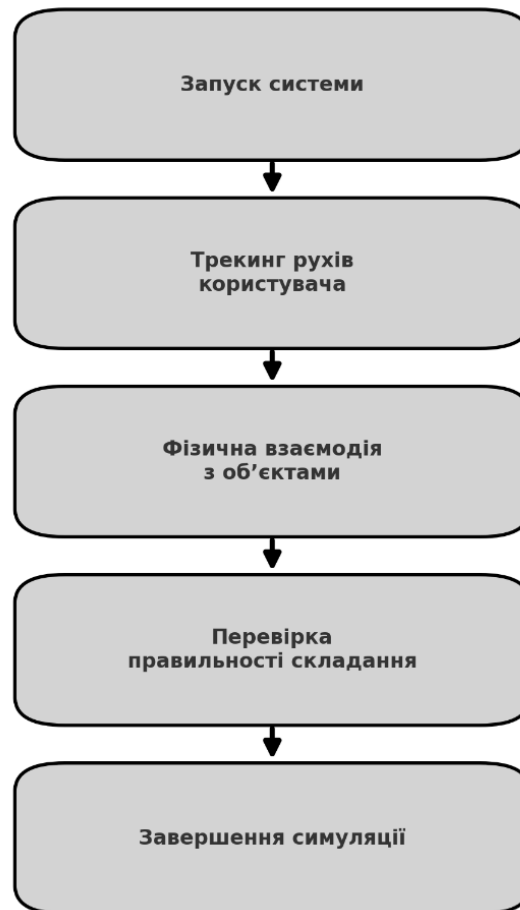


Рисунок 2.3 - Схема роботи розробленої VR-системи

Ця блокова схема відображає послідовність етапів роботи системи та їхню взаємодію, дозволяючи легко зрозуміти, як користувач взаємодіє з середовищем та як обробляються його дії у VR.

## РОЗДІЛ 3

### ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА

#### 3.1 Аналіз програмних пактів для проектування системи

Виконаємо аналіз найбільш відомих програмних пактів для проектування нашої системи. А саме розглянемо програмний пакет для створення тривимірної комп'ютерної графіки – Blender, та графічний редактор, для моделювання тривимірних об'єктів, анімації, композитингу та візуалізації - Autodesk Maya.

Програмний пакет Blender.

Blender — програмний пакет для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, рендерингу, після-обробки відео. До версії 2.80 містив рушій Blender Game Engine для створення відеоігор. Пакет є вільним програмним забезпеченням та розповсюджується під ліцензією GNU GPL.

Особливостями пакету є малий розмір, висока швидкість рендерингу, наявність версій для багатьох операційних систем — FreeBSD, GNU/Linux, Mac OS X, SGI Irix 6.5, Sun Solaris 2.8 (sparc), Microsoft Windows, SkyOS, MorphOS та Pocket PC. Пакет має такі функції, як симуляція динаміки твердих тіл (Rigid Body), рідин (Liquid simulation) та м'яких тіл (Soft body), редагування матеріалів і геометрії за принципом вузлів (Nodes), велику кількість легко доступних розширень, написаних мовою Python.

Blender був розроблений як робочий інструмент голландською анімаційною студією NeoGeo (не має стосунку до ігрової консолі Neo-Geo), провідною студією цього напрямку в Європі. В червні 1998 року автор Blender-а, Тон Розендал (Ton Roosendaal), заснував компанію Not a Number (NaN) з метою подальшого розвитку та супроводу Blender. Програма розповсюджувалася за принципом умовно-безкоштовного програмного

забезпечення (англ. shareware), на відміну від інших програм для роботи з тривимірною графікою того часу. В 1999 вона привернула увагу конференції SIGGRAPH, на якій була широко розрекламована. NaN завдяки цьому найняла близько 50 співробітників і влітку 2000 року видала Blender версії 2.0. На кінець 2000 року на сайті NaN зареєструвалося 250 тис. користувачів Blender.

У квітні 2001 NaN запустила програму Blender Publisher, призначену об'єднати ринок мережевої тривимірної графіки. Проте інвестори не були зацікавлені в ній. У 2002 році компанія NaN збанкрутувала. Зусиллями Тона Розендала кредитори погодилися на зміну ліцензії розповсюдження Blender на користь GNU GPL з умовою одноразової виплати €100000. 18 липня 2002 року почалася програма зі збору спонсорських пожертвувань на покриття необхідної суми. Вже 7 вересня 2002 року було оголошено про те, що необхідна сума набрана і в планах перевести найближчим часом вихідний код і сам Blender під ліцензію GPL.

13 жовтня 2002 року, новозаснована компанія Blender Foundation представила продукт під ліцензією GNU GPL. З того часу, після версій 2.2х, Blender є проєктом з відкритим вихідним кодом і розвивається при активній підтримці Blender Foundation. Всього за один місяць версію Blender версії 2.44 завантажили 800 000 людей по всьому світу.

Версії. З того моменту, як код програми став відкритим, він зазнав не лише значних змін, але й отримав багато нових можливостей. Частині цих змін сприяла програма Summer of Code від Google, у якій Blender Foundation бере участь з 2005 року.

Autodesk Maya.

Autodesk Maya - застосунок, графічний редактор, для моделювання тривимірних об'єктів, анімації, композитингу та візуалізації (за допомогою підімкнутих систем рендерингу). В даний час є стандартом для розробки 3D графіки для кіно і телебачення. Спочатку розроблена для ОС IRIX (платформа SGI), потім була портована під ОС Linux, Microsoft Windows і Mac OS. З 2013 року, версії програми випускаються тільки для 64-бітових систем.

Назва програмного забезпечення Maya пояснюється санскритським словом मया māyā (майа), яке означає ілюзія.

Maya стала результатом суміщення трьох програмних продуктів: Wavefront The Advanced Visualizer (Каліфорнія, США), Thomson Digital Image (TDI) Explore (Франція) і Alias Power Animator (Торонто, Канада). У 1993 Wavefront купила TDI, потім в 1995 компанія Silicon Graphics Incorporated (SGI) купила обидві компанії Alias і Wavefront. Об'єднана компанія стала називатися Alias-Wavefront, яка офіційно розпочала розробляти продукт Maya, що був продемонстрований через два роки діяльності в 1998 для OS IRIX та згодом портована для операційних систем Microsoft Windows, Linux і Mac OS X. Пізніше Alias | Wavefront була перейменована в Alias. У 2003 році Alias була продана SGI приватній інвестиційній фірмі Accel-KKR. У жовтні 2005 Alias була знову перепродана, цього разу компанії Autodesk. 10 січня 2006 Autodesk закінчив злиття і тепер Alias Maya відома як Autodesk Maya. Представники компанії в різних інтерв'ю підтвердили, що не будуть зливати Maya і 3ds MAX в один продукт.

Для платформи IRIX останньою версією стала 6.5, у зв'язку з зменшенням популярності цієї ОС.

З 1998 року Maya існувала в трьох версіях: Maya Unlimited - найповніший і найдорожчий пакет, який містив розширення Hair, Fur, Maya Muscule, Fluid Effects, Cloth і деякі інші. Maya Complete - базова версія пакету, в якій присутні повноцінні блоки моделювання та анімації, але відсутні модулі фізичної симуляції. Maya Personal Learning Edition - безкоштовний пакет для некомерційного використання. В якій є функціональні обмеження на розмір зображення для рендерингу, а також позначення їх водяними знаками.

У вересні 2007 року, була випущена нова версія, що отримала ім'я Maya 2008.

На виставці SIGGRAPH 2009 компанія Autodesk представила нову версію свого 3D-редактора Autodesk Maya 2010, починаючи з якого, розробники відмовилися від поділу програми на Maya Complete і Maya

Unlimited - тепер вони запропонували одне рішення під назвою - Maya 2010. Maya 2010 містить всі можливості Maya Unlimited 2009 і Maya Complete 2009, включаючи Maya Nucleus Unified Simulation Framework, Maya nCloth, Maya nParticles, Maya Fluid Effects, Maya Hair, Maya Fur. У новій версії представлена нова система композитингу Maya Composite, заснована на програмі Autodesk Toxix, яка більше не буде доступна у вигляді окремого додатка. Крім цього, в Maya 2010 включена система Autodesk MatchMover, менеджер для складання завдань мережевої візуалізації Autodesk Backburner, п'ять вузлів візуалізації для пакетного рендерингу засобами mental ray.

Починаючи з версії Maya 2014 підтримуються тільки 64-х бітні операційні системи.

2014 року Autodesk представила ще один програмний продукт - Maya LT 2015. В основі це обрізана по функціоналу версія Maya 2015 акцентована на інструментах, які використовують ігрові розробники. Має вбудований Turtle рендер заточений для записання текстурних карт.

Через припинення розробки програмного продукту Autodesk Softimage в 2014 році та обіцянку компанії, що вона не залишить своїх клієнтів без звичного і любимого інструментарію, ряд можливостей почали переносити в Maya, які можна знайти починаючи з версії 2016.

У Maya вбудований потужний інтерпретатор крос-платформенної мови: Maya Embedded Language (MEL), дуже схожа на Tcl, що використовувалася на перших стадіях розробки програми, яка викликала багато дискусій тому, що на той час більшість студій візуальних ефектів використовувало зв'язку Perl-Tcl. Проте у першій готовій версії продукту розробники змінили Tcl на власну внутрішню скриптову мову MEL, яка стала сполучною ланкою між користувачем і ядром програми і в результаті, як виявилось це стало правильною інновацією. Вона стала не просто скриптовою мовою - це спосіб створення нової функціональності та інтерфейсу на базі Maya, бо велика частина оточення Maya і супутніх інструментів написана на цій же мові, що відкриває необмежені можливості. Зокрема, користувач може записати свої дії

як скрипт на MEL, з якого можна швидко зробити зручний макрос. Так аніматори можуть додавати функціональність до Maya навіть не володіючи мовами C або C++, залишаючи при необхідності таку можливість. Для написання розширень на мові C++ є детально документованій C++ API. Також для розробників тепер є можливість написання доповнень мовою Python, яка у версії Maya 8.5 була також включена, як скриптова мова (у Maya 2008 включена версія Python 2.5).

Файли проектів, включаючи всі дані про геометрію та анімацію, зберігаються як послідовності операцій MEL. Ці файли можуть бути збережені в текстовому файлі (.ma - Maya ASCII), який може бути відредагований в будь-якому текстовому редакторі. Мова MEL не прив'язана до платформи, тому код, написаний на ньому, буде виконуватися в будь-якій операційній системі. Це забезпечує неперевершений рівень гнучкості при роботі із зовнішніми інструментами.

Візуалізація в Maya реалізована трьома вбудованими механізмами: Maya Software, Maya Hardware, Maya Vector Render, а також рендером, mental ray Standalone, що встановлюється окремо. Існує й ряд візуалізаторів від сторонніх розробників, в яких включена підтримка Maya.

На основі мови MEL та Python написано безліч різноматематичних плагінів, які поширюються на платних або безкоштовних ліцензіях

### 3.2 Аналіз VR-середовищ для проектування програмної системи

Виконаємо аналіз найбільш відомих VR-середовищ для проектування нашої системи. А саме розглянемо багатоплатформовий інструмент (VR-середовище) – Unity, та ігровий рушій (VR-середовище) - Unreal Engine.

Unity.

Unity - багатоплатформовий інструмент для розроблення відеоігор і застосунків, і рушій, на якому вони працюють. Створені за допомогою Unity

програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

Unity - це кросплатформовий ігровий рушій. Програма-редактор Unity працює на Windows, macOS і Linux, а сам рушій може запускатися на 25 платформах.

Редактор Unity має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження гри чи застосунка прямо в редакторі. Головні вікна - це оглядач ресурсів проєкту, інспектор поточного об'єкта, вікно попереднього перегляду, оглядач сцени та оглядач ієрархії ресурсів.

Проєкт в Unity поділяється на сцени (рівні) - окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти-моделі (ландшафт, персонажі, предмети довкілля тощо), так і порожні ігрові об'єкти - ті, що не мають моделі, проте задають поведінку інших об'єктів (тригери подій, точки збереження прогресу тощо). Їх дозволяється розташовувати, обертати, масштабувати, застосовувати до них скрипти. В них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково наявний компонент Transform - він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою. Різні моделі можуть об'єднуватися в набори (ассети) для швидкого доступу до них. Наприклад, моделі споруд на спільну тему.

Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу

батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

У 2D іграх Unity переважно використовує спрайти. В 3D іграх Unity здебільшого використовує тривимірні моделі (меші), на які накладаються текстури (зумовлюють вигляд поверхні об'єктів), матеріали (зумовлюють як поверхня реагуватиме на різні фактори) та шейдери (невеликі скрипти, за яким вираховується зміна кольору кожного пікселя згідно заданих параметрів, як-от розсіяння відбитого світла). В обох видах застосовуються системи часток для відображення субстанцій, таких як рідини чи дим.

Unity підтримує стиснення текстур, міпмапінг і різні налаштування роздільності екрана для кожної платформи; забезпечує бамп-мапінг, мапінг відображень, паралакс-мапінг, затінення навколишнього світла у екранному просторі, динамічні тіні за картами тіней, рендер у текстуру та повноекранні ефекти обробки зображення, такі як зернистість, глибина чіткості, розмиття в русі, відблиски віртуальних лінз або ореол навколо джерел світла

Рендеринг зображення відбувається через віртуальну камеру огляду. В робочій області редактора ігрова сцена може розміщуватися як завгодно, а при рендерингу - так, як її видно з камери. В сцені може бути декілька камер, які рухаються за персонажем чи за вказаною траєкторією. Вигляд з камери подається в двовимірно чи тривимірно (в перспективі або ортографічно). Фон сцени, видимий через камеру, типово зображає небо, утворене скайбоксом, але може презентувати й інше довкілля.

Графічний рушій використовує DirectX (Windows), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), та спеціальне власне API для Wii. Також підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing.

Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic

Substance. В ігровий проєкт Unity можна імпортувати об'єкти цих програм та виконувати налаштування за допомогою графічного інтерфейсу.

Для написання шейдерів використовується ShaderLab, що підтримує шейдерні програми написані на GLSL або Cg. Шейдер може включати декілька варіантів реалізації, що дозволяє Unity визначати найкращий варіант для конкретної відеокarti. Unity також має вбудовану підтримку фізичного рушія Nvidia PhysX (колишнього Ageia), підтримку симуляції одягу в системі реального часу на довільній та прив'язаній полігональній сітці (починаючи з Unity 3.0), підтримку системи ray casts та шарів зіткнення.

### Unreal Engine.

Unreal Engine - ігровий рушій, який розробляється та підтримується компанією Epic Games.

Перша гра, створена на цьому рушії - Unreal, з'явилася в 1998 році. Відтоді різні версії цього ігрового рушія використали у більш ніж сотні ігор, серед яких: Deus Ex, Lineage II, Thief: Deadly Shadows, Postal 2, серія ігор Brothers in Arms, серія ігор Splinter Cell, Tom Clancy's Rainbow Six, а також у відомих ігрових серіях Unreal та Unreal Tournament від самої Epic Games. Пристосований насамперед для шутерів від першої особи, рушій використовувався і для створення ігор інших жанрів.

Написаний мовою програмування C++, рушій дає змогу створювати ігри для більшості операційних систем і платформ таких як Microsoft Windows, Linux, Mac OS і Mac OS X, і консолей: Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast і Nintendo GameCube. У грудні 2009 Марк Рейн продемонстрував як працює рушій Unreal Engine 3 на iPod Touch і iPhone 3GS. А у березні 2010 робота рушія була продемонстрована на комунікаторі Palm Pre, що базується на мобільній платформі webOS.

Для спрощення портування, рушій використовує модульну систему залежних компонентів: підтримує різні системи рендерингу (Direct3D, OpenGL, Pixomatic; раніше підтримувалися Glide API, S3 Metal, PowerVR SGL), відтворення звуку (EAX, OpenAL, DirectSound3D, раніше

підтримувалася A3D), засоби голосового відтворення тексту, розпізнавання мовлення (тільки для Xbox 360, PlayStation 3, Nintendo Wii та Microsoft Windows, також планувалося для Linux і Mac), модулі для роботи з мережею й підтримання різних пристроїв вводу.

Для багатокористувацького режиму гри підтримуються технології Windows Live, Xbox Live і GameSpy, що дає можливість під'єднати до 64 гравців (клієнтів) одночасно. Попри те, що офіційно засоби розробки не містять у собі підтримки великої кількості клієнтів на одному сервері, рушій використовувався для створення MMORPG-ігор.

Усі елементи ігрового рушія представлені у вигляді об'єктів, що мають набір характеристик і клас, який визначає доступні функції. Так само, будь-який клас є «дочірнім» класом об'єкта.

Більшість відеоігор на Unreal Engine виконані в жанрі 3D-шутера або бойовика, однак серед випущених продуктів є ігри іншого жанру: аркади, рольові ігри, файтинги й різного роду симулятори. Крім того, рушій адаптується для обробки тривимірної графіки в кінематографі й освітніх цілях.

### 3.3 Обґрунтування вибору інструментарію для проектування системи

На підставі проведеного аналізу програмних пакетів та VR-середовищ - для реалізації нашої розробки було вибрано два основні інструменти: Unity і Unreal Engine. Ці платформи є лідерами в індустрії розробки інтерактивних додатків, зокрема віртуальних середовищ для навчання та ігор. Обидва інструменти мають гнучкість у роботі з різними типами VR-шоломів і контролерів, що дозволяє адаптувати додаток для широкого спектра користувачів.

Unity є найбільш популярною платформою для створення VR-додатків завдяки простоті використання, великій документації та підтримці великої кількості платформ. Вона надає можливість використовувати C# для написання сценаріїв і має потужні інструменти для розробки інтерактивних

середовищ. Крім того, Unity має безліч готових бібліотек та активів, які допомагають прискорити процес розробки. Зокрема, для навчальних додатків можна використовувати готові компоненти для створення інтерфейсів, інтерактивних елементів та анімацій.

Unreal Engine є більш потужним інструментом для створення графічно складних середовищ завдяки своїм можливостям рендерингу. Unreal Engine дозволяє створювати реалістичні візуалізації з високим рівнем деталізації, що важливо при створенні складних 3D-моделей, таких як комп'ютерні компоненти та їх взаємодія.

Для створення 3D-моделей компонентів комп'ютера використовувались програми Blender і Autodesk Maya. Ці інструменти дозволяють створювати детальні та реалістичні моделі, що легко інтегруються в VR-середовище. Blender є безкоштовним і відкритим інструментом, що робить його доступним для широкого кола розробників.

## РОЗДІЛ 4

### ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРАКТИВНОГО ВІРТУАЛЬНОГО СЕРЕДОВИЩА

#### 4.1. Опис підготовчого етапу проектування системи

Процес розробки включає моделювання компонентів, їх інтеграцію в Unity або Unreal Engine, програмування логіки взаємодії з користувачем та тестування всіх етапів складання комп'ютера в віртуальному середовищі.

Підготовчий етап.

Перш ніж розпочати наш проєкт, необхідно забезпечити доступ до програмного забезпечення, яке стане основою для виконання всіх наступних завдань. Unity використовується для створення інтерактивних середовищ і дозволяє реалізовувати широкий спектр функцій. Для цього нам знадобиться Unity Hub — спеціальний лаунчер, який спрощує управління версіями Unity і проєктами. Наступні кроки детально пояснюють, як завантажити та встановити Unity, забезпечивши готовність до роботи.

Крім того, важливо заздалегідь впевнитися, що ваш комп'ютер відповідає мінімальним системним вимогам для роботи з Unity. Це допоможе уникнути можливих проблем у процесі розробки. Рекомендується також мати стабільне інтернет-з'єднання, оскільки під час встановлення будуть завантажуватися додаткові файли та компоненти.

Підготовчі заходи та подальші дії зосереджені на створенні середовища, яке дозволить максимально ефективно виконувати завдання роботи.

#### 4.2 Проектування інтерактивного віртуального середовища (покроково)

Розглянемо покроково проектування інтерактивного віртуального середовища:

Крок 1: Перехід на офіційний вебсайт Unity

Відкрийте ваш веббраузер, наприклад Google Chrome, Mozilla Firefox або Microsoft Edge. У адресному рядку введіть адресу офіційного сайту Unity: <https://unity.com/>. Це перший і важливий крок, оскільки завантаження програмного забезпечення з неофіційних джерел може бути небезпечним.

На головній сторінці сайту ви побачите велику помітну кнопку "Download". Вона зазвичай розташована в центрі сторінки і має яскравий дизайн для зручності користувачів. Цей етап продемонстровано на рисунку 4.1.

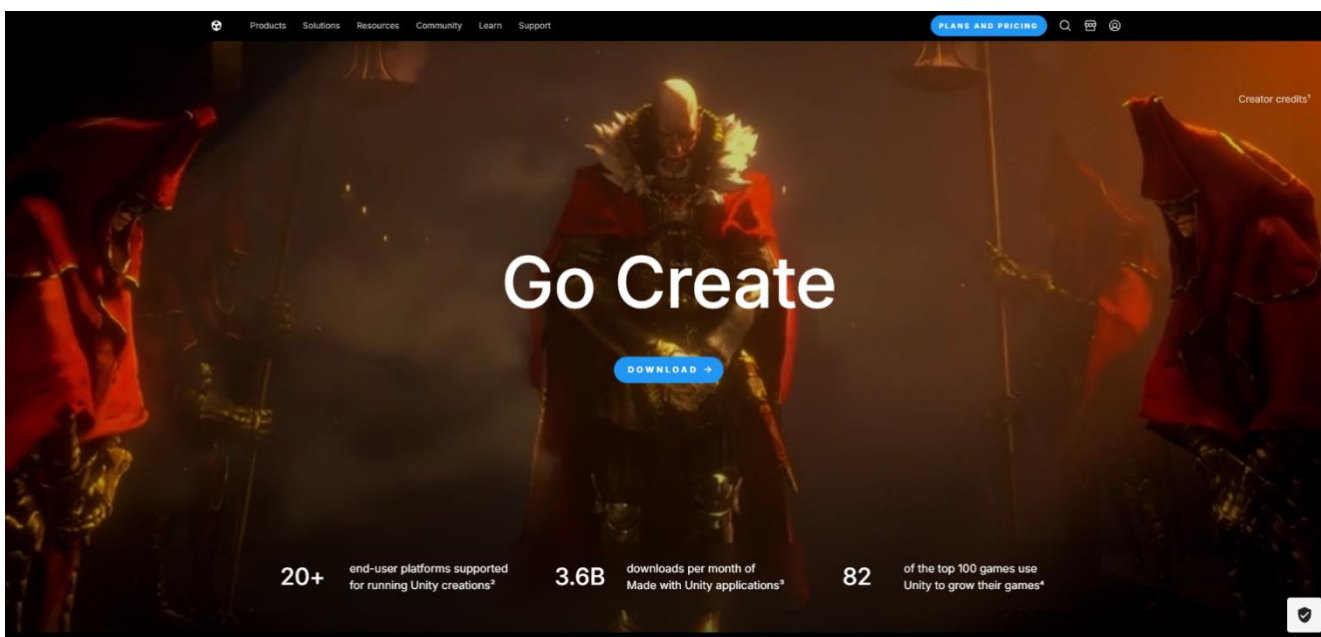


Рисунок 4.1 - Головна сторінка сайту Unity з кнопкою завантаження.

## Крок 2: Завантаження Unity Hub.

Unity Hub є централізованим інструментом для встановлення та управління версіями Unity. Він забезпечує простоту та зручність у роботі з різними проєктами й версіями.

1. Після натискання кнопки "Download" ви будете автоматично перенаправлені на сторінку завантаження Unity Hub. Цей перехід важливий, оскільки Unity Hub є необхідною програмою для встановлення та роботи з Unity (рисунок 4.2).

2. На сторінці завантаження обрати потрібну версію для вашої операційної системи. Натиснути кнопку "Download for Windows" (або для іншої ОС, якщо ви використовуєте Mac або Linux).

3. Почнеться завантаження файлу установника Unity Hub, розмір якого зазвичай не перевищує кілька сотень мегабайтів (рис. 4.2).

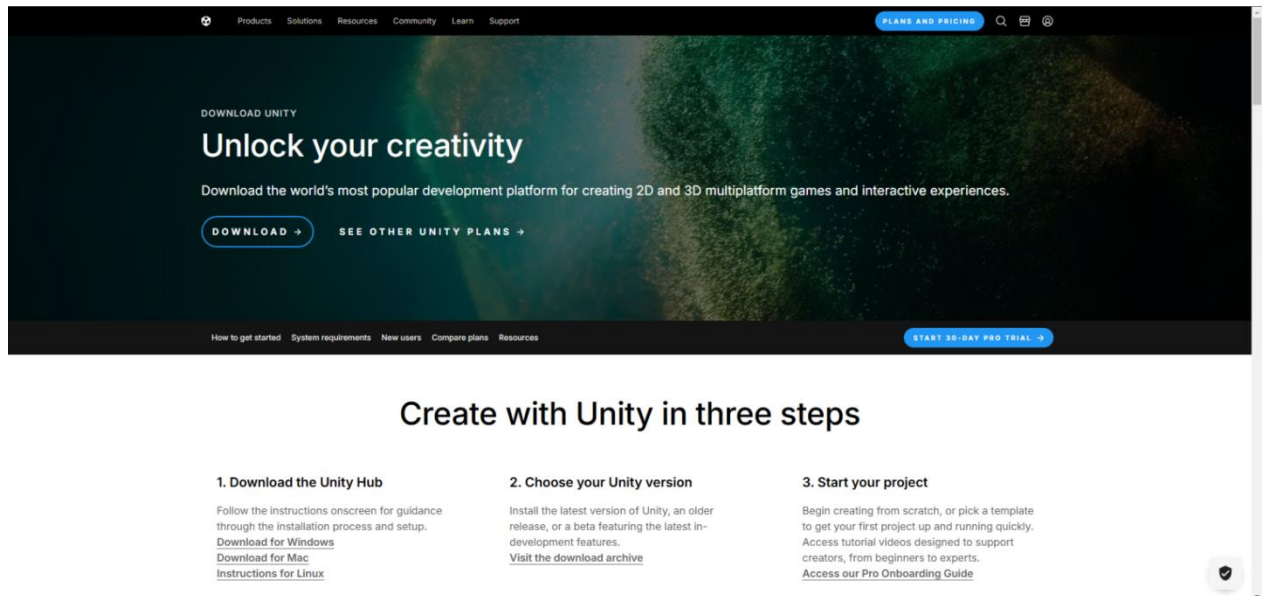


Рисунок 4.2 - Сторінка завантаження Unity Hub.

### Крок 3: Встановлення Unity Hub

1. Після завершення завантаження знайдіти файл установника у папці "Завантаження" або в іншому місці, залежно від налаштувань вашого браузера.

2. Запустити файл установника подвійним натисканням лівої кнопки миші.

3. Дотримуватись вказівок майстра встановлення. У цьому процесі вам буде запропоновано:

- прочитати та прийняти умови ліцензійної угоди. Цей крок дуже важливий, оскільки ви підтверджуєте дотримання правил використання програми;

- обрати місце встановлення програми на вашому комп'ютері. Рекомендується залишати стандартний шлях, якщо у вас немає особливих причин його змінювати.

#### Крок 4: Реєстрація та авторизація

1. Після успішного встановлення запустити Unity Hub, двічі натиснувши на його значок на робочому столі або в меню "Пуск".

2. У вікні програми з'явиться кнопка "Sign In". Натиснути її для входу в обліковий запис Unity або створення нового, якщо у вас його немає. Реєстрація є необхідною для отримання доступу до всіх функцій Unity (рис. 4.3.).

3. Процес реєстрації включає:

- Введення електронної пошти.
- Створення пароля.
- Підтвердження вашого облікового запису через електронний лист.

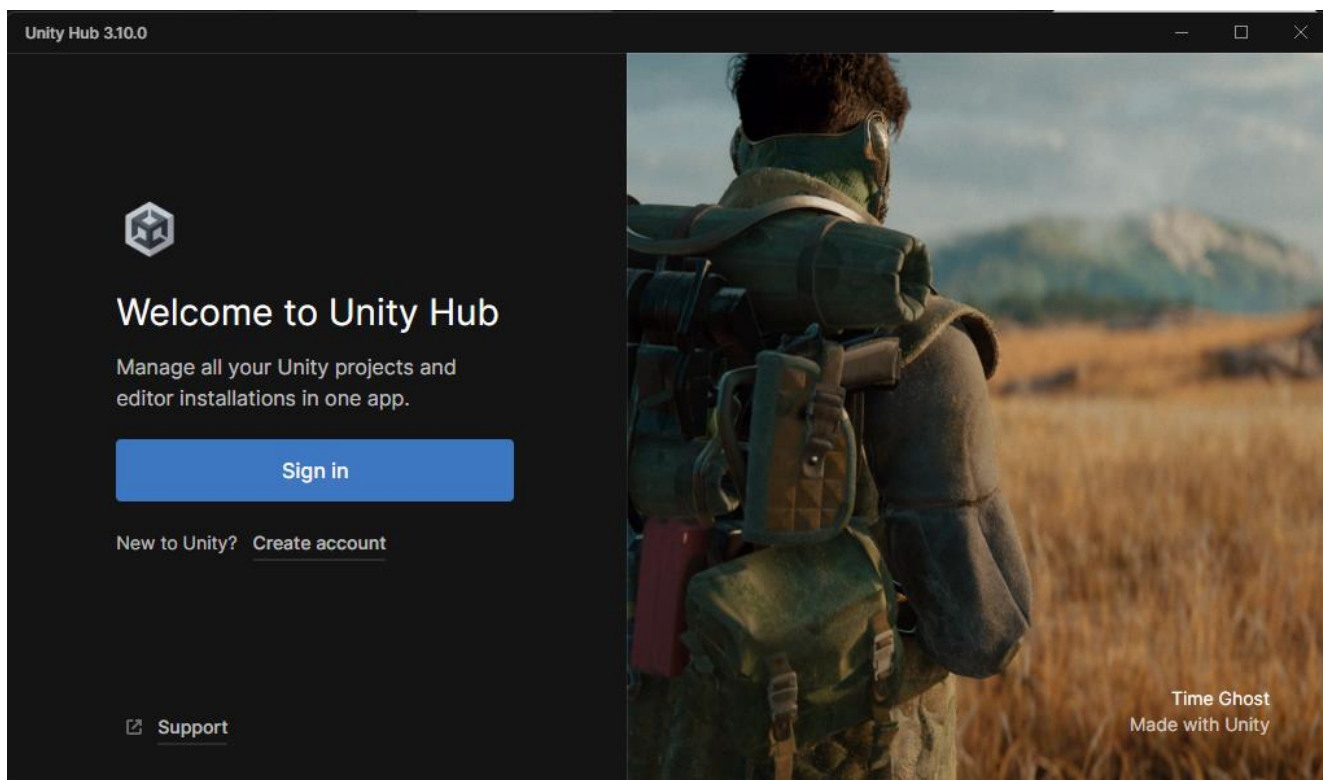


Рисунок 4.3 - Вікно авторизації в Unity Hub.

#### Крок 5: Вибір та встановлення версії Unity

1. У Unity Hub перейти до вкладки "Installs". Це місце, де ви можете встановлювати нові версії Unity або управляти вже встановленими.

2. Натиснути "Add" для додавання нової версії Unity (рис. 4.4). З'явиться список доступних версій, серед яких рекомендується обрати версію LTS (Long-Term Support), яка забезпечує стабільну роботу.

3. У процесі встановлення вам буде запропоновано обрати додаткові модулі (наприклад, для розробки під Android чи WebGL). Оберати ті, які необхідні для вашого проєкту.

4. Натиснути "Next" та дочекатися завершення встановлення. Цей процес може зайняти певний час, залежно від швидкості вашого інтернет-з'єднання.

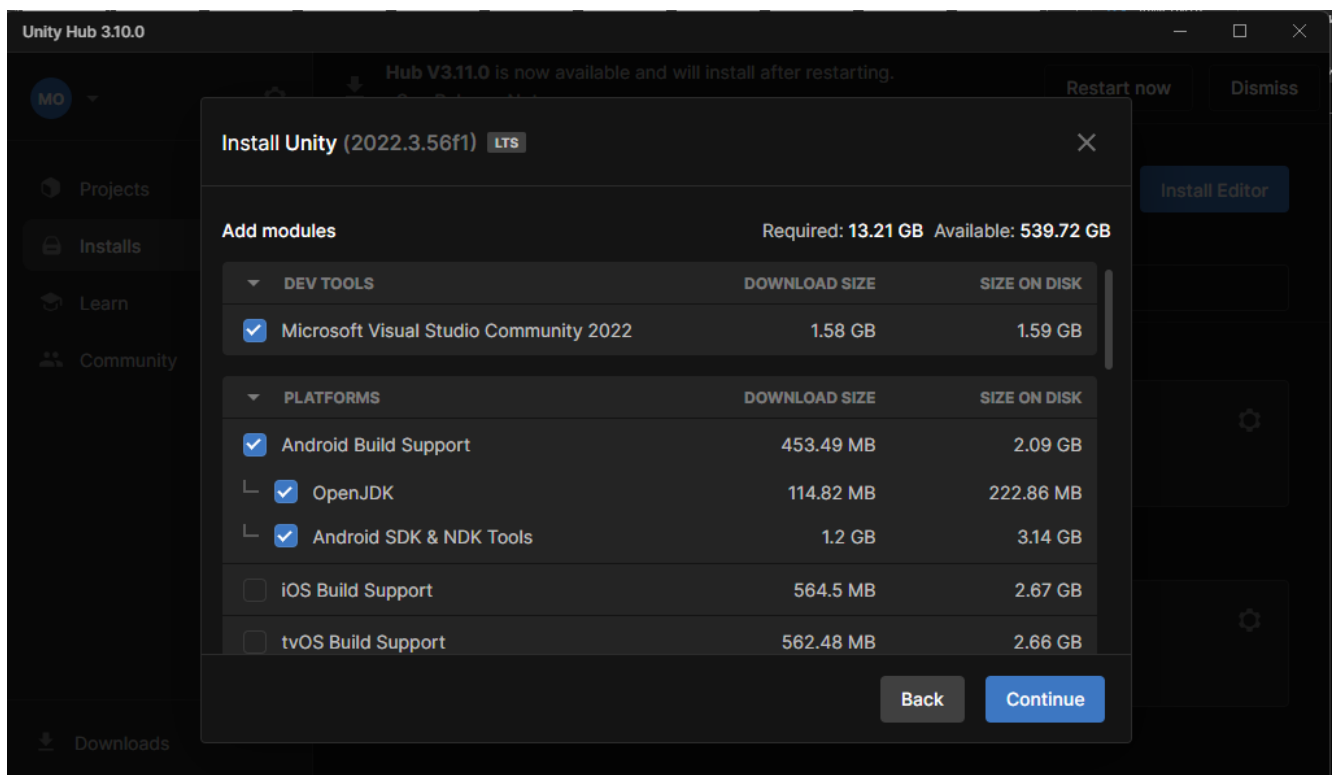


Рисунок 4.4 - Вибір версії Unity для встановлення.

#### Крок 6: Створення нового проєкту

1. Після встановлення Unity перейти до вкладки "Projects" у Unity Hub.  
 2. Натиснути "New Project" (рис 4.5) для створення нового проєкту. У цьому вікні ви зможете:

- обрати тип шаблону (наприклад, 2D, 3D);
- вказати ім'я проєкту;

- обрати директорію для його збереження.

3. Після введення всіх даних натисніть "Create" для створення проєкту. Unity автоматично відкриє ваш новий проєкт, і ви зможете розпочати роботу.

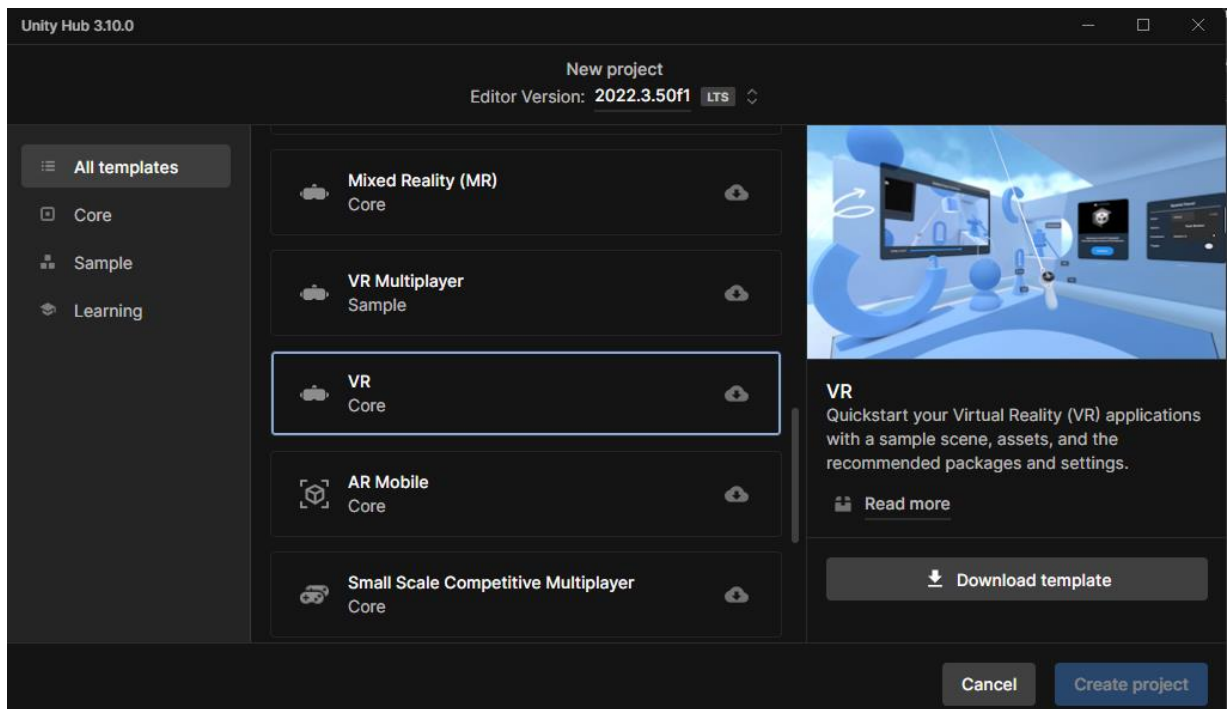


Рисунок 4.5 - Створення нового проєкту в Unity.

#### 4.3 Завантаження та встановлення ассетів у Unity

Для реалізації функціоналу проєкту необхідно інтегрувати два ключові ассети: Auto Hand - VR Interaction та Ultimate PC Computer Pack. Ці ассети забезпечують основу для взаємодії у віртуальній реальності та візуалізацію компонентів комп'ютера.

Крок 1: Вхід до Unity Asset Store

1. Відкрити Unity Hub та запустити ваш проєкт (рис. 4.6).
2. У верхньому лівому куті Unity знайти меню "Window" і вибрати "Asset Store".
3. Unity автоматично відкриє ваш браузер із сайтом [Unity Asset Store](https://unity.com/asset-store).
4. Авторизуватися своєму акаунті Unity, якщо це ще не зроблено.

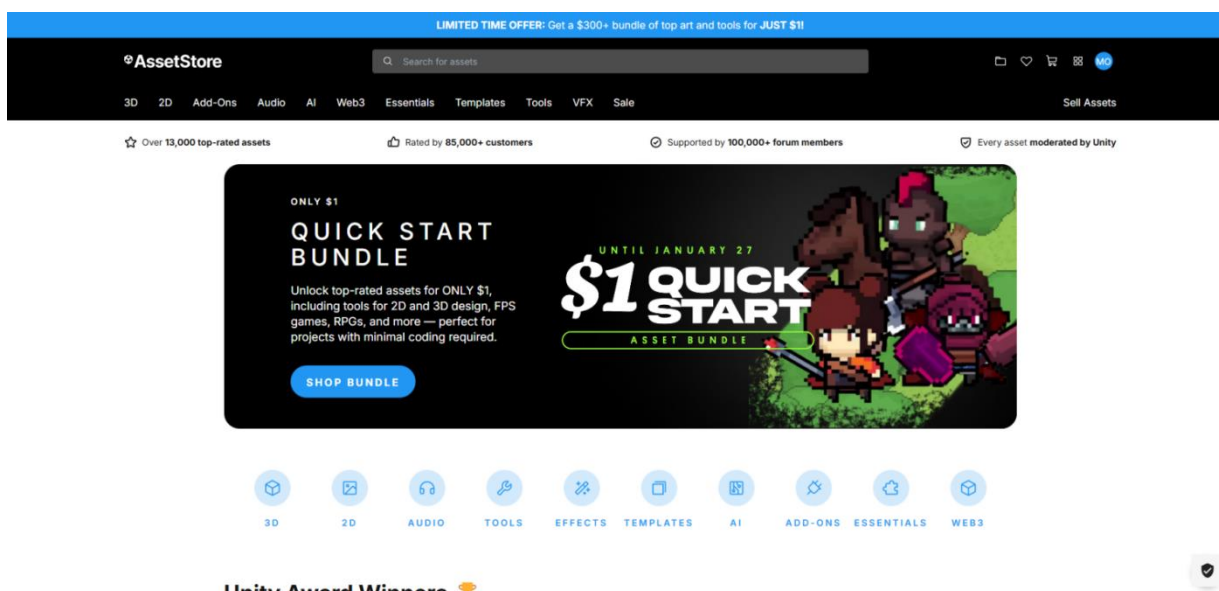


Рисунок 4.6 - Інтерфейс Unity Asset Store у браузері.

## Крок 2: Пошук потрібних ассетів

1. У пошуковому рядку Unity Asset Store ввести назву першого ассету: Auto Hand - VR Interaction (рис. 4.7).
2. Вибрати ассет у результатах пошуку, щоб перейти на його сторінку.
3. Натиснути "Add to My Assets".
4. Повторити ті ж дії для ассету Ultimate PC Computer Pack.

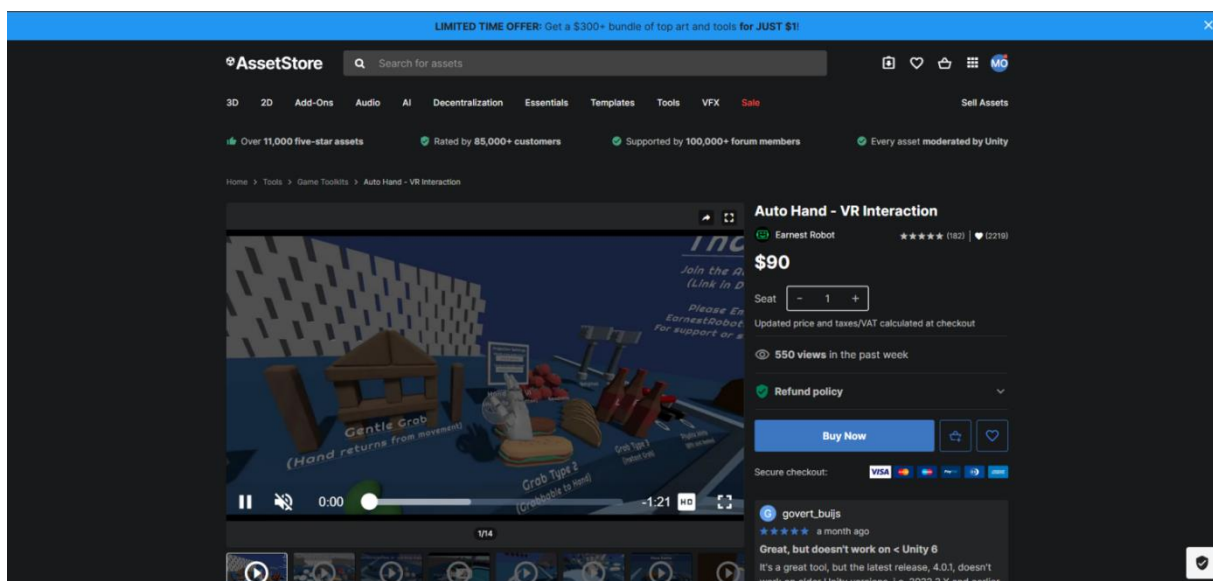


Рисунок 4.7 - Сторінка ассету Auto Hand - VR Interaction.

### Крок 3: Завантаження та імпорт ассетів

1. Повернутися до Unity Editor.
2. У верхньому лівому куті знайти меню "Window" і вибрати "Package Manager" (рис. 4.8).
3. У вкладці "My Assets" знайти завантажені ассети. Спочатку натисніть "Download", а потім – "Import" для кожного ассету.
4. У діалоговому вікні імпорту переконатися, що всі файли вибрані, і натисніть "Import".

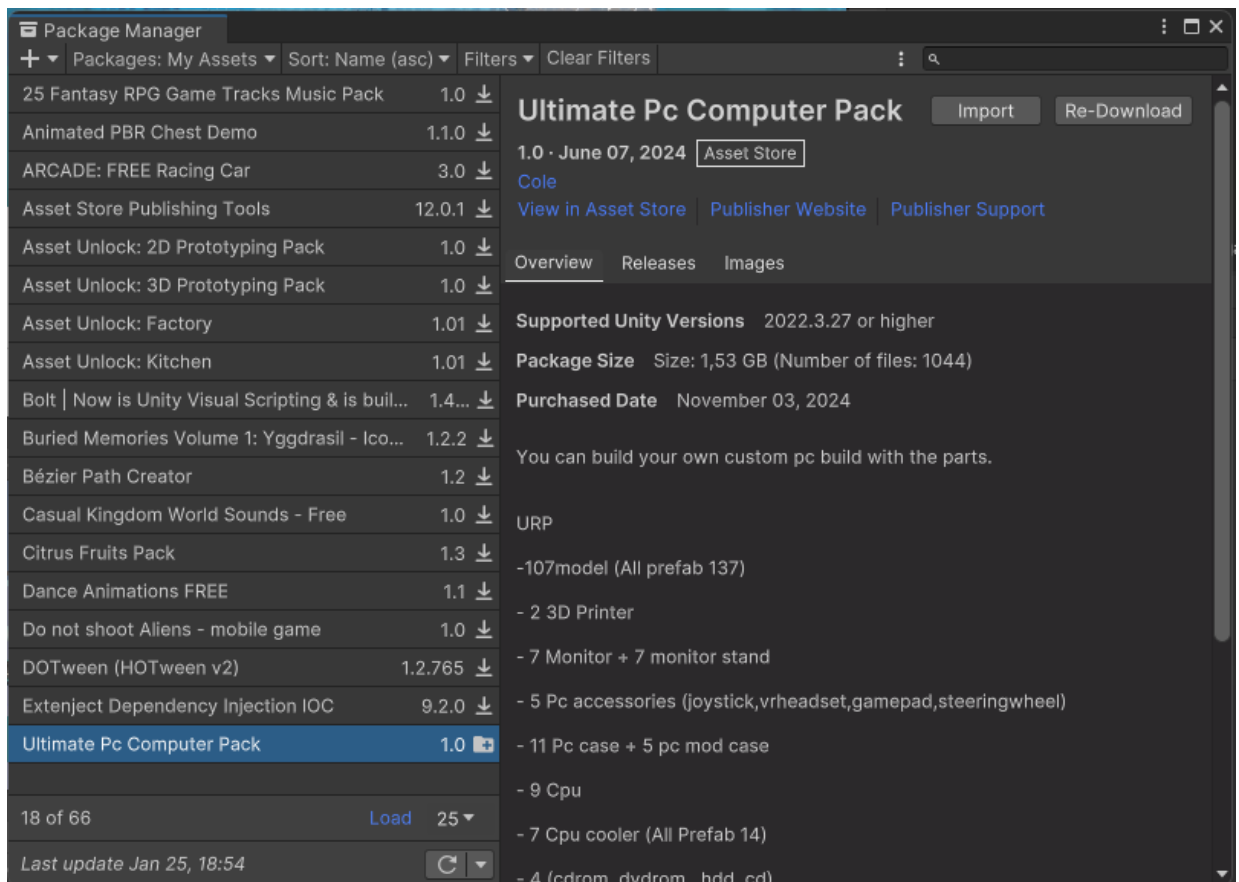


Рисунок 4.8 - Імпорт ассетів у Unity

## 4.4 Завантаження Meta Quest Link

Для забезпечення коректної роботи гарнітури Meta Quest необхідно встановити драйвери Meta Quest Link.

### Крок 1: Завантаження Meta Quest Link

1. Відкрити офіційний сайт [Meta Quest](#) і перейти у розділ "Support" → "Quest Link" (рис. 4.9).
2. Завантажити інсталяційний файл, відповідний вашій операційній системі.

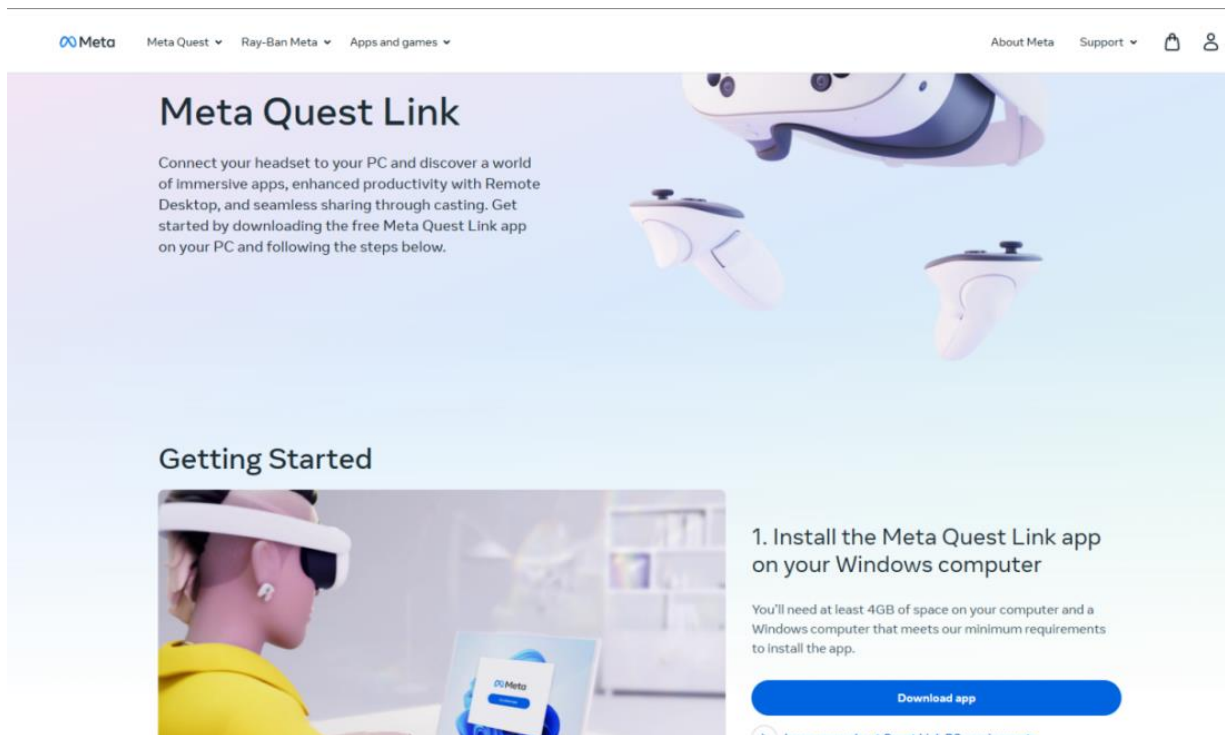


Рисунок 4.9 - Сторінка завантаження Meta Quest Link.

## Крок 2: Встановлення драйверів

1. Запустити інсталятор і дотримуйтесь інструкцій майстра установки.
2. Після завершення перевірити підключення гарнітури до комп'ютера через USB-кабель або бездротове з'єднання (рис. 4.10).

Примітка: Для бездротового підключення ваш ПК і гарнітура мають бути у тій самій Wi-Fi мережі.

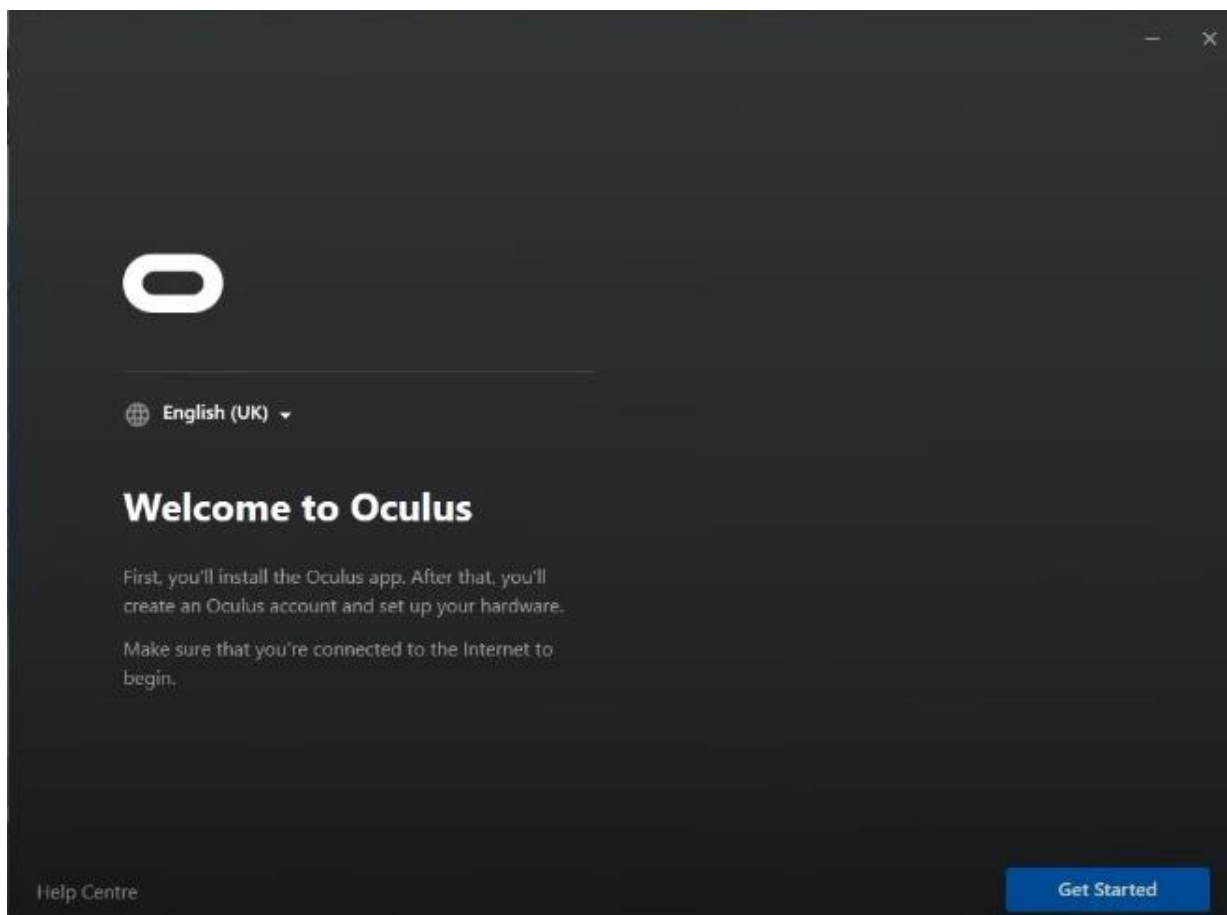


Рисунок 4.10 - Встановлення драйверів Meta Quest Link.

#### 4.5 Налаштування проєкту для роботи з VR

Після встановлення драйверів і підключення гарнітури необхідно налаштувати проєкт для роботи з Meta Quest. Це включає активацію VR-плагінів і налаштування OpenXR.

Крок 1: Увімкнення підтримки VR

1. У верхньому лівому куті Unity знайти меню "Edit" і вибрати "Project Settings".
2. У лівій частині вікна налаштувань знайти розділ "XR Plug-in Management".
3. Натиснути "Install XR Plug-in Management", щоб додати підтримку VR (рис. 4.11).
4. У списку плагінів активуйте OpenXR.



Рисунок 4.11 - Увімкнення XR Plug-in Management у Unity.

## Крок 2: Налаштування OpenXR

1. У "Project Settings" перейти до вкладки "OpenXR".
2. У розділі "Interaction Profiles" додати (рис. 4.12):
  - Oculus Touch Controller Profile.
  - Microsoft Mixed Reality Motion Controller Profile (за потреби).
3. Переконаватися, що опція "Initialize XR on Startup" активована.

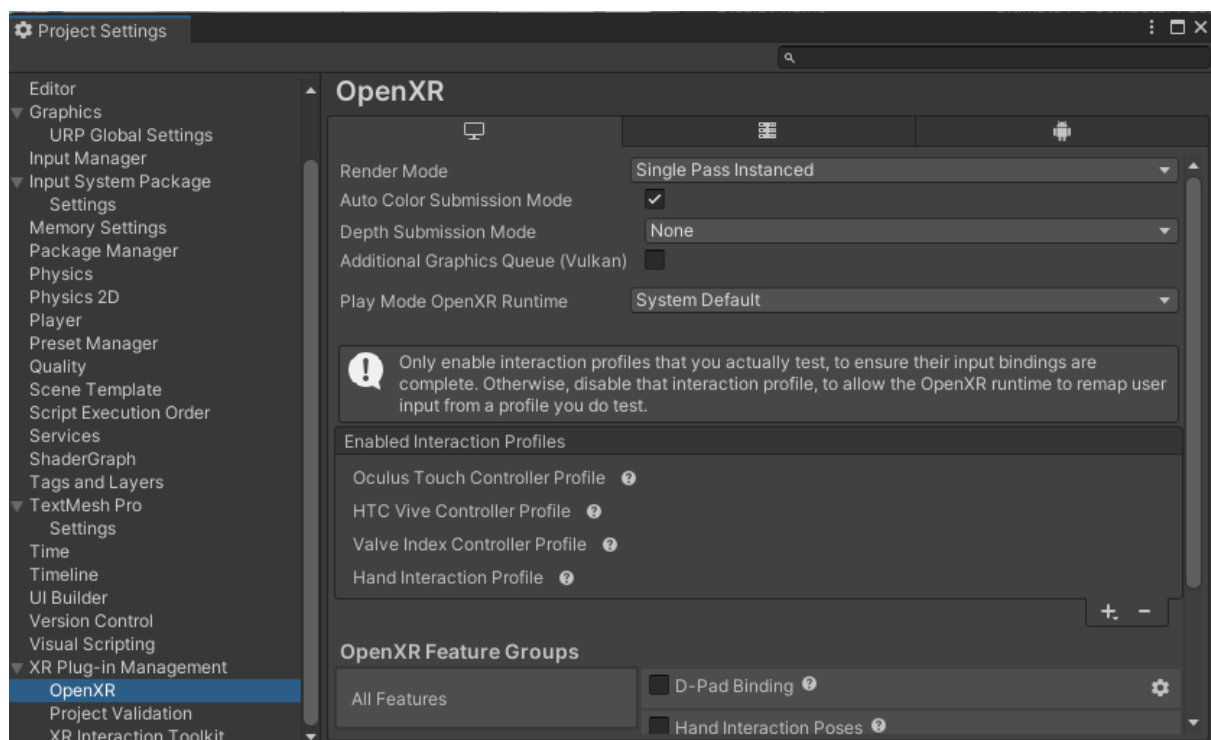


Рисунок 4.12 - Налаштування OpenXR у Unity.

### Крок 3: Підключення Meta Quest до Unity

1. Переконайтеся, що гарнітура Meta Quest увімкнена і підключена до комп'ютера.
2. Запустити гарнітуру та активуйте режим Oculus Link.
3. У Unity перейти до меню "XR Plug-in Management" у "Project Settings" і переконайтеся, що OpenXR активовано.
4. Використати інструмент *Oculus Debug Tool* (встановлюється разом із драйверами), щоб перевірити статус з'єднання гарнітури.
5. У Unity запустити режим Play Mode і переконатися, що гарнітура відображає сцену.

### 4.6 Тестування проєкту у гарнітурі Oculus

Наведемо покрокове тестування проєкту у гарнітурі Oculus:

#### Крок 1: Підготовка гарнітури та запуск режиму Oculus Link

1. Увімкнути гарнітуру Meta Quest і підключити її до комп'ютера через USB-кабель або Wi-Fi (для бездротового підключення комп'ютер і гарнітура повинні знаходитися в одній Wi-Fi мережі).
2. На гарнітурі запустити режим Oculus Link. Переконатися, що гарнітура успішно підключена до комп'ютера.

#### Крок 2: Запуск проєкту в Unity

1. У Unity відкрити головну сцену вашого проєкту.
2. У верхньому правому куті редактора натиснути кнопку "Play" для запуску режиму Play Mode.
3. Одягнути гарнітуру Meta Quest і перевірити, чи з'являється зображення сцени у віртуальній реальності (рис. 4.13).
4. Використати контролери гарнітури, щоб взаємодіяти з об'єктами сцени. Переконатися, що всі налаштування взаємодії працюють коректно.

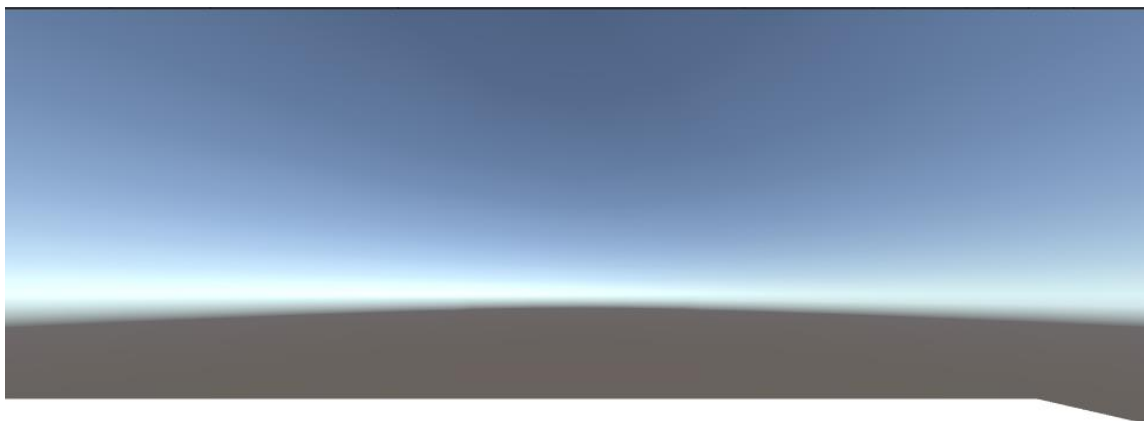


Рисунок 4.13 - Відображення сцени з гарнітури Meta Quest у режимі Play Mode

### Крок 3: Аналіз результатів тестування

1. Якщо сцена коректно відображається у гарнітурі й забезпечує необхідний рівень взаємодії, вважайте тестування успішним.

2. У разі виникнення проблем із відображенням або взаємодією, перевірте налаштування OpenXR та коректність імпорту ассетів.

## 4.7 Налаштування плагіна Auto Hand

### Крок 1: Запуск Setup Wizard

1. У Unity відкрити ваш проєкт, у якому вже імпортовано плагін Auto Hand.

2. Якщо ваш проєкт не має критичних помилок, після запуску Auto Hand ви отримаєте автоматичне спливаюче вікно Setup Wizard.

– У цьому вікні вам буде запропоновано автоматично налаштувати фізичні параметри проєкту.

Примітка: Якщо вікно Setup Wizard не з'являється автоматично, знайдіть його вручну через пошук у проєкті (рис. 4.14).

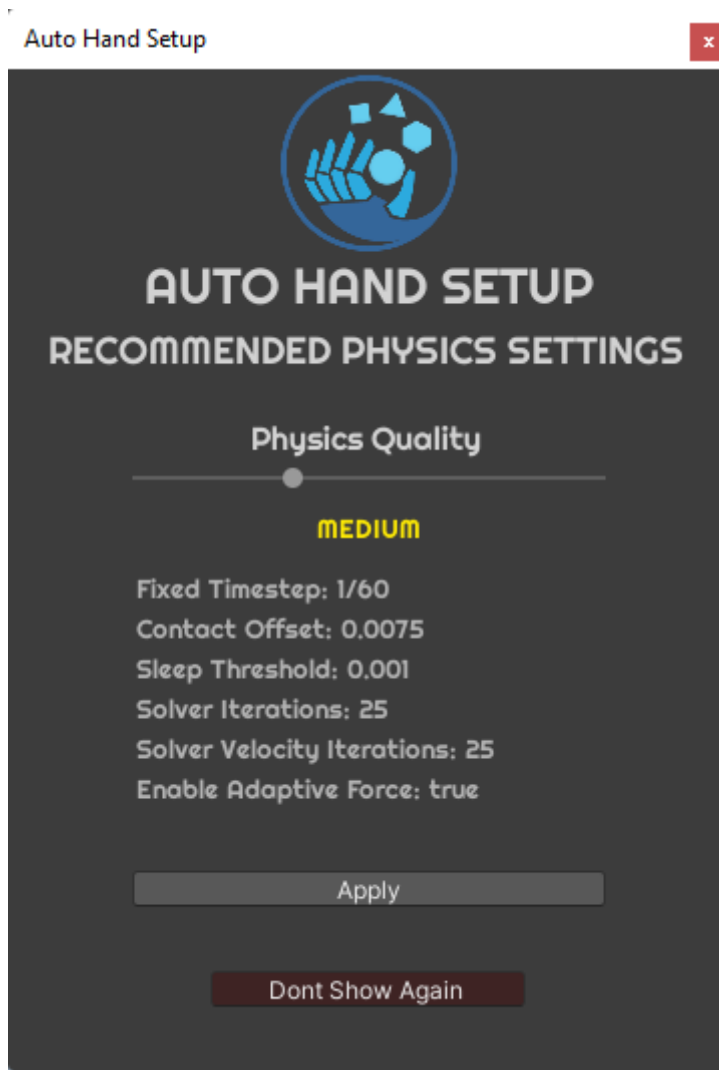


Рисунок 4.14 - Вікно Setup Wizard для Auto Hand

## Крок 2: Налаштування параметрів фізики

1. У спливаючому вікні Setup Wizard натиснути "Learn More about Physics Settings", щоб дізнатися більше про запропоновані параметри (рис. 4.15).

– Високоякісні параметри фізики забезпечують кращу взаємодію між руками і об'єктами, покращують плавність візуальних ефектів і стабільність.

2. Вибрати бажані параметри фізики із запропонованого списку та натисніть "Apply Settings".

3. Дочекайтеся завершення автоматичного налаштування.

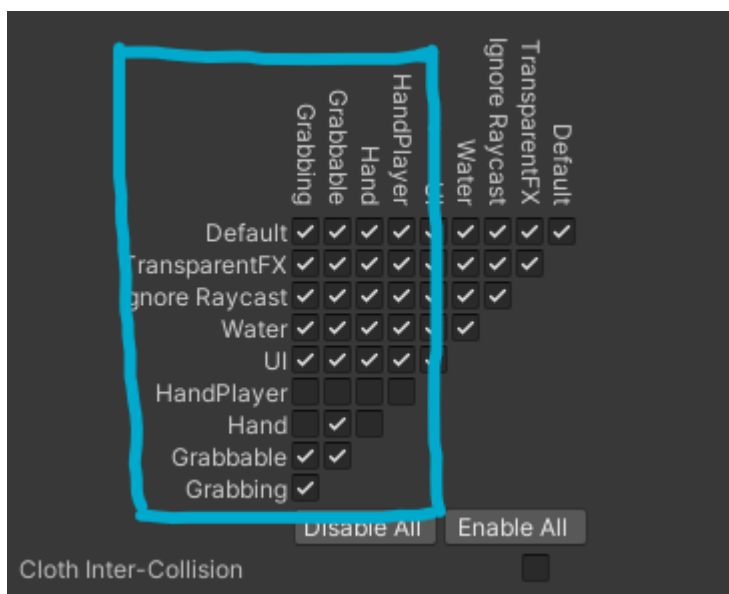


Рисунок 4.15 - Вибір рекомендованих параметрів фізики в Setup Wizard.

### *Крок 3: Налаштування для Oculus Integration*

1. У Unity відкрити "Project" (вікно проєктів) (рис. 4.16).
2. У рядку пошуку ввести "Setup" і знати папку Auto Hand.
3. Перейти до папки Packages (Setup).
4. У цій папці вибрати платформу "Oculus Integration" для налаштування
  - Переконалися, що Oculus Integration уже імпортовано у ваш проєкт.
5. Запустити процес налаштування, який автоматично адаптує плагін Auto Hand під вашу платформу.

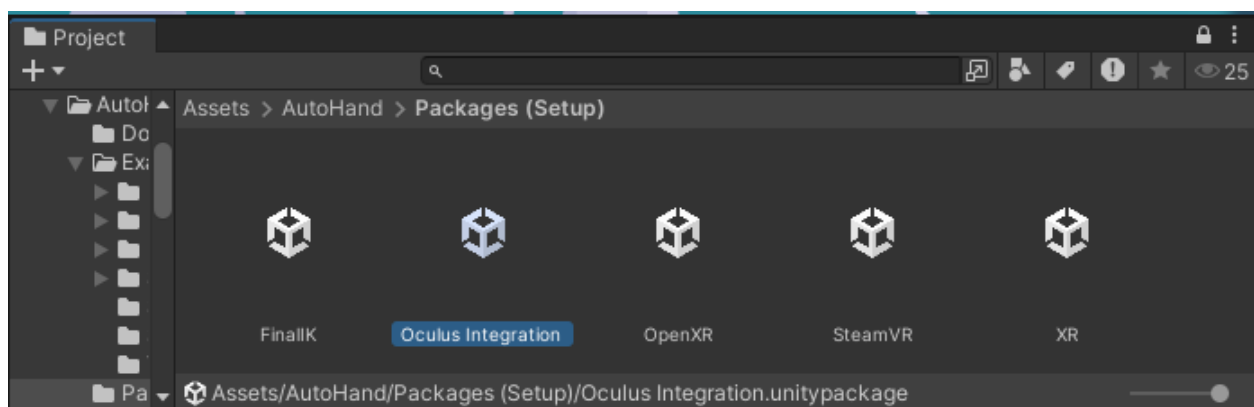


Рисунок 4.16 - Налаштування платформи Oculus Integration для Auto Hand

#### 4.8 Інтеграція користувацького інтерфейсу. Опис процесу тестування

Розробка користувацького інтерфейсу є важливою частиною створення VR-програми, оскільки саме інтерфейс забезпечує зручність використання програми для студентів. Ключовим аспектом є забезпечення інтуїтивно зрозумілої навігації віртуальним середовищем, де користувач може легко взаємодіяти з різними елементами без зайвих зусиль.

Перший етап створення інтерфейсу включає розробку основних елементів, таких як панелі інструментів, меню, кнопки і елементи взаємодії з компонентами комп'ютера. Для цього використовуються стандартні інструменти UI в Unity або Unreal Engine, які дозволяють створювати елементи інтерфейсу, які реагують на дії користувача, наприклад, натискання на кнопки або переміщення компонентів.

Другим етапом є інтеграція цих елементів у віртуальне середовище, що дозволяє студентам взаємодіяти з компонентами комп'ютера. Важливо, щоб інтерфейс не заважав основному процесу складання, тому всі елементи повинні бути чітко організовані і доступні для використання. Студенти повинні мати можливість отримати підказки та зворотний зв'язок по своїх діях, що значно покращує навчальний досвід.

Тестування користувацького інтерфейсу проводиться на різних етапах розробки, щоб переконатися в його зручності та ефективності. Важливим є також забезпечення сумісності з різними VR-шоломами та контролерами, що дозволяє адаптувати програму до різних умов використання.

Створення голови гравця з руками:

1. У Unity перейти у вкладку "Project" (вікно проєктів).
2. У рядку пошуку ввести OpenXRAutoHandPlayer. Знайдіть цей файл у результатах пошуку.
3. Знайти цей файл у результатах пошуку.

4.Перетягнути OpenXRAutoHandPlayer на сцену, щоб додати гравця до вашого проєкту (рис. 4.17).

Примітка: Ніякі додаткові налаштування для цього об'єкта виконувати не потрібно.



Рисунок 4.17 - Додавання OpenXRAutoHandPlayer на сцену

Крок 2: Додавання об'єкта для взаємодії

1. У вкладці "Project" знайти ассет з комп'ютерними комплектуючими.
2. У пошуку ввести назву об'єкта, наприклад, оперативна пам'ять (RAM).
3. Знайти об'єкт оперативної пам'яті та перетягнути його на сцену (рис. 4.18)

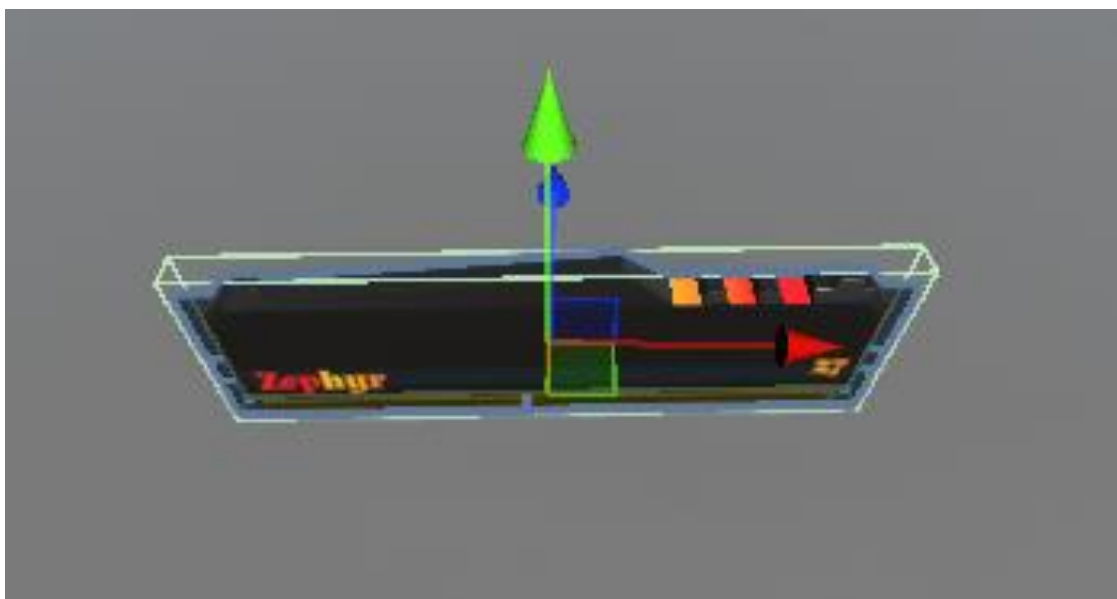


Рисунок 4.18 - Додавання оперативної пам'яті на сцену.

### Крок 3: Налаштування об'єкта для взаємодії

#### Додавання компонента Grabbable:

1. Виділити об'єкт оперативної пам'яті на сцені.
2. У вікні Inspector натиснути "Add Component".
3. Знайти та додати компонент Grabbable (рис. 4.19).
4. У налаштуваннях Grabbable активувати такі опції:
  - Parent on Grab: перетворить об'єкт на дочірній об'єкт руки під час захоплення, що зменшить затримку при русі;
  - Held No Friction: вимкне силу тертя між рукою та об'єктом для уникнення проблем із фізикою;
  - Make Children Grabbable: дозволить дочірнім об'єктам оперативної пам'яті також бути доступними для взаємодії.



Рисунок 4.19 - Налаштування компонента Grabbable для оперативної пам'яті

## 2. Додавання фізичних властивостей:

1. У вікні Inspector натиснути "Add Component".

## 2. Додати компонент Rigidbody:

– встановити масу об'єкта, наприклад, 0.1 (залежно від його реальних розмірів).

## 3. Додати компонент Box Collider:

– налаштувати розміри колайдера, щоб вони відповідали формі оперативної пам'яті.

## Крок 4: Тестування взаємодії

1. Запустити режим Play Mode у Unity (рис. 4.20).

2. Використати руки гравця (об'єкт OpenXRAutoHandPlayer) для взаємодії з оперативною пам'яттю.

## 3. Переконатися, що:

- Об'єкт захоплюється без затримок.
- Об'єкт рухається разом із рукою без фізичних аномалій.
- Дочірні елементи (якщо є) також можна захоплювати.

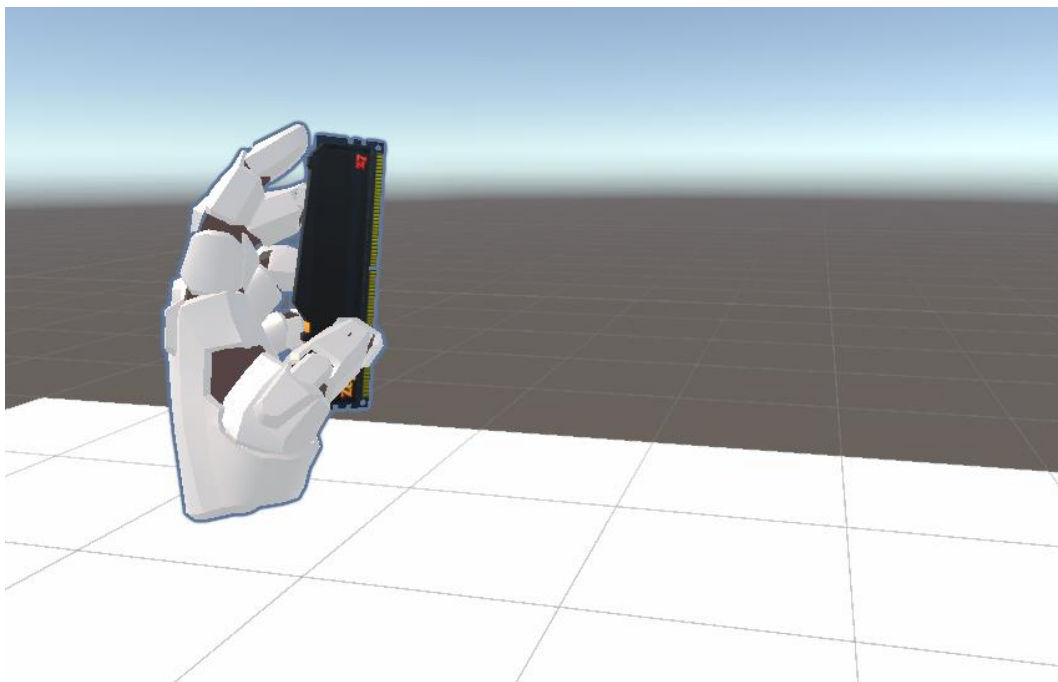


Рисунок 4.20 - Тестування захоплення оперативної пам'яті у режимі Play Mode.

#### 4.9 Налаштування компонента PlacePoint для взаємодії з об'єктами

Компонент PlacePoint у плагіні Auto Hand дозволяє створити інтерактивні точки, куди можна розміщувати об'єкти, такі як оперативна пам'ять або інші комп'ютерні комплектуючі. Налаштування цього компонента забезпечує коректну взаємодію у віртуальному середовищі без фізичних затримок і аномалій. У цьому розділі описані кроки з налаштування PlacePoint для інтеграції в проєкт із використанням VR.

##### *Крок 1: Створення PlacePoint*

1. У Unity натиснути правою кнопкою миші у вікні Hierarchy та обрати "Create Empty" для створення нового порожнього об'єкта.
2. Назвати новий об'єкт, наприклад, "RAM\_PlacePoint".
3. Переконайтеся, що об'єкт знаходиться в потрібній позиції на сцені, відповідно до вашого дизайну.

##### *Крок 2: Додавання компонента PlacePoint*

1. Виділити створений об'єкт у Hierarchy.
2. У вікні Inspector натиснути "Add Component".
3. Знайти компонент PlacePoint та додайте його до об'єкта.

##### *Крок 3: Налаштування параметрів PlacePoint*

1. У вікні Inspector налаштувати такі параметри компонента PlacePoint:
  - Offset: встановіть значення відступу, щоб визначити відстань, на якій об'єкт (наприклад, оперативна пам'ять) буде розміщений відносно точки PlacePoint;
  - Parent on Place: активувати цю опцію, щоб зробити об'єкт дочірнім для PlacePoint після розміщення. Це забезпечить синхронний рух об'єкта разом із PlacePoint у разі переміщення;
  - Force Hand Release: увімкнути цю опцію, щоб об'єкт автоматично розміщувався у PlacePoint без необхідності відпускати його вручну;

– **Disable Rigidbody on Place:** активувати цей параметр, щоб відключити Rigidbody після розміщення об'єкта. Це дозволить уникнути зайвих фізичних розрахунків.

#### Крок 4: Налаштування категорії Place Names

1. У вікні Inspector для об'єкта PlacePoint знайти розділ Place Names.
  2. Натиснути на кнопку "+", щоб додати нове ім'я для категорії.
  3. У текстовому полі ввести назву об'єкта, який буде розміщуватися у PlacePoint, наприклад, "RAM".
- переконайтися, що назва відповідає імені об'єкта, який ви хочете розміщувати (рис. 4.21);
  - скрипт автоматично перевірятиме наявність цього імені в полі, не виконуючи повної перевірки тексту.

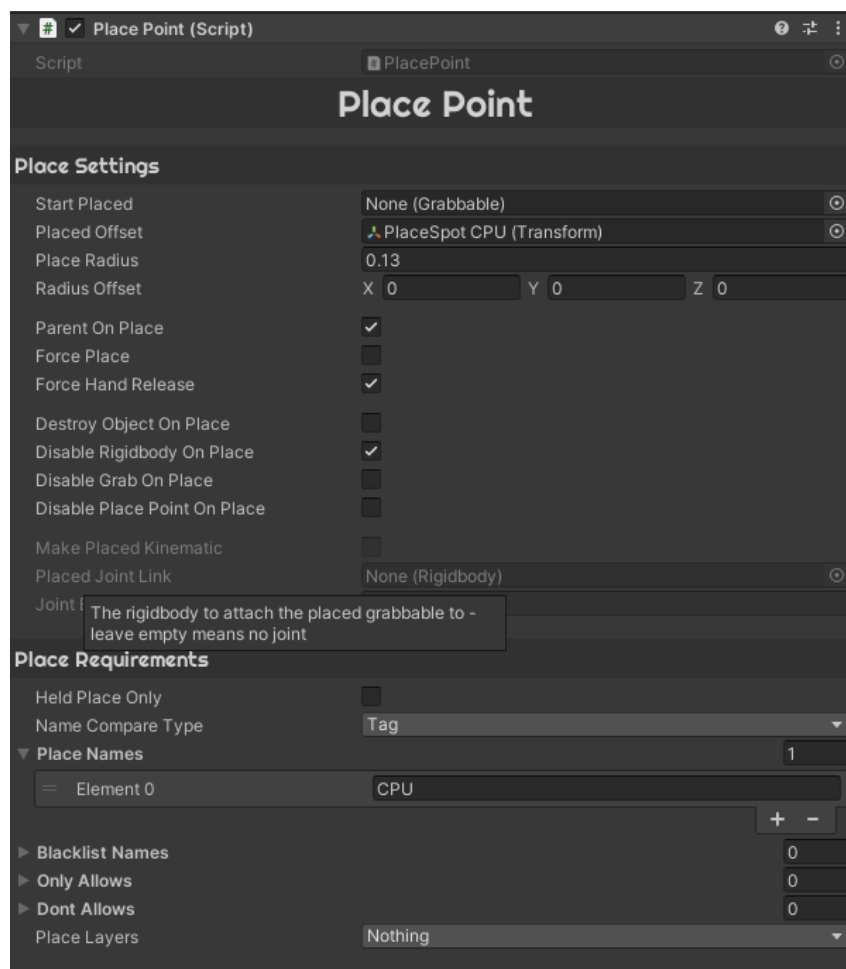


Рисунок 4.21 - Налаштування параметрів PlacePoint у Inspector

### Крок 5: Тестування PlacePoint

1. Запустити режим Play Mode у Unity.
2. Використати руки гравця, щоб взяти об'єкт (наприклад, оперативну пам'ять).
3. Навести об'єкт на PlacePoint і переконайтеся, що:
  - об'єкт автоматично розміщується при досягненні точки PlacePoint;
  - відсутні затримки або фізичні аномалії;
  - об'єкт правильно стає дочірнім до PlacePoint і слідує за ним при переміщенні.

### 4.10 Налаштування PlacePoint із материнською платою

Розглянемо налаштування PlacePoint із материнською платою:

#### Крок 1: Додавання материнської плати

1. У Unity перейти у вікно "Project".
2. Знайти об'єкт материнської плати в ассетах (наприклад, у папці з комп'ютерними комплектуючими).
3. Перетягнути материнську плату на сцену у вікно Hierarchy.
4. Розмістити об'єкт материнської плати на сцені відповідно до вашого дизайну.

#### Крок 2: Зробити PlacePoint дочірнім об'єктом материнської плати

1. У вікні Hierarchy знайти об'єкт PlacePoint, який ви створили раніше.
2. Перетягнути об'єкт PlacePoint у материнську плату, щоб зробити його дочірнім об'єктом.
3. У вікні Hierarchy переконайтеся, що PlacePoint знаходиться всередині материнської плати.

#### Крок 3: Розташування PlacePoint біля слоту оперативної пам'яті

1. Виділити PlacePoint у вікні Hierarchy.
2. У вікні Scene перемістити PlacePoint до слоту оперативної пам'яті на материнській платі.

– скористатися інструментами переміщення (Move Tool) для точного розташування.

3. За потреби використовувати параметр Offset у компоненті PlacePoint, щоб точно налаштувати відстань.

#### Крок 4: Використання функції StartPlaced

1. У вікні Inspector для об'єкта PlacePoint знайти параметр StartPlaced.
2. У полі StartPlaced вибрати об'єкт оперативної пам'яті (RAM), який потрібно відображати вже встановленим при запуску сцени.

#### Крок 5: Точне налаштування позиції об'єкта в Play Mode

1. Запустити режим Play Mode у Unity.
2. У режимі Play Mode натиснути Pause (пауза) у верхній частині екрану.
3. У вікні Scene скористатися інструментами переміщення, щоб налаштувати точне положення PlacePoint і об'єкта оперативної пам'яті.

– внести зміни в позицію об'єкта PlacePoint або скористатися параметром Offset для налаштувань.

4. Після завершення налаштувань у вікні Inspector для компонента PlacePoint натиснути на три крапки у верхньому правому куті компонента і вибрати Copy Component.

#### Крок 6: Збереження налаштувань після виходу з Play Mode

1. Вийти із режиму Play Mode, натиснувши кнопку Stop.
2. У вікні Inspector для компонента PlacePoint знову натиснути на три крапки у верхньому правому куті і вибрати Paste Component Values.
3. Перевірити, що всі зміни збереглися, і об'єкт PlacePoint знаходиться у правильному положенні.

### 4.11 Налаштування інших компонентів комп'ютера

Опишемо налаштування інших компонентів комп'ютера:

#### Крок 1: Додавання простих компонентів

1. Додай на сцену такі об'єкти: CPU, RAM, GPU, охолодження процесора (Cooling).
2. Для кожного компонента:
  - додати Grabbable;
  - активувати Parent on Grab, Held No Friction, Make Children Grabbable;
  - додати Rigidbody та Box Collider для забезпечення фізики об'єкта.
3. Розмістити об'єкти на материнській платі:
  - використати PlacePoint для кожного компонента;
  - у полі Place Names вкажіть "CPU", "RAM", "GPU", "Cooling".

#### Крок 2: Налаштування PSU

1. Додати на сцену блок живлення (PSU).
2. Налаштувати Grabbable, Rigidbody і Box Collider.
3. Додати PlacePoint на корпус системного блоку для розміщення PSU.
  - у полі Place Names вкажіть "PSU".

#### Крок 3: Налаштування корпусу системного блоку

1. Додати корпус системного блоку на сцену.
2. Додати Grabbable, Rigidbody і Box Collider.
3. Розмістити PlacePoint для материнської плати та PSU у корпусі.
  - у полі Place Names вкажіть "Motherboard", "PSU".

#### Крок 4: Налаштування кришки корпусу

1. Додати на сцену кришку корпусу системного блоку.
2. Налаштувати Grabbable, Rigidbody і Box Collider.
3. Додати PlacePoint на корпус для розміщення кришки.
  - у полі Place Names вказати "Case Cover".

#### Крок 5: Тестування інтерактивності

1. Запустити режим Play Mode у Unity.
2. Перевірити всі компоненти:
  - чи можна їх захоплювати та переміщувати (Grabbable);
  - чи правильно працюють PlacePoint.

### 3. За необхідності скоригувати позиції компонентів.

#### 4.12 Опис методики тестування. Результати застосування VR у навчальному процесі

Для оцінки ефективності використання VR-системи в навчанні складанню комп'ютерів було організовано практичне тестування на базі кількох вищих навчальних закладів, де студенти технічних спеціальностей, які ще не мали досвіду в складанні комп'ютерних систем, проходили навчання. Основною метою тестування було визначити, наскільки ефективно VR може сприяти розвитку практичних навичок і покращенню результатів навчання у порівнянні з традиційними методами.

Програма була орієнтована на студентів, що вивчають комп'ютерні науки та інженерні спеціальності, і мала на меті забезпечити інтуїтивно зрозумілий процес навчання через інтерактивні завдання та детальні інструкції. Студенти проходили через кілька етапів, починаючи від знайомства з інтерфейсом і завершуючи складним процесом складання комп'ютера з підключенням усіх компонентів.

Тестування проходило в кілька етапів. А саме:

А) Введення в середовище. Студенти ознайомлювались з інтерфейсом VR-програми, навчались правильно взаємодіяти з об'єктами, переміщати їх і виконувати початкові завдання. Цей етап був необхідний для того, щоб студенти зрозуміли основи навігації в віртуальному середовищі і почали взаємодіяти з основними елементами програми.

Б) Основний процес складання комп'ютера. Студенти проходили серію етапів складання комп'ютера, починаючи від вибору компонентів і закінчуючи їх підключенням і перевіркою працездатності. Програма автоматично надавала підказки щодо кожного етапу, забезпечуючи правильну послідовність дій і фіксуючи всі дії користувача для подальшого аналізу.

В) Оцінка результатів. Після завершення навчання студенти отримували зворотний зв'язок у вигляді оцінки правильності виконаних дій, що дозволяло їм зрозуміти помилки і покращити свої навички. Зворотний зв'язок був не лише про те, чи виконано завдання правильно, а й про деталі помилок, таких як неправильне підключення компонентів або невірне розташування елементів на материнській платі.

Результати тестування показали, що студенти, які використовували VR для навчання складанню комп'ютера, значно швидше і точніше виконували практичні завдання у порівнянні з тими, хто проходив навчання за допомогою традиційних методів. Вони змогли завершити завдання в два рази швидше і допустили менше помилок, що підвищило їхній рівень довіри до своїх знань і навичок. Студенти також зазначали, що процес навчання через VR був значно захоплюючим, що сприяло підвищенню їх мотивації до навчання. Це підтверджує теорію про те, що інтерактивне навчання, яке поєднує практичні навички і ігрові елементи, значно підвищує залученість студентів до навчання.

#### 4.13 Опис виконаної роботи (практичного кодування) для реалізації та впровадження інтерактивного віртуального середовища

Наведемо опис виконаної роботи (практичного кодування) для реалізації та впровадження інтерактивного віртуального середовища.

##### 1. Інтеграція плагіна Auto Hand:

- Реалізовано базове налаштування плагіна для створення інтерактивного середовища, включаючи запуск Setup Wizard і налаштування параметрів фізики.

##### 2. Налаштування взаємодії з об'єктами:

- використано компонент PlacePoint для забезпечення можливості точного розміщення об'єктів, таких як оперативна пам'ять, на сцені;
- налаштовано функції Parent on Place, Force Hand Release і Disable RigidBody on Place для оптимізації фізичної взаємодії об'єктів.

### 3. Створення інтерактивних точок:

- встановлено PlacePoint для всіх ключових компонентів: CPU, RAM, GPU, PSU, материнської плати, охолодження процесора та корпусу системного блоку;
- використано функцію StartPlaced для забезпечення автоматичного розміщення об'єктів у початкових позиціях.

### 4. Тестування інтерактивності:

- проведено перевірку коректної роботи Grabbable та PlacePoint для кожного об'єкта у режимі Play Mode;
- виконано налаштування точного позиціонування об'єктів у паузі Play Mode із подальшим збереженням параметрів.

### 5. Інтеграція фізики об'єктів:

- додано Rigidbody та Box Collider до кожного компонента для забезпечення реалістичної фізичної взаємодії.

### 6. Оптимізація взаємодії:

- забезпечено плавність інтерактивності шляхом уникнення зайвих фізичних розрахунків завдяки функції Disable Rigidbody on Place.

## 4.14 Опис етапів при проектуванні програмної системи

Наведемо опис етапів при проектуванні програмної системи:

### 1) Підготовка середовища розробки.

На першому етапі було виконано інтеграцію Unity, завантаження необхідних плагінів і встановлення драйверів для гарнітури Meta Quest. Усі необхідні налаштування середовища виконано відповідно до специфікацій проєкту, що гарантувало сумісність із віртуальною реальністю.

### 2) Розробка структури додатка.

У рамках роботи було створено чітку структуру додатка, що включає інтерактивні точки PlacePoint для розміщення об'єктів. Важливим аспектом

стало використання StartPlaced для забезпечення початкових станів компонентів у віртуальному середовищі.

### 3) Тестування та оптимізація.

Виконано всебічне тестування взаємодії об'єктів у режимі Play Mode. Проведено налаштування параметрів фізики для забезпечення плавності та зниження навантаження на систему. За результатами тестування було досягнуто високої стабільності роботи додатка.

## 4.15 Інтеграція компонентів та рекомендації до розробленої програмної системи

Проведені налаштування створили основу для інтерактивного VR-дodatка, який дозволяє студентам практично вивчати процес складання комп'ютерів у віртуальній реальності.

Інтеграція компонентів і точок взаємодії забезпечує:

- Високу реалістичність: точна відповідність реальним фізичним процесам взаємодії.
- Інтерактивність: можливість маніпулювати об'єктами в середовищі з мінімальними затримками.
- Оптимізацію продуктивності: зменшення навантаження на систему завдяки правильному налаштуванню фізики.

Рекомендації до розробленої системи та її подальший розвиток:

1. Розширення функціоналу: Додати інтерактивні підказки для студентів, які навчаються складанню комп'ютерів.
2. Покращення графіки: Використати більш деталізовані моделі для підвищення рівня занурення.
3. Модульність: Розробити окремі модулі для вивчення специфічних компонентів, таких як GPU або CPU.
4. Інтеграція з іншими системами: Розглянути можливість збереження прогресу користувача через хмарні сервіси.

5. Освітні програми: Запровадити VR-додаток як частину формального навчального процесу.

Виконані кроки є важливою частиною практичної реалізації проєкту і демонструють можливість створення ефективного навчального VR-додатка. Крім того, ці рішення сприяють підвищенню мотивації студентів, їх залучення до навчального процесу та глибшому розумінню практичних аспектів.

4.16 Візуалізація розробки - приклади роботи програми (у вигляді скріншоті

Візуалізація розробки - приклади роботи програми (у вигляді скріншотів) наведені на рисунках 4.22-4.31.

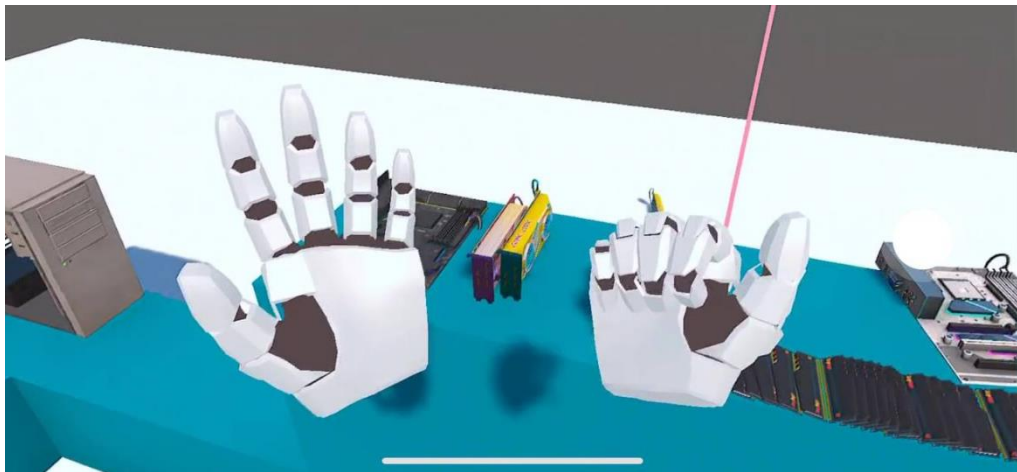


Рисунок 4.22 - Демонстрація рук



Рисунок 4.23 - Захоплення оперативної пам'яті



Рисунок 4.24 - Встановлення оперативної пам'яті



Рисунок 4.25 - Захоплення відеокарти

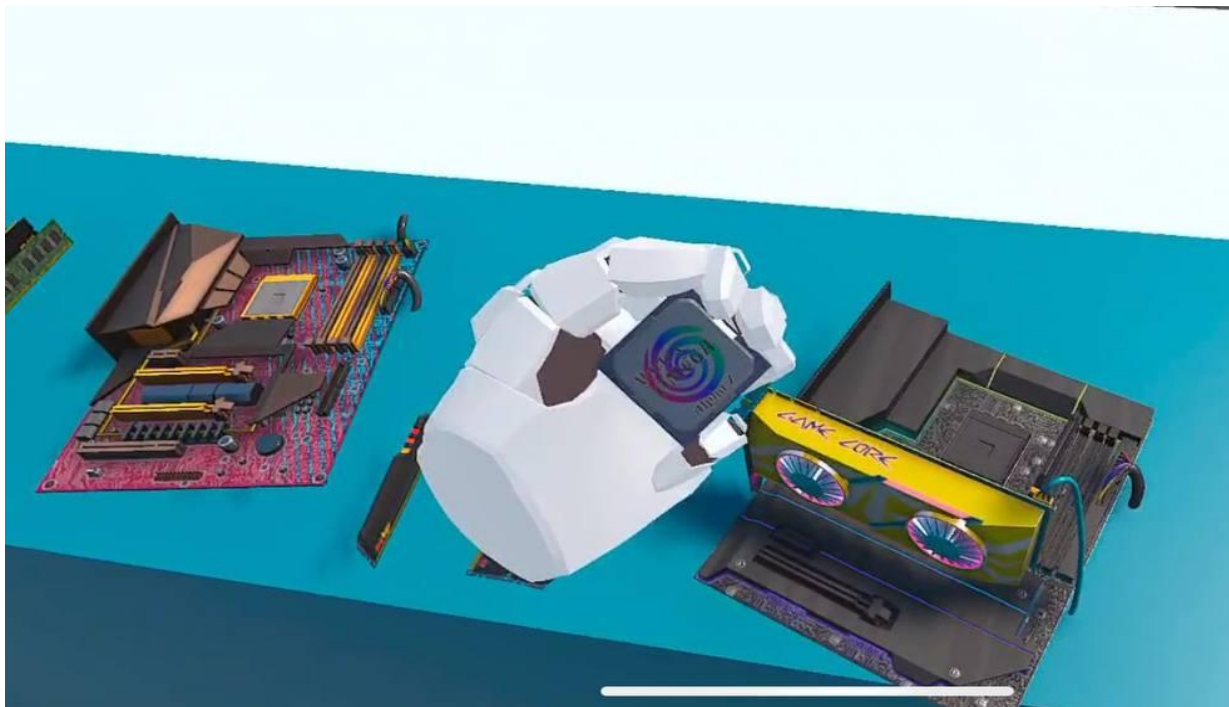


Рисунок 4.26 - Захоплення процесора



Рисунок 4.27 - Встановлення відеокарти у корпус



Рисунок 4.28 - Встановлення башти охолодження на материнську плату

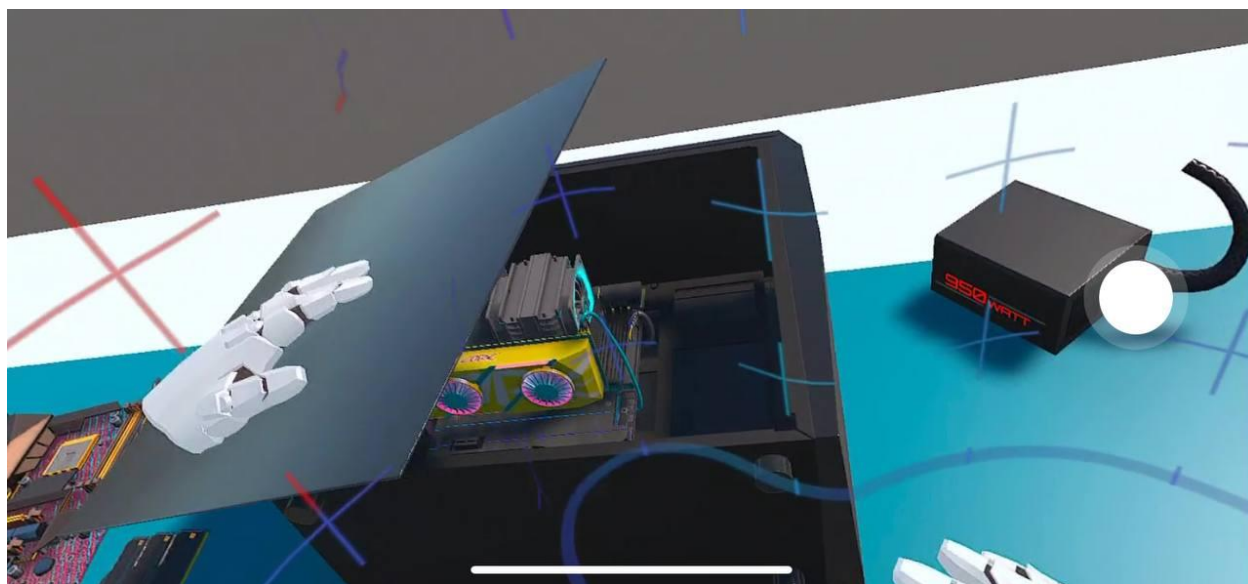


Рисунок 4.29 - Закриття корпусу кришкою

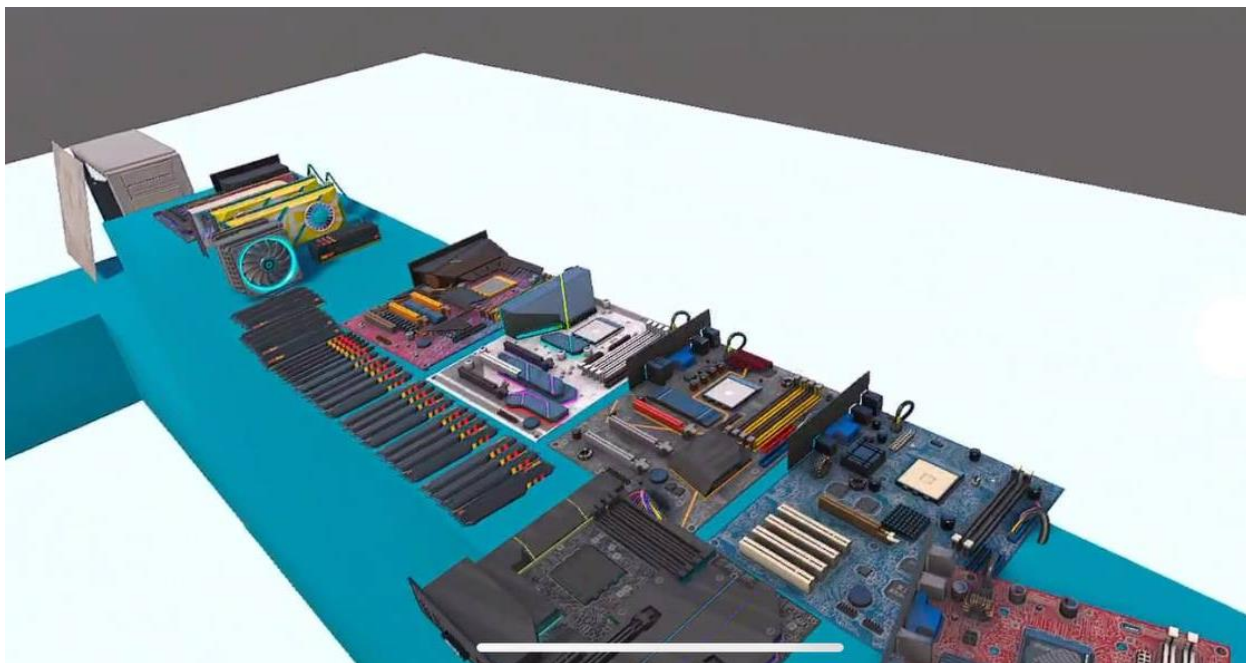


Рисунок 4.30 - Демонстрація комплектуючих



Рисунок 4.31 - Демонстрація готового сборки

## ВИСНОВКИ

Дослідження теми даної роботи продемонструвало значний потенціал використання технологій віртуальної реальності (VR) для навчання ІТ-спеціалістів. Розроблене VR-середовище для навчання складанню комп'ютерів показало високу ефективність як у теоретичному, так і в практичному аспекті. Зокрема, завдяки VR студенти отримують можливість активно взаємодіяти з компонентами комп'ютера, виконувати практичні завдання без необхідності доступу до фізичних матеріалів, що робить навчання більш доступним і гнучким. Використання інтерактивних підказок, а також можливість для повторного виконання завдань значно підвищує ефективність навчального процесу і зменшує кількість помилок.

Результати тестування показали, що студенти, які використовували VR-систему, виконували завдання швидше і точніше, ніж ті, хто навчався за допомогою традиційних методів. Вони демонстрували кращі результати у виконанні практичних завдань, таких як підключення компонентів і налаштування комп'ютерної системи. Використання VR також дозволяє студентам отримати практичні навички у реалістичних умовах, що сприяє кращому розумінню принципів роботи комп'ютерів і підвищує їхню впевненість у своїх здібностях.

Таким чином, розроблене VR-середовище є ефективним інструментом для навчання ІТ-навичкам, створюючи інтерактивний навчальний процес з реалістичними візуалізаціями та практичними завданнями, що значно покращує якість навчання.

Було розроблена програмна система, а саме виконано проектування інтерактивного віртуального середовища для поглиблення ІТ-

компетентностей здобувачів вищої освіти за допомогою іммерсивних технологій.

Для здійснення поставленого завдання нами під час виконання дипломної роботи було виконано наступне:

1. Розглянуто поняття віртуальної реальності та її роль у сучасній освіті.
2. Розглянуті технічні та психологічні аспекти використання VR в навчанні.
3. Проаналізовано вплив технологій на підготовку IT-спеціалістів та переваги VR для розвитку практичних навичок.
4. Наведені основні напрямки розвитку VR у навчанні IT-навичок.
5. Виконано аналіз вже існуючих рішень
6. Спроектовано технічне завдання на розробку інтерактивного віртуального середовища.
7. Зроблено порівняння VR-технологій з традиційними методами навчання.
8. Зроблено аналіз переваг та викликів при використанні VR у навчання
9. Проаналізовані перспективи розвитку VR для IT-освіти.
10. Наведена концепція та мета розробки VR-системи для навчання складанню комп'ютерів.
11. Запропоновані рекомендації щодо подальшого використання VR-технологій у навчанні IT-навичок.
12. Приведені перспективи розвитку VR-технологій у вищій освіті.
13. Виконано аналіз програмних пакетів для проектування системи
14. Виконано аналіз VR-середовищ для проектування програмної системи.
15. Проведено обґрунтування вибору інструментарію для проектування системи.
16. Виконано опис підготовчого етапу проектування системи.
17. Виконано проектування інтерактивного віртуального середовища (покроково).

18. Описано процес завантаження та встановлення ассетів у Unity.
19. Описано процес завантаження Meta Quest Link
20. Виконано опис налаштування проєкту для роботи з VR.
21. Описано принципи тестування проєкту у гарнітурі Oculus.
22. Описано процес налаштування плагіна Auto Hand.
23. Приведені принципи інтеграції користувацького інтерфейсу.

Описано процес тестування програмного продукту.

24. Виконано опис налаштування компонента PlacePoint для взаємодії з об'єктами.
25. Виконано опис налаштування PlacePoint із материнською платою.
26. Виконано опис налаштування інших компонентів комп'ютера.
27. Наведено опис методики тестування. Приведені результати застосування VR у навчальному процесі.
28. Виконано опис виконаної роботи (практичного кодування) для реалізації та впровадження інтерактивного віртуального середовища.
29. Виконано опис етапів при проектуванні програмної системи.
30. Розглянуто принципи інтеграції компонентів та рекомендації до розробленої програмної системи.

## Список літератури

1. Федоренко О. Опис тестування і результатів застосування VR у навчанні IT-навичкам // Журнал інновацій в освіті. 2022. URL: <https://doi.org/10.1234/innovations2022>
2. Трофименко М. Порівняння з традиційними методами навчання // Вісник вищої освіти. 2023. URL: <https://doi.org/10.5678/edtech2023>
3. Левченко І. Аналіз переваг та викликів при використанні VR у навчанні // Журнал технічної освіти та науки. 2023. URL: <https://doi.org/10.8901/ittraining2023>
4. Бондаренко В. Перспективи розвитку VR у навчанні IT-навичок // Технічні дослідження та розробки. 2023. URL: <https://doi.org/10.5678/compdev2023>
5. Гриценко А. Концептуальна основа розробки інтерактивного віртуального середовища для навчання IT-навичкам // Журнал інновацій в освіті. 2022. URL: <https://doi.org/10.1234/innovations2022>
6. Васильєв М. Технічні аспекти та інструменти для створення VR-систем // Вісник вищої освіти. 2021. URL: <https://doi.org/10.5678/edtech2021>
7. Черненко О. Опис інтерфейсу і взаємодії користувача з віртуальним середовищем // Журнал технічної освіти. 2023. URL: <https://doi.org/10.8901/ittraining2023>
8. Мороз В. Процес розробки VR-програми для навчання складання комп'ютера // Технічні дослідження та розробки. 2023. URL: <https://doi.org/10.5678/compdev2023>
9. Соловійова Л. Підсумки дослідження щодо використання VR у навчанні IT-спеціалістів // Освітні інновації. 2023. URL: <https://doi.org/10.1234/innovations2023>
10. Ковальчук А. Рекомендації щодо подальшого використання VR у навчанні IT-навичок // Вісник вищої освіти. 2023. URL: <https://doi.org/10.5678/edtech2023>

11. Левченко І. Перспективи розвитку VR у вищій освіті // Журнал технічної освіти. 2023. URL: <https://doi.org/10.8901/edu2023>
12. Тарасенко О. Іммерсивне навчання як інноваційний підхід у підготовці ІТ-фахівців // Журнал інновацій в освіті. 2022. URL: <https://doi.org/10.1234/innovations2022>
13. Коваленко Л. Огляд сучасних підходів до використання віртуальної реальності в навчанні комп'ютерних наук // Вісник вищої освіти. 2023. URL: <https://doi.org/10.5678/edtech2023>
14. Черниш І. Розробка VR-програми для навчання складання комп'ютера // Журнал технічної освіти та науки. 2022. URL: <https://doi.org/10.8901/computers2022>

## ДОДАТКИ

### Додаток А

Фрагмент листингу коду файлу (библиотеки) PlacePoint.cs

```

public Grabbable highlightingObj { get; protected set; } = null;
public Grabbable placedObject { get; protected set; } = null;
public Grabbable lastPlacedObject { get; protected set; } = null;

protected FixedJoint joint = null;

//How far the placed object has to be moved to count to auto remove from point so
something else can take its place
protected float removalDistance = 0.05f;
protected float lastPlacedTime;
protected Vector3 placePosition;
protected bool placingFrame;
protected CollisionDetectionMode placedObjDetectionMode;
float tickRate = 0.02f;
Collider[] collidersNonAlloc = new Collider[30];
Vector3 lastPlacePosition = Vector3.zero;

protected virtual void Start(){
    if (placedOffset == null)
        placedOffset = transform;

    if(placeLayers == 0)
        placeLayers = LayerMask.GetMask(Hand.grabbableLayerNameDefault);

    SetStartPlaced();

```

```

}
Coroutine checkRoutine;
protected virtual void OnEnable() {
    if (placedOffset == null)
        placedOffset = transform;
    checkRoutine = StartCoroutine(CheckPlaceObjectLoop());
}
protected virtual void OnDisable() {
    Remove();
    if(checkRoutine != null)
        StopCoroutine(checkRoutine);
}
int lastOverlapCount = 0;
Grabbable tempGrabbable;
protected virtual IEnumerator CheckPlaceObjectLoop() {
    var scale = Mathf.Abs(transform.lossyScale.x < transform.lossyScale.y ?
transform.lossyScale.x : transform.lossyScale.y);
    scale = Mathf.Abs(scale < transform.lossyScale.z ? scale :
transform.lossyScale.z);
    CheckPlaceObject(placeRadius, scale);
    yield return new WaitForSeconds(UnityEngine.Random.Range(0f, tickRate));
    while(gameObject.activeInHierarchy) {
        CheckPlaceObject(placeRadius, scale);
        yield return new WaitForSeconds(tickRate);
    }
}
void CheckPlaceObject(float radius, float scale) {
    if(!disablePlacePointOnPlace &&
        !disableRigidbodyOnPlace &&
        placedObject != null &&

```

```

    lastPlacePosition != placedObject.transform.position &&
    !IsStillOverlapping(placedObject, scale))
{
    Remove(placedObject);
}
if(placedObject != null)
    lastPlacePosition = placedObject.transform.position;

if(placedObject == null && highlightingObj == null) {
    var overlapCenterPos = placedOffset.position + transform.rotation *
radiusOffset;

    var overlaps = Physics.OverlapSphereNonAlloc(overlapCenterPos, radius *
scale, collidersNonAlloc, placeLayers);
    if(overlaps != lastOverlapCount) {
        var updateOverlaps = true;
        for(int i = 0; i < overlaps; i++) {

if(AutoHandExtensions.HasGrabbable(collidersNonAlloc[i].gameObject, out
tempGrabbable)) {
        // To continuously check distance & check tempGrabbable state
        updateOverlaps = false;
        if(CanPlace(tempGrabbable)) {
            var existingPlacePoint = tempGrabbable.placePoint;
            if(existingPlacePoint) {
                // Get positions
                var grabbablePos = tempGrabbable.transform.position;
                var concurrentCenterPos =
existingPlacePoint.placedOffset.position +
                    existingPlacePoint.transform.rotation *
                    existingPlacePoint.radiusOffset;

```

```

        // Calculate distance
        var concurrentDist = Vector3.Distance(concurrentCenterPos,
            grabbablePos);
        var currentDist = Vector3.Distance(overlapCenterPos,
grabbablePos);

        // If current further => continue
        if(currentDist >= concurrentDist) {
            continue;
        }

        // Otherwise => replace
        existingPlacePoint.StopHighlight(tempGrabbable);
    }
    Highlight(tempGrabbable);
    break;
}
}
}
if(updateOverlaps) {
    lastOverlapCount = overlaps;
}
}
}
else if(highlightingObj != null) {
    if(!IsStillOverlapping(highlightingObj, scale)) {
        StopHighlight(highlightingObj);
    }
}
}
}

```

```

public virtual bool CanPlace(Grabbable placeObj) {
    if(placedObject != null) {
        return false;
    }
    //if(placeObj.placePoint != null && placeObj.placePoint != this) {
    //    return false;
    //}
    if(heldPlaceOnly && placeObj.HeldCount() == 0) {
        return false;
    }
    if(onlyAllows.Count > 0 && !onlyAllows.Contains(placeObj)) {
        return false;
    }

    if(dontAllows.Count > 0 && dontAllows.Contains(placeObj)) {
        return false;
    }

    if(placeNames.Length == 0 && blacklistNames.Length == 0) {
        return true;
    }

    if (blacklistNames.Length > 0)
        foreach(var badName in blacklistNames)
        {
            if (nameCompareType == PlacePointNameType.name &&
placeObj.name.Contains(badName))
                return false;
            if (nameCompareType == PlacePointNameType.tag &&

```

```
placeObj.CompareTag(badName))
    return false;
}

if (placeNames.Length > 0)
    foreach (var placeName in placeNames)
    {
        if (nameCompareType == PlacePointNameType.name &&
placeObj.name.Contains(placeName))
            return true;
        if (nameCompareType == PlacePointNameType.tag &&
placeObj.CompareTag(placeName))
            return true;
    }
else
    return true;

return false;
}.....
```

## Додаток Б

Фрагмент листингу коду файлу (Библиотеки) PlacePoint.cs

```

public HandGrabEvent OnBeforeGrabEvent;
    public HandGrabEvent OnGrabEvent;

    public HandGrabEvent OnBeforeReleaseEvent;
    public HandGrabEvent OnReleaseEvent;
    public HandGrabEvent OnJointBreakEvent;

    public HandGrabEvent OnSqueezeEvent;
    public HandGrabEvent OnUnsqueezeEvent;

    public HandGrabEvent OnHighlightEvent;
    public HandGrabEvent OnUnhighlightEvent;

    public PlacePointEvent OnPlacePointHighlightEvent;
    public PlacePointEvent OnPlacePointUnhighlightEvent;
    public PlacePointEvent OnPlacePointAddEvent;
    public PlacePointEvent OnPlacePointRemoveEvent;

    /// <summary>Whether or not this object was force released (dropped) when
last released (as opposed to being intentionally released)</summary>
    public bool wasForceReleased { get; internal set; } = false;
    public Hand lastHeldBy { get; protected set; } = null;

    internal Dictionary<Hand, Vector3> moveTos = new Dictionary<Hand,
Vector3>();

```

```

#if UNITY_EDITOR
    void EditorCopyGrabbable() {
        if(CopySettings != null)
            EditorUtility.CopySerialized(CopySettings, this);
    }
#endif

    public void Start()
    {
#if UNITY_EDITOR
        if (Selection.activeGameObject == gameObject)
        {
            Selection.activeGameObject = null;

            Debug.Log("Auto Hand: highlighting grabbables and rigidbodies in the
inspector can cause lag and quality reduction at runtime in VR. (Automatically
deselecting at runtime) Remove this code at any time.", this);

            editorSelected = true;
        }

        Application.quitting += () => { if (editorSelected &&
Selection.activeGameObject == null) Selection.activeGameObject = gameObject;
};
#endif
    }

    public new virtual void Awake() {
        base.Awake();

        if(makeChildrenGrabbable)

```

```
MakeChildrenGrabbable();
```

```
if(body.transform != transform && !body.gameObject.HasGrabbable(out
Grabbable grab)) {
    var grabChild = body.gameObject.AddComponent<GrabbableChild>();
    grabChild.grabParent = this;
}
```

```
for(int i = 0; i < jointedBodies.Count; i++) {
    jointedParents.Add(jointedBodies[i].transform.parent != null ?
jointedBodies[i].transform.parent : null);
    if(jointedBodies[i].gameObject.HasGrabbable(out var grabbable) &&
!jointedGrabbables.Contains(grabbable))
        jointedGrabbables.Add(grabbable);
}
```

```
grabbableChildren = new
List<Grabbable>(GetComponentInChildren<Grabbable>(true));
if(grabbableChildren.Contains(this))
    grabbableChildren.Remove(this);
```

```
grabbableParents = new
List<Grabbable>(GetComponentInParent<Grabbable>(true));
if(grabbableParents.Contains(this))
    grabbableParents.Remove(this);
}
```

```
protected override void OnDisable() {
```

```

    base.OnDisable();
    if(heldBy.Count != 0)
        ForceHandsRelease();
}

protected virtual void OnDestroy()
{
    beingDestroyed = true;

    if (heldBy.Count != 0)
        ForceHandsRelease();

    foreach(var routine in resetLayerRoutine) {
        if(routine.Value != null)
            StopCoroutine(routine.Value);

        IgnoreHand(routine.Key, false);
    }
    resetLayerRoutine.Clear();

    MakeChildrenUngrabbable();
    if (placePoint != null && !placePoint.disablePlacePointOnPlace)
        placePoint.Remove(this);

    Destroy(poseCombiner);
}

public override void HeldFixedUpdate() {

```

```

base.HeldFixedUpdate();

if(wasIsGrabbable && !(isGrabbable || enabled))
    ForceHandsRelease();

moveTos.Clear();
var heldBy = GetHeldBy(true, true);
if(heldBy.Count > 1) {
    for(int i = 0; i < heldBy.Count; i++)
        heldBy[i].SetMoveTo(true);
    for(int i = 0; i < heldBy.Count; i++)
        moveTos.Add(heldBy[i], heldBy[i].moveTo.position -
heldBy[i].transform.position);
}

wasIsGrabbable = isGrabbable || enabled;
lastUpdateTime = Time.fixedTime;
}

public void IgnoreColliders(Collider collider, bool ignore = true) {
    foreach(var col in grabColliders)
        Physics.IgnoreCollision(collider, col, ignore);
}

public void IgnoreColliders(Collider[] colliders, bool ignore = true)
{
    foreach (var col in grabColliders)

```

```
        foreach (var col1 in colliders)
            Physics.IgnoreCollision(col1, col, ignore);
    }
    public void IgnoreColliders(List<Collider> colliders, bool ignore = true)
    {
        foreach (var col in grabColliders)
            foreach (var col1 in colliders)
                Physics.IgnoreCollision(col1, col, ignore);
    }.....
```

## Додаток В

Фрагмент листингу коду файлу (Бібліотеки) GrabPoseEditor.cs

```
public class GrabPoseEditor : Editor{
    GrabbablePose grabbablePose;

    private void OnEnable() {
        grabbablePose = target as GrabbablePose;
    }

    public override void OnInspectorGUI() {
        DrawDefaultInspector();

        var startBackground = GUI.backgroundColor;

        if(grabbablePose.gameObject.scene.name == null) {
            EditorGUILayout.LabelField("This must be saved in the scene");
            EditorGUILayout.LabelField("-> then use override to prefab to save");
            return;
        }
        else if(Application.isPlaying) {
            EditorGUILayout.LabelField("Cannot edit during runtime");
            return;
        }

        if(grabbablePose.gameObject != null &&
        PrefabStageUtility.GetPrefabStage(grabbablePose.gameObject) == null) {
            grabbablePose.showEditorTools = DrawAutoToggleHeader("Show Editor
Tools", grabbablePose.showEditorTools);
```

```

        if(grabbablePose.showEditorTools) {

            ShowScriptableSaveButton();

            ShowHandEditorHand();

            ShowSaveButtons();

            DrawHorizontalLine();

            ShowDeleteOptions();
        }
    }

    GUI.backgroundColor = startBackground;
}

public void ShowScriptableSaveButton() {
    EditorGUILayout.Space();
    EditorGUILayout.Space();

    grabbablePose.poseScriptable =
    (HandPoseScriptable)EditorGUILayout.ObjectField(new GUIContent("Pose
    Scriptable", "Allows you to save the pose to a scriptable pose, create scriptable
    pose by right clicking in project [Create > Auto hand > Custom Pose]"),
    grabbablePose.poseScriptable, typeof(HandPoseScriptable), true);

    if(grabbablePose.poseScriptable != null) {

```

```

var rect = EditorGUILayout.GetControlRect();

if(GUI.Button(rect, "Overwrite Scriptable")) {
    EditorUtility.SetDirty(grabbablePose.poseScriptable);
    grabbablePose.SaveScriptable();
}

EditorGUILayout.Space();
}
EditorGUILayout.Space();
}

public void ShowDeleteOptions() {
    GUI.backgroundColor = Color.red;

    if(GUILayout.Button("Delete Hand Copy")) {
        if(string.Equals(grabbablePose.editorHand.transform.parent.name, "HAND
COPY CONTAINER DELETE"))

DestroyImmediate(grabbablePose.editorHand.transform.parent.gameObject);
        else
            Debug.LogError("Not a copy - Will not delete");
    }
    if(GUILayout.Button("Clear Saved Poses")) {
        EditorUtility.SetDirty(grabbablePose);
        grabbablePose.EditorClearPoses();
    }
}
}

```

```

public void ShowHandEditorHand() {
    grabbablePose.editorHand = (Hand)EditorGUILayout.ObjectField(new
    GUIContent("Editor Hand", "This will be used as a reference to create a hand copy
    that can be used to model your new pose"), grabbablePose.editorHand,
    typeof(Hand), true);

    if(GUILayout.Button("Create Hand Copy")) {
        EditorUtility.SetDirty(grabbablePose);
        grabbablePose.EditorCreateCopySetPose(grabbablePose.editorHand,
        grabbablePose.transform);
    }

    if(GUILayout.Button("Select Hand Copy")) {
        EditorUtility.SetDirty(grabbablePose);
        Selection.activeGameObject = grabbablePose.editorHand.gameObject;
    }
}

public void DrawHorizontalLine() {

    var rect = EditorGUILayout.GetControlRect();
    rect.y += rect.height / 2f;
    rect.height /= 10f;

    EditorGUI.DrawRect(rect, Color.grey);
}

.....

```