

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

2026 p.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2026 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Косовському Сергію Богдановичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення VR-застосунку для моделювання навчального процесу із складання персональних комп'ютерів у середовищі Unity

затверджена наказом університету від 26 травня 2026 року № 435Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 11 червня 2026 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Огляд сучасних VR-технологій та існуючих рішень для моделювання складання персональних комп'ютерів.2. Моделювання структури VR-застосунку та механік взаємодії користувача з компонентами ПК.3. Реалізація та тестування VR-застосунку для навчального моделювання складання персонального комп'ютера.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) скріншоти інтерфейсу середовища Unity та VR-застосунку, демонстраційні зображення роботи навчального симулятора.

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

| Найменування розділу | Консультант<br>(посада, прізвище, ім'я, по батькові) | Позначка консультанта<br>про виконання розділу |      |
|----------------------|------------------------------------------------------|------------------------------------------------|------|
|                      |                                                      | підпис                                         | дата |
|                      |                                                      |                                                |      |
|                      |                                                      |                                                |      |

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи                          | Строк / терміни виконання етапів роботи | Примітка |
|-------|----------------------------------------------|-----------------------------------------|----------|
| 1     | Отримання завдання на кваліфікаційну роботу  | 13.04.2026                              |          |
| 2     | Аналіз завдання, підбір літератури           | 15.04.26-18.04.26                       |          |
| 3     | Аналіз літератури з проблеми, що вирішується | 19.04.26-21.04.26                       |          |
| 4     | Аналіз існуючих методів оптимізації          | 22.04.26-24.04.26                       |          |
| 5     | Формування кроків вирішення проблеми         | 25.04.26-02.05.26                       |          |
| 6     | Програмна реалізація                         | 27.04.26-19.05.26                       |          |
| 7     | Аналіз отриманих результатів                 | 20.05.26-01.06.26                       |          |
| 8     | Оформлення пояснювальної записки             | 02.06.26-10.06.26                       |          |
| 9     | Перевірка на нормоконтроль                   | 10.06.26-19.06.26                       |          |
| 10    | Перевірка на плагіат                         | 11.06.26-30.06.26                       |          |
| 11    | Рецензування                                 | 20.06.26-30.06.26                       |          |
| 12    | Підготовка презентації та доповіді           | 20.06.26-30.06.26                       |          |
| 13    | Занесення роботи в електронний архів         | 30.06.26-09.07.26                       |          |
| 14    | Попередній захист кваліфікаційної роботи     | 30.06.26-09.07.26                       |          |

Дата видачі завдання 13 квітня 2026 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Кобилін О. А.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 75 с., 2 табл., 42 рис., 32 джерела.

ВІРТУАЛЬНА РЕАЛЬНІСТЬ, МОДЕЛЮВАННЯ, ТРИВИМІРНА ГРАФІКА, UNITY, VR-ЗАСТОСУНОК, VR-СИМУЛЯТОР.

Об'єктом роботи є процес навчального моделювання складання персональних комп'ютерів у середовищі віртуальної реальності.

Метою роботи є розроблення VR-застосунку у середовищі Unity для моделювання складання персонального комп'ютера.

Під час роботи використано: методи тривимірного моделювання, засоби віртуальної реальності, XR Interaction Toolkit, алгоритми перевірки сумісності компонентів та методи оптимізації тривимірної графіки.

Практична цінність роботи полягає у створенні VR-застосунку для моделювання процесу складання персонального комп'ютера у безпечному інтерактивному середовищі.

Розроблено VR-застосунок із підтримкою інтерактивної взаємодії користувача з компонентами персонального комп'ютера.

Область застосування: освітні VR-системи, технічна освіта, комп'ютерні лабораторії, моделювання комп'ютерної архітектури.

MODELING, THREE-DIMENSIONAL GRAPHICS, UNITY, VIRTUAL REALITY, VR APPLICATION, VR SIMULATOR.

The object of this work is the process of educational simulation of personal computer assembly in a virtual reality environment.

The goal of this work is to develop a VR application in Unity for simulating personal computer assembly.

The following were used during the work: three-dimensional modeling methods, virtual reality technologies, XR Interaction Toolkit, component compatibility verification algorithms, and three-dimensional graphics optimization methods.

The practical value of the work lies in the creation of a VR application for educational simulation of personal computer assembly in a safe interactive environment.

A VR application with support for interactive user interaction with personal computer components has been developed.

Applications: educational VR systems, technical education, computer laboratories, computer architecture simulation.

## ЗМІСТ

|                                                                                                                             |    |
|-----------------------------------------------------------------------------------------------------------------------------|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів .....                                                    | 7  |
| Вступ.....                                                                                                                  | 9  |
| 1 Огляд проблеми та постановка задачі.....                                                                                  | 10 |
| 1.1 Технології віртуальної реальності та їх місце у сучасній освіті....                                                     | 10 |
| 1.1.1 Еволюція VR-технологій та впровадження їх в освітній процес.....                                                      | 10 |
| 1.1.2 Переваги та недоліки традиційного навчання порівняно з VR-модельованням.....                                          | 13 |
| 1.2 Огляд сучасних підходів до апаратно-програмного забезпечення VR-застосунків.....                                        | 15 |
| 1.3 Аналіз методів відтворення фізичних процесів та колізій у тривимірному просторі .....                                   | 17 |
| 1.4 Суперечності та обмеження сучасних методів візуалізації комп'ютерної архітектури .....                                  | 18 |
| 1.5 Аналіз існуючих рішень та симуляторів збирання персональних комп'ютерів. ....                                           | 19 |
| 1.6 Постановка задачі .....                                                                                                 | 21 |
| 2 Математичні моделі фільтрації зображень.....                                                                              | 23 |
| 2.1 Аналіз вимог до логіки та фізики навчального симулятора.....                                                            | 23 |
| 2.2 Модельовання VR-середовища навчального симулятора.....                                                                  | 25 |
| 2.3 Модельовання механік взаємодії користувача у VR-середовищі ..                                                           | 28 |
| 2.4 Модельовання системних зв'язків між фізичними властивостями об'єктів та логікою збирання персонального комп'ютера ..... | 32 |
| 2.5 Модельовання механіки з'єднання деталей.....                                                                            | 34 |
| 2.5.1 Алгоритми перевірки сумісності конекторів .....                                                                       | 34 |
| 2.5.2 Математичні моделі позиціонування об'єктів у тригерних зонах .....                                                    | 35 |

|       |                                                                                                    |    |
|-------|----------------------------------------------------------------------------------------------------|----|
| 2.6   | Теоретичні підходи до оптимізації тривимірних об'єктів у віртуальній реальності.....               | 37 |
| 3     | Практична реалізація та тестування VR-застосунку для моделювання складання ПК.....                 | 40 |
| 3.1   | Обґрунтування вибору інструментальних засобів .....                                                | 40 |
| 3.2   | Організація тривимірного простору та підготовка графічних тривимірних асетів.....                  | 44 |
| 3.2.1 | Імпорт тривимірних моделей комплектуючих комп'ютера та організація тривимірного простору .....     | 44 |
| 3.2.2 | Оптимізація тривимірного простору та моделей.....                                                  | 47 |
| 3.3   | Програмна реалізація функціональних складових навчального симулятора.....                          | 49 |
| 3.3.1 | Програмна реалізація контролера користувача у VR-середовищі.....                                   | 49 |
| 3.3.2 | Програмування механік взаємодії з комплектуючими .....                                             | 53 |
| 3.3.3 | Реалізація системи інтерактивних підказок, UI-панелей та аудіовізуального зворотного зв'язку ..... | 62 |
| 3.4   | Тестування розробленого застосунку та оцінювання продуктивності.....                               | 64 |
| 3.5   | Аналіз результатів практичного застосування та перспективи подальшого вдосконалення системи.....   | 67 |
|       | Висновки .....                                                                                     | 70 |
|       | Перелік джерел посилання .....                                                                     | 72 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПК – персональний комп'ютер

3D – Three-Dimensional (тривимірний простір або тривимірна графіка)

Atlas Texture – текстурний атлас для об'єднання кількох текстур в один файл

Baked Lighting – метод попереднього обчислення освітлення сцени

Collider – компонент Unity для визначення меж фізичної взаємодії об'єктів

CPU – Central Processing Unit (центральний процесор)

Draw Call – окремий виклик рендерингу об'єкта графічною системою

FPS – Frames Per Second (кількість кадрів за секунду)

GPU – Graphics Processing Unit (графічний процесор)

GPU Instancing – метод рендерингу великої кількості однакових об'єктів за один draw call

Layer Mask – механізм Unity для фільтрації взаємодії між об'єктами різних типів

Lerp – Linear Interpolation (лінійна інтерполяція значень або позицій)

LOD – Level of Detail (система рівнів деталізації тривимірних моделей)

Mesh – структура тривимірної моделі, що містить вершини, полігони та геометрію об'єкта

MeshCombiner – метод або скрипт для об'єднання кількох Mesh-об'єктів в один

Occlusion Culling – метод оптимізації рендерингу шляхом приховування невидимих об'єктів

Quaternion – математична модель представлення обертання об'єктів у тривимірному просторі

Rigidbody – компонент Unity для моделювання фізичних властивостей об'єкта

SDK – Software Development Kit (набір засобів для розроблення програмного забезпечення)

Slerp – Spherical Linear Interpolation (сферична лінійна інтерполяція кватерніонів)

Snap System – система автоматичного вирівнювання та позиціонування об'єктів

Socket Interactor – компонент Unity для автоматичної фіксації об'єктів у визначених точках

Static Batching – метод об'єднання статичних об'єктів сцени для оптимізації рендерингу

Trigger Zone – тригерна зона для визначення входу або виходу об'єкта з області взаємодії

UI – User Interface (інтерфейс користувача)

VR – Virtual Reality (віртуальна реальність)

XR – Extended Reality (розширена реальність)

XR Direct Interactor – компонент Unity для безпосередньої взаємодії з об'єктами у VR-середовищі

XR Grab Interactable – компонент Unity для захоплення та утримання об'єктів

XR Interaction Toolkit – пакет Unity для реалізації взаємодії у VR та XR середовищах

XR Origin – компонент Unity для керування положенням користувача у VR-середовищі

XR Ray Interactor – компонент Unity для дистанційної взаємодії з об'єктами за допомогою променя

## ВСТУП

Технології віртуальної реальності (VR) активно впроваджуються в освітній процес та професійну підготовку, дозволяючи створювати інтерактивні середовища для взаємодії з тривимірними об'єктами. Завдяки розвитку ігрових рушіїв та VR-пристроїв, ця технологія стала провідним напрямом розробки сучасних навчальних симуляторів [1].

Практика виконувалась у межах теми кваліфікаційної роботи «Розроблення VR-застосунку для моделювання навчального процесу із складання персональних комп'ютерів у середовищі Unity». У ході практики було здійснено аналіз предметної області, сформульовано функціональні вимоги до застосунку, обґрунтовано вибір інструментальних засобів, досліджено можливості рушію Unity, компонента XR Interaction Toolkit і сучасних засобів оптимізації тривимірної графіки.

Метою практики є закріплення теоретичних знань, набуття практичних навичок у сфері розробки прикладного ПЗ відповідно до кваліфікаційної роботи. У процесі виконання практики було розроблено модель застосунку, опрацьовано методи інтерактивної взаємодії користувача з віртуальними об'єктами, алгоритми валідації сумісності апаратних компонентів та способи оптимізації тривимірних моделей для забезпечення високої продуктивності у VR-середовищі.

Актуальність роботи полягає у необхідності створення інструменту, що поєднує теоретичну підготовку з практичним імерсивним моделюванням складання ПК. На відміну від двовимірних симуляторів, розроблений застосунок забезпечує ефект повної присутності та дозволяє відпрацювати просторові навички. Він стане ефективним рішенням для підготовки ІТ-фахівців і проведення безпечних лабораторних занять без ризику пошкодження дорогого реального обладнання.

## 1 ОГЛЯД ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Технології віртуальної реальності та їх місце у сучасній освіті

#### 1.1.1 Еволюція VR-технологій та впровадження їх в освітній процес

Технології віртуальної реальності (Virtual Reality, VR) є одним із найбільш перспективних напрямів розвитку сучасних інформаційних технологій. Вони забезпечують створення інтерактивного цифрового середовища, у якому користувач може взаємодіяти з віртуальними об'єктами в режимі реального часу. Завдяки використанню спеціалізованих пристроїв, таких як VR-шоломи, контролери руху та системи відстеження положення, користувач отримує ефект повного занурення у цифровий простір.

Перші концепції віртуальної реальності почали формуватися ще у другій половині XX століття. Одним із перших пристроїв, що реалізував принцип занурення у штучне середовище, стала система Sensorama, розроблена Мортоном Хейлігом у 1962 році. Вона поєднувала стереоскопічне зображення, звук, вібрацію та навіть імітацію запахів. Хоча технічні можливості того часу були обмеженими, саме ці розробки заклали основу для подальшого розвитку VR-технологій [2].

У 1980–1990-х роках розвиток комп'ютерної графіки та обчислювальної техніки сприяв появі перших комерційних VR-систем. Проте через високу вартість обладнання, низьку продуктивність та недостатню якість візуалізації технологія не отримала широкого поширення. Лише після 2010 року, із розвитком графічних процесорів, мобільних пристроїв та сенсорних систем, VR почала активно інтегруватися у різні сфери діяльності людини [3].

У 2026 році очікується поява нових поколінь VR-платформ, серед яких особливу увагу привертає система Steam Frame, що розглядається як один із найбільш технологічно розвинених етапів розвитку сучасної віртуальної реальності. Передбачається, що платформа забезпечуватиме покращену

якість тривимірної графіки, точніше фовеальна трансляція (foveated streaming) – це нова функція, що оптимізує деталізацію в області, на яку спрямований погляд користувача, що дозволяє більш ніж у 10 разів підвищити якість зображення та ефективність використання пропускну здатності (рис. 1.1). Особливе значення для освітніх VR-застосунків матиме інтеграція сучасних систем просторового позиціонування та оптимізованих засобів взаємодії з цифровими моделями. Такі можливості дозволять створювати більш реалістичні навчальні симулятори, зокрема для моделювання процесів складання комп'ютерних систем та взаємодії з апаратними компонентами у віртуальному середовищі.

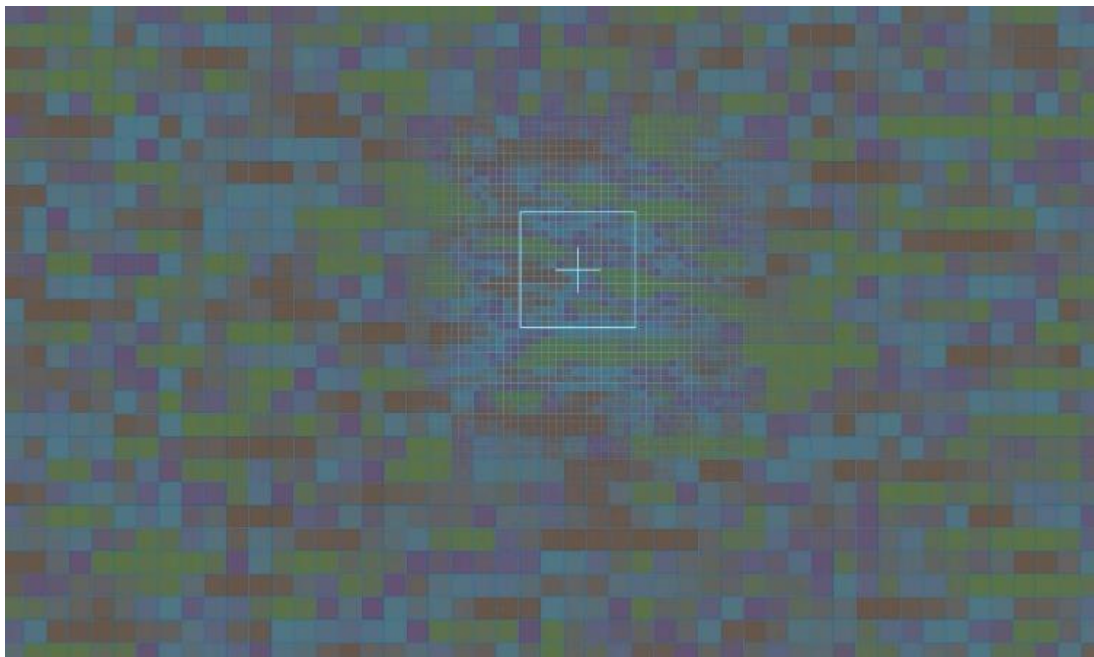


Рисунок 1.1 – Приклад роботи фовеальної трансляції

Сучасні VR-системи дозволяють створювати складні інтерактивні середовища з високим рівнем деталізації та реалістичною фізичною взаємодією об'єктів. Особливого поширення VR набула у сфері освіти, де технологія стала інструментом для моделювання навчальних ситуацій, візуалізації складних процесів та організації практичної підготовки студентів.

Використання VR у навчальному процесі пояснюється кількома важливими перевагами. По-перше, технологія дозволяє створювати безпечне

середовище для відпрацювання практичних навичок. Це особливо важливо у технічних дисциплінах, де робота з реальним обладнанням може бути дорогою або небезпечною. По-друге, VR забезпечує високий рівень залученості студентів завдяки інтерактивності та ефекту присутності. По-третє, віртуальні симуляції дозволяють багаторазово повторювати навчальні сценарії без додаткових витрат ресурсів.

У сучасній освіті VR активно використовується у медицині, авіації, архітектурі, інженерії та військовій підготовці. Наприклад, студенти-медики можуть тренувати проведення операцій у віртуальному середовищі, а майбутні інженери – взаємодіяти з тривимірними моделями складних механізмів.

Окрему увагу слід приділити використанню VR у технічній освіті. Віртуальні лабораторії дозволяють моделювати роботу апаратного забезпечення, демонструвати внутрішню структуру комп'ютерних систем та проводити практичні заняття без необхідності використання фізичного обладнання. Такий підхід особливо актуальний для вивчення комп'ютерної архітектури та процесу складання персональних комп'ютерів.

Попри значний потенціал VR, існують і певні проблеми впровадження технології в освітній процес. До них належать висока вартість обладнання, необхідність потужних комп'ютерних систем, складність розроблення якісного програмного забезпечення та недостатня адаптація навчальних програм до нових технологій. Крім того, деякі користувачі можуть відчувати дискомфорт під час тривалого перебування у VR-середовищі через так званий ефект «VR sickness».

Таким чином, технології віртуальної реальності поступово стають важливою складовою сучасної освіти. Їх використання дозволяє підвищити ефективність навчального процесу, покращити рівень засвоєння матеріалу та створити умови для практичного відпрацювання навичок у безпечному інтерактивному середовищі.

### 1.1.2 Переваги та недоліки традиційного навчання порівняно з VR-моделюванням

Традиційні методи навчання тривалий час залишаються основою освітнього процесу. Вони передбачають використання лекцій, друкованих матеріалів, лабораторних робіт та безпосередньої взаємодії між викладачем і студентом. Такий підхід має низку переваг, серед яких структурованість подачі матеріалу, доступність та можливість живого спілкування.

Проте традиційне навчання має і суттєві обмеження. Однією з основних проблем є недостатній рівень наочності під час пояснення складних технічних процесів. Наприклад, при вивченні комп'ютерної архітектури студенти часто працюють лише з теоретичними схемами або окремими фізичними компонентами, що ускладнює розуміння принципів взаємодії апаратного забезпечення.

Ще одним недоліком є обмежена можливість практичного відпрацювання навичок. Навчальні лабораторії потребують значних фінансових витрат на обладнання, а кількість доступних пристроїв часто не дозволяє забезпечити індивідуальну практику для кожного студента. Крім того, неправильне складання або підключення компонентів може призвести до пошкодження техніки.

VR-моделювання пропонує альтернативний підхід до організації навчального процесу. Завдяки використанню інтерактивного тривимірного середовища студенти можуть взаємодіяти з віртуальними об'єктами так, ніби вони є реальними. Це забезпечує кращу візуалізацію процесів та підвищує рівень залучення користувача.

Однією з головних переваг VR-навчання є можливість безпечного експериментування. Студент може багаторазово повторювати процедури складання комп'ютера, тестувати різні конфігурації обладнання та аналізувати результати без ризику пошкодження фізичних компонентів.

Також важливою перевагою VR є інтерактивність. На відміну від пасивного сприйняття інформації під час лекцій, користувач бере активну участь у навчальному процесі. Це позитивно впливає на концентрацію уваги, мотивацію та рівень засвоєння матеріалу.

Педагогічний ефект VR заснований на поєднанні візуалізації та активної взаємодії. Теорія когнітивного навантаження (Cognitive Load Theory) стверджує, що візуалізація складних понять у інтерактивному середовищі знижує рівень абстракції та спрощує розуміння. Конструктивістський підхід у навчанні підкреслює важливість діяльної компоненти – студенти краще засвоюють знання, коли самостійно взаємодіють із навчальним середовищем [4]. VR-застосунки на дають змогу поєднувати обидва підходи, роблячи процес навчання більш гнучким і продуктивним [5]. Далі наведений графік з дослідження про використання VR в освіті (рис. 1.2).

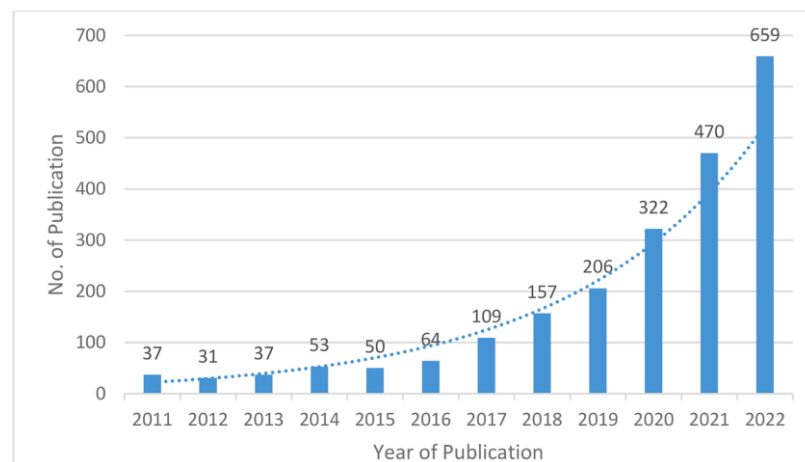


Рисунок 1.2 – Графік розподілу документів за роками використання віртуальної та доповненої реальності в освіті з 2011 по 2022 роки.

VR-системи дозволяють моделювати ситуації, які складно або неможливо реалізувати у звичайних навчальних умовах. Наприклад, студент може вивчати внутрішню будову процесора у збільшеному масштабі або спостерігати за передачею даних між компонентами комп'ютера у реальному часі.

Проте VR-навчання також має певні недоліки. Основним із них є висока вартість обладнання та розроблення програмного забезпечення. Для повноцінної роботи VR-застосунків необхідні потужні комп'ютери та спеціалізовані пристрої, що може бути проблемою для багатьох навчальних закладів.

Іншою проблемою є складність адаптації користувачів до нового формату навчання. Частина студентів та викладачів може мати недостатній рівень технічної підготовки для ефективного використання VR-технологій. Крім того, тривале перебування у VR-середовищі може викликати втому, запаморочення або дискомфорт.

Таким чином, VR-моделювання не замінює традиційне навчання повністю, а виступає як його доповнення. Поєднання класичних методів викладання з інтерактивними VR-технологіями дозволяє створити більш ефективне та сучасне освітнє середовище.

## 1.2 Огляд сучасних підходів до апаратно-програмного забезпечення VR-застосунків

Розроблення VR-застосунків потребує використання комплексного апаратно-програмного забезпечення, яке забезпечує реалістичне відображення тривимірного середовища, відстеження рухів користувача та обробку фізичної взаємодії об'єктів у режимі реального часу.

Основою будь-якої VR-системи є пристрій візуалізації – VR-шолом. Сучасні VR-шоломи оснащуються дисплеями високої роздільної здатності, системами стереоскопічного відображення та датчиками положення голови. Найбільш популярними пристроями є Meta Quest, HTC Vive та Valve Index. Вони забезпечують ефект присутності завдяки низькій затримці відображення та точному відстеженню рухів користувача.

Для взаємодії з віртуальним середовищем використовуються контролери руху. Вони дозволяють імітувати дії рук користувача та взаємодіяти з віртуальними об'єктами. Деякі сучасні системи підтримують технології hand tracking, які забезпечують відстеження рук без використання додаткових контролерів.

Важливим компонентом VR-систем є програмні рушії для створення тривимірних застосунків. Найбільш поширеними платформами є Unity та Unreal Engine. Unity характеризується простотою розроблення, великою кількістю навчальних матеріалів та підтримкою широкого спектра платформ [6]. Unreal Engine, у свою чергу, забезпечує високий рівень графічної деталізації та розвинуту систему фізичного моделювання.

Для створення VR-застосунків також використовуються спеціалізовані SDK та API, такі як OpenXR, SteamVR та Oculus SDK [7]. Вони забезпечують взаємодію програмного забезпечення з VR-пристроями та спрощують інтеграцію функцій відстеження руху, рендерингу та обробки подій.

Однією з ключових вимог до VR-застосунків є забезпечення високої продуктивності. Через необхідність одночасного рендерингу двох зображень для кожного ока система повинна підтримувати стабільну частоту кадрів, зазвичай не нижче 72–90 FPS. Низька продуктивність може призводити до затримок відображення та виникнення дискомфорту у користувача.

Для оптимізації продуктивності використовуються різні методи, серед яких рівні деталізації моделей (LOD), відсікання перекритих об'єктів (occlusion culling), попередній розрахунок освітлення (baked lighting) та оптимізація фізичних обчислень. Також значну роль відіграє правильна організація архітектури застосунку та ефективне використання ресурсів графічного процесора.

Таким чином, сучасні VR-застосунки є складними програмно-апаратними системами, які поєднують високопродуктивне обладнання, тривимірну графіку, фізичне моделювання та інтерактивні механізми взаємодії.

### 1.3 Аналіз методів відтворення фізичних процесів та колізій у тривимірному просторі

Одним із ключових елементів VR-застосунків є моделювання фізичної взаємодії об'єктів. Реалістична фізика дозволяє користувачеві сприймати віртуальне середовище як природне та логічне, що особливо важливо для навчальних симуляторів.

Фізичне моделювання у тривимірному просторі охоплює кілька основних напрямів: моделювання руху об'єктів, обробку колізій, симуляцію гравітації та взаємодію твердих тіл. Для реалізації таких механізмів у сучасних рушіях використовуються спеціалізовані фізичні рушії [8].

Одним із найбільш поширених підходів є використання rigid body physics – моделі твердих тіл. У цьому випадку кожен об'єкт має набір фізичних параметрів: масу, швидкість, силу тертя, інерцію та інші характеристики. Під час взаємодії система розраховує результат зіткнення відповідно до фізичних законів.

Важливою складовою фізичного моделювання є система колізій. Вона відповідає за визначення моменту зіткнення об'єктів та обчислення реакції на нього. Для оптимізації обчислень використовуються різні типи фізичних оболонок (collision meshes): паралелепіпедні (box collision), сферичні (sphere collision), капсульні (capsule collision) та на основі полігональної сітки (mesh collision).

У VR-застосунках особливого значення набуває точність фізичної взаємодії. Наприклад, під час симуляції складання персонального комп'ютера система повинна коректно визначати положення компонентів, можливість їх встановлення у відповідні слоти та взаємодію між елементами конструкції.

Для моделювання складних фізичних процесів сучасні рушії використовують технології PhysX, Havok та Chaos Physics [9]. Вони

дозволяють реалізовувати симуляцію руйнування об'єктів, тканин, рідин та складної взаємодії великої кількості фізичних елементів.

Проте надмірна деталізація фізичних розрахунків може негативно впливати на продуктивність системи. Через це розробники змушені знаходити баланс між реалістичністю симуляції та швидкодією застосунку.

Таким чином, фізичне моделювання є критично важливим компонентом VR-середовища, оскільки саме воно забезпечує природну взаємодію користувача з віртуальними об'єктами та підвищує рівень реалістичності симуляції.

#### 1.4 Суперечності та обмеження сучасних методів візуалізації комп'ютерної архітектури

Візуалізація комп'ютерної архітектури є важливим елементом технічної освіти, оскільки дозволяє студентам краще зрозуміти принципи роботи апаратного забезпечення. Проте сучасні методи візуалізації мають низку обмежень та суперечностей.

Однією з основних проблем є недостатня інтерактивність традиційних засобів навчання. Більшість навчальних матеріалів представлена у вигляді схем, зображень або відеоматеріалів, які не дозволяють студенту безпосередньо взаємодіяти з компонентами системи.

Іншою проблемою є складність просторового сприйняття. Під час вивчення комп'ютерної архітектури студентам необхідно розуміти розташування компонентів, способи їх підключення та принципи фізичної взаємодії між ними. Двовимірні схеми не завжди дозволяють сформувати повне уявлення про структуру системи.

Сучасні 3D-візуалізації частково вирішують цю проблему, проте часто залишаються недостатньо інтерактивними. У багатьох випадках користувач може лише переглядати модель без можливості повноцінної взаємодії з нею.

Також існує суперечність між рівнем деталізації та продуктивністю системи. Деталізовані моделі компонентів потребують значних обчислювальних ресурсів, особливо у VR-середовищі. Через це розробники змушені спрощувати геометрію моделей або обмежувати кількість одночасно активних об'єктів.

Ще одним обмеженням є відсутність універсальних платформ для навчального VR-моделювання комп'ютерної архітектури. Більшість існуючих рішень орієнтовані або на розважальні симулятори, або на вузькоспеціалізовані професійні системи [10].

Таким чином, сучасні методи візуалізації комп'ютерної архітектури потребують подальшого розвитку у напрямі підвищення інтерактивності, реалістичності та доступності для освітнього процесу.

### 1.5 Аналіз існуючих рішень та симуляторів збирання персональних комп'ютерів.

Сьогодні існує кілька програмних рішень, призначених для симуляції процесу складання персональних комп'ютерів. Найбільш відомим прикладом є PC Building Simulator [11]. Даний застосунок дозволяє користувачам виконувати віртуальне складання комп'ютерів, встановлювати компоненти та проводити тестування системи (рис. 1.3).

Основною перевагою PC Building Simulator є висока деталізація моделей комплектуючих та наближеність процесу складання до реальних умов. Користувач може встановлювати процесори, модулі оперативної пам'яті, системи охолодження та інші компоненти у відповідні слоти.

Проте система орієнтована переважно на розважальний сегмент та не забезпечує повноцінного освітнього функціоналу. Наприклад, застосунок недостатньо пояснює принципи взаємодії апаратних компонентів та не містить глибоких навчальних матеріалів.

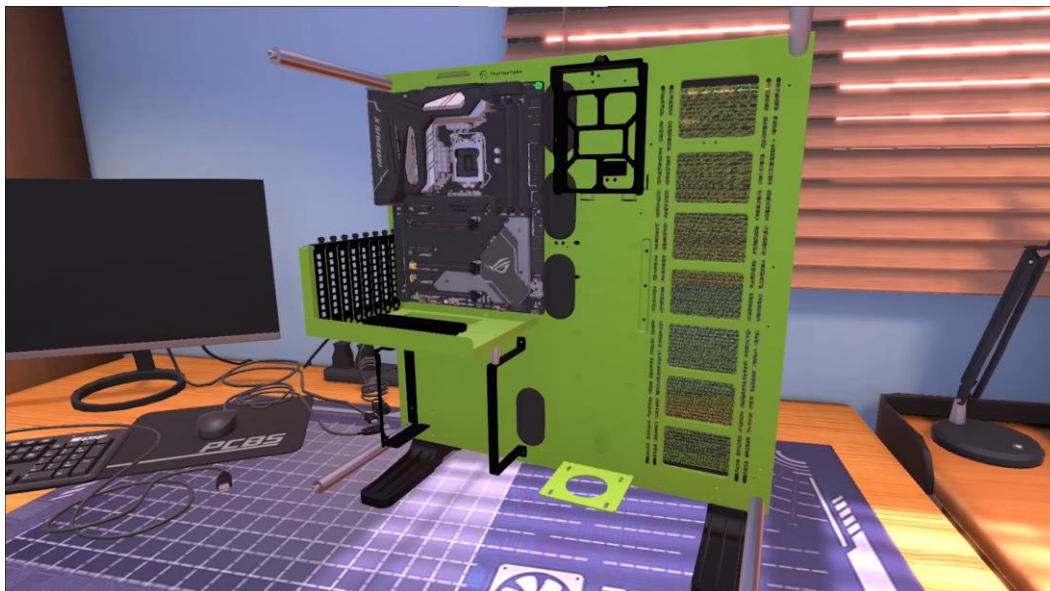


Рисунок 1.3 – Скриншот з застосунку PC Building Simulator

Проте даний застосунок немає версії для VR пристроїв, що значно збільшує можливості щодо оптимізації але обмежує рівень імерсивності та не дозволяє повноцінно відпрацювати просторові навички роботи з реальними деталями. Взаємодія за допомогою миші та клавіатури не передає фізичних особливостей з'єднання конекторів.

Існують також вебплатформи для конфігурації комп'ютерів, такі як PCPartPicker [12]. Вони дозволяють підбирати сумісні комплектуючі, проте не підтримують інтерактивне тривимірне моделювання процесу складання.

Деякі навчальні VR-проекти використовуються у технічних університетах для демонстрації будови комп'ютерних систем. Проте більшість таких рішень є закритими дослідницькими проектами та не мають широкого поширення.

Основними недоліками існуючих систем є:

- обмежена інтерактивність;
- недостатній рівень фізичного моделювання;
- відсутність адаптації до навчального процесу;
- висока вартість розроблення та підтримки;
- недостатня інтеграція VR-технологій;
- низька якість 3D моделей через проблеми оптимізації.

Таким чином, аналіз існуючих рішень демонструє актуальність створення VR-застосунку для навчального моделювання процесу складання персональних комп'ютерів із використанням сучасних методів фізичної взаємодії та тривимірної візуалізації.

## 1.6 Постановка задачі

Розвиток технологій віртуальної реальності відкриває нові можливості для вдосконалення технічної освіти та створення інтерактивних навчальних систем. Особливо актуальним є використання VR для моделювання процесів, які потребують практичної взаємодії з обладнанням. Одним із таких напрямів є вивчення комп'ютерної архітектури та процесу складання персональних комп'ютерів.

Об'єктом роботи є процес навчального моделювання складання персональних комп'ютерів у середовищі віртуальної реальності.

Метою роботи є розроблення VR-застосунку у середовищі Unity для моделювання процесу складання персонального комп'ютера з використанням сучасних технологій тривимірної візуалізації та фізичної взаємодії об'єктів.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати сучасні технології віртуальної реальності та особливості їх використання в освіті;
- дослідити сучасні підходи до розроблення VR-застосунків та апаратно-програмного забезпечення для їх реалізації;
- проаналізувати методи фізичного моделювання та системи колізій у тривимірному просторі;
- дослідити проблеми сучасних методів візуалізації комп'ютерної архітектури;
- виконати аналіз існуючих симуляторів складання персональних комп'ютерів та визначити їх переваги і недоліки;

- розробити структуру VR-застосунку для навчального моделювання складання ПК;
- реалізувати систему інтерактивної взаємодії користувача з компонентами комп'ютера у VR-середовищі;
- протестувати працездатність розробленого застосунку та оцінити ефективність його використання у навчальному процесі.

Очікуваним результатом роботи є створення функціонального VR-застосунку, який забезпечить інтерактивне навчання процесу складання персональних комп'ютерів та дозволить безпечно відпрацьовувати практичні навички у віртуальному середовищі.

## 2 МАТЕМАТИЧНІ МОДЕЛІ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

### 2.1 Аналіз вимог до логіки та фізики навчального симулятора

Під час розроблення VR-застосунку для моделювання процесу складання персонального комп'ютера важливим етапом є визначення вимог до логіки роботи системи та фізичної взаємодії між об'єктами. Саме правильна організація логіки взаємодії користувача з компонентами ПК забезпечує реалістичність навчального процесу та зручність використання VR-середовища.

На відміну від класичних відеоігор, навчальний VR-симулятор орієнтований не лише на взаємодію користувача із середовищем, а й на формування практичних навичок. У зв'язку з цим система повинна забезпечувати точне позиціонування компонентів, перевірку правильності складання, контроль сумісності деталей.

Однією з головних вимог до логіки симулятора є організація інтерактивної взаємодії між користувачем та об'єктами сцени. Користувач повинен мати можливість брати компоненти, переміщувати їх у просторі, повертати, встановлювати у відповідні слоти. Для цього необхідно реалізувати систему взаємодії, яка підтримує захоплення об'єктів VR-контролерами, визначення активного об'єкта, перевірку сумісності компонентів, систему фіксації деталей у визначених точках.

Важливим аспектом є також організація фізичної моделі взаємодії. У класичних фізичних симуляціях використовується гравітація та повноцінне моделювання руху тіл. Проте у навчальному VR-застосунку надмірно реалістична фізика може ускладнювати взаємодію користувача з об'єктами та негативно впливати на комфорт роботи, наприклад користувач може випадково викинути кудись деталь і вона загубиться десь на сцені.

У зв'язку з цим у застосунку передбачається вимкнення гравітації для більшості компонентів персонального комп'ютера. Такий підхід дозволяє

уникнути випадкового падіння деталей або їх втрати, спростити процес встановлення компонентів, зменшити кількість фізичних помилок, підвищити точність позиціонування об'єктів та зробити взаємодію з VR-середовищем більш комфортною для користувача. Крім того, відсутність гравітації позитивно впливає на стабільність роботи фізичного рушія Unity, оскільки зменшується кількість зайвих фізичних обчислень та небажаних колізій між компонентами.

Фактично компоненти у симуляторі функціонують у режимі керованої фізичної взаємодії. Об'єкт може переміщуватись користувачем вручну, проте його положення контролюється системою логічних обмежень та тригерних зон. Подібний підхід особливо важливий у VR-середовищі, де навіть незначні помилки позиціонування можуть негативно впливати на точність взаємодії та загальний комфорт користувача.

Особливу увагу необхідно приділити точності взаємодії між компонентами. Наприклад, процес встановлення оперативної пам'яті повинен враховувати правильний тип слота, допустиму орієнтацію модуля, коректне положення відносно материнської плати. Аналогічні вимоги застосовуються до встановлення процесора, відеокарти, накопичувачів та інших елементів системного блока. У реальних умовах неправильне встановлення компонентів може призвести до пошкодження обладнання, тому VR-симулятор повинен не лише демонструвати процес складання, а й формувати у користувача правильні практичні навички роботи з комп'ютерними компонентами.

Ще однією важливою вимогою є організація системи навчального супроводу користувача. Симулятор повинен забезпечувати підказки, візуальні індикатори сумісності [13]. Це дозволяє використовувати VR-застосунок не лише як симулятор, а і як повноцінний навчальний інструмент.

Таким чином, логіка та фізика навчального VR-симулятора повинні бути орієнтовані не на максимальну фізичну реалістичність, а на

забезпечення стабільної, зручної та зрозумілої взаємодії користувача з компонентами персонального комп'ютера.

## 2.2 Моделювання VR-середовища навчального симулятора

Створення VR-застосунку для моделювання процесу складання персонального комп'ютера потребує формування інтерактивного тривимірного середовища, яке забезпечує комфортну взаємодію користувача з об'єктами сцени та підтримує стабільну роботу застосунку у режимі реального часу. У межах даної роботи VR-середовище реалізується у вигляді спрощеної навчальної сцени, що дозволяє зосередити увагу користувача безпосередньо на процесі взаємодії з компонентами персонального комп'ютера.

Основою VR-простору є тривимірна сцена, створена у середовищі Unity. Сцена представляє собою прямокутне приміщення відкритого типу без даху, яке містить набір інтерактивних та допоміжних об'єктів. Такий підхід дозволяє зменшити навантаження на систему рендерингу та покращити продуктивність VR-застосунку [14].

Для побудови сцени використовуються базові тривимірні примітиви Unity, зокрема Plane, Cube та Quad. Стіни приміщення створені за допомогою масштабованих Mesh-об'єктів із текстурованими матеріалами. На стінах розміщуються логотипи кафедри та інформаційні елементи інтерфейсу, що формують навчальне середовище та покращують візуальне сприйняття сцени користувачем.

Важливим елементом VR-середовища є зона тестування взаємодії з об'єктами. У сцені розміщуються тестові тривимірні моделі, зокрема куби та інші геометричні примітиви, які використовуються для перевірки роботи системи захоплення, фізичної взаємодії та механік контролера користувача (рис. 2.1). Такий підхід дозволяє виконувати поетапне тестування окремих

механік ще до інтеграції повноцінних моделей компонентів персонального комп'ютера.



Рисунок 2.1 – Приклад тестових тривимірних моделей

Окрему роль у структурі VR-середовища виконує інформаційний екран із візуальною інструкцією. На екрані відображаються базові правила взаємодії з VR-системою, способи захоплення об'єктів, використання контролерів та елементи керування. Використання візуальних інструкцій дозволяє спростити адаптацію користувача до VR-середовища та зменшує потребу у зовнішньому навчальному супроводі.

Для створення компонентів персонального комп'ютера у сцені має бути реалізовано спеціальну панель вибору деталей. Даний елемент інтерфейсу містить набір кнопок, кожна з яких відповідає за створення певного компонента системного блока. Після натискання кнопки система виконує Spawn відповідного об'єкта у визначеній області сцени [15]. Такий підхід дозволяє реалізувати динамічне створення компонентів без необхідності постійного зберігання великої кількості моделей у сцені.

Механізм створення об'єктів може бути описаний у вигляді моделі:

$$O_i = \text{Spawn}(P_i, R_i, T_i), \quad (2.1)$$

де  $O_i$  – створений об'єкт;

$P_i$  – позиція створення;

$R_i$  – орієнтація об'єкта;

$T_i$  – тип компонента.

Для оптимізації VR-середовища у сцені потрібно реалізувати систему видалення непотрібних компонентів. Для цього використовується спеціальна зона у вигляді віртуального контейнера або «кошика», при взаємодії з яким об'єкти автоматично видаляються зі сцени. Це дозволяє зменшити кількість активних об'єктів та уникнути надмірного навантаження на фізичну систему Unity.

Видалення об'єкта виконується після перевірки входження у тригерну область:

$$D = \begin{cases} 1, & \text{якщо } O_i \in \text{TriggerZone} \\ 0, & \text{інакше} \end{cases}, \quad (2.2)$$

де  $D$  – стан видалення;

$O_i$  – об'єкт сцени;

*TriggerZone* – область деспавну.

Під час побудови VR-сцени також використовуються методи оптимізації тривимірної графіки. Для більшості статичних об'єктів застосовується baked lighting та static batching. Це дозволяє зменшити кількість draw calls та забезпечити стабільну частоту кадрів у VR-середовищі.

Таким чином, моделювання структури VR-середовища забезпечує формування функціонального навчального простору, який поєднує

інтерактивність, простоту взаємодії та оптимізовану організацію тривимірної сцени.

### 2.3 Моделювання механік взаємодії користувача у VR-середовищі

Одним із ключових елементів VR-застосунку є система взаємодії користувача із віртуальним середовищем. У навчальному симуляторі складання персонального комп'ютера механіки керування повинні забезпечувати природну взаємодію з компонентами системи, підтримувати комфортне переміщення користувача у сцені та створювати ефект присутності у віртуальному просторі.

Для реалізації системи керування краще всього використовувати стандартний VR-контролер Unity на основі XR Interaction Toolkit та OpenXR. Даний підхід дозволяє забезпечити сумісність застосунку з сучасними VR-пристроями та реалізувати базові механіки взаємодії без необхідності створення власної низькорівневої системи обробки вводу.

Користувач взаємодіє із середовищем за допомогою двох VR-контролерів – лівого та правого. Кожен контролер має набір функціональних кнопок та систем позиційного відстеження у просторі [16]. Координати контролерів постійно оновлюються у тривимірному просторі відповідно до положення рук користувача (рис. 2.2).

Позиція контролера може бути представлена у вигляді вектора:

$$P = (x, y, z), \quad (2.3)$$

де  $x, y, z$  – координати положення контролера у VR-просторі.

Для взаємодії з об'єктами використовується система захоплення XR Grab Interactable. Коли контролер наближається до об'єкта на допустимому відстань, система активує можливість взаємодії. Після натискання кнопки

захоплення компонент приєднується до контролера та починає рухатися разом із ним.

Відстань між контролером та об'єктом визначається за евклідовою метрикою:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \quad (2.4)$$

де  $d$  – відстань між об'єктами;

$(x_1, y_1, z_1)$  – координати контролера;

$(x_2, y_2, z_2)$  – координати об'єкта.

Якщо значення  $d$  є меншим за встановлений поріг взаємодії, система дозволяє захоплення об'єкта.

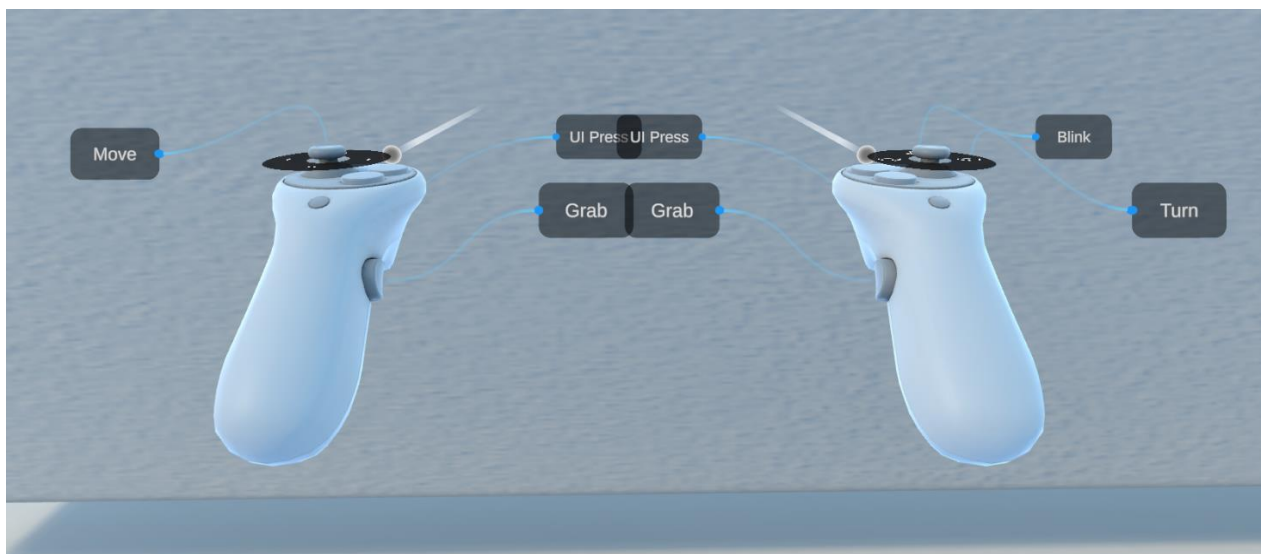


Рисунок 2.2 – Зображення контролерів у VR-застосунку

Для переміщення користувача у VR-середовищі використовується механіка Continuous Locomotion. Пересування реалізується за допомогою лівого аналогового стіка VR-контролера. Напрямок руху визначається відносно орієнтації VR-камери користувача, що забезпечує більш природне керування у просторі.

Швидкість переміщення може бути описана рівнянням:

$$V = S \cdot I, \quad (2.5)$$

де  $V$  – швидкість руху;

$S$  – базова швидкість;

$I$  – значення нахилу стіка.

Окрім стандартного переміщення у VR-середовищі, важливу роль відіграють альтернативні методи навігації, які забезпечують комфортне пересування користувача у віртуальному просторі. Одним із найбільш поширених рішень є телепортація, що дозволяє миттєво змінювати позицію користувача без плавного руху камери як показано на рисунку 2.3 (а) та (б).

Такий підхід використовується для зменшення навантаження на вестибулярний апарат людини та мінімізації ефекту «VR sickness», який може виникати через невідповідність між візуальним рухом у VR та фізичним положенням користувача у реальному просторі [17].



(а)

(б)

Рисунок 2.3 – Зображення механіки миттєвого переміщення

(а) натискання клавіші телепортації; (б) нова позиція користувача

Важливим елементом VR-взаємодії є підтримка дворукого захоплення об'єктів. Подібний механізм дозволяє одночасно використовувати два контролери для взаємодії з одним компонентом. У такому випадку система аналізує зміну положення та орієнтації обох контролерів і відповідно коригує

положення об'єкта у просторі. Це забезпечує більш природну взаємодію з великими або складними тривимірними моделями та підвищує точність маніпуляцій у VR-середовищі.

Для точного позиціонування компонентів також застосовуються механізми обертання об'єктів під час захоплення однією рукою. Подібний підхід дозволяє користувачеві змінювати орієнтацію тривимірного об'єкта без необхідності фізичного переміщення руки у просторі. Використання додаткового керування обертанням значно спрощує процес встановлення компонентів у визначені точки взаємодії та підвищує зручність роботи у VR-середовищі.

Для реалізації взаємодії між користувачем та об'єктами VR-середовища використовуються такі компоненти та інформаційні технології:

Для точного позиціонування компонентів також застосовуються механізми обертання об'єктів під час захоплення однією рукою. Подібний підхід дозволяє користувачеві змінювати орієнтацію тривимірного об'єкта без необхідності фізичного переміщення руки у просторі. Використання додаткового керування обертанням значно спрощує процес встановлення компонентів у визначені точки взаємодії та підвищує зручність роботи у VR-середовищі.

Для реалізації взаємодії між користувачем та об'єктами VR-середовища використовуються такі компоненти та інформаційні технології:

- базова система координат та позиціонування користувача у VR-просторі, яка забезпечує роботу камери та контролерів у межах сцени (XR Origin);

- система променевої взаємодії, що дозволяє взаємодіяти з віддаленими об'єктами за допомогою віртуального променя (XR Ray Interactor);

- механізм прямої фізичної взаємодії з об'єктами на близькій відстані (XR Direct Interactor);

- компонент Unity для реалізації захоплення, утримання та переміщення тривимірних об'єктів користувачем (XR Grab Interactable);
- система обробки введення, яка забезпечує підтримку кнопок, стіків та інших елементів VR-контролерів (Input Action System);
- фізичні компоненти Unity для моделювання руху та взаємодії об'єктів у просторі (Rigidbody);
- система перевірки зіткнень та взаємодії між тривимірними об'єктами сцени (Collision System Unity).

Важливою особливістю VR-контролера є підтримка фізичної взаємодії у реальному часі. Усі рухи контролерів безпосередньо впливають на положення об'єктів сцени, що створює ефект безпосереднього контакту з компонентами персонального комп'ютера.

Таким чином, моделювання механік взаємодії користувача у VR-середовищі забезпечує реалізацію інтуїтивного керування, підтримку фізичної взаємодії та формування комфортного навчального процесу у віртуальній реальності.

## 2.4 Моделювання системних зв'язків між фізичними властивостями об'єктів та логікою збирання персонального комп'ютера

У процесі створення VR-симулятора складання персонального комп'ютера важливим етапом є моделювання взаємозв'язків між фізичними характеристиками компонентів та логікою їх встановлення. На відміну від звичайного тривимірного середовища, навчальний симулятор повинен враховувати як геометричні властивості об'єктів, так і технічні правила сумісності компонентів.

Кожен елемент системного блока у VR-застосунку представляється окремим інтерактивним об'єктом із набором параметрів, серед яких тип компонента, габаритні розміри, точки з'єднання, допустимі способи

встановлення, параметри сумісності та логічні стани. Такий підхід дозволяє створити універсальну систему взаємодії, у якій кожен компонент має власний набір характеристик та правил поведінки.

Логіка складання ПК будується на основі системи залежностей між компонентами. Наприклад, встановлення процесора можливе лише за умови наявності відповідного сокета, правильної орієнтації процесора, відсутності іншого встановленого процесора та збігу типу сокета материнської плати. Подібний підхід застосовується і для інших компонентів системи.

Фізична модель взаємодії поєднується із логічною системою перевірок. Коли користувач переміщує компонент у тригерну область встановлення, система виконує перевірку типу компонента, аналіз допустимого слота, перевірку сумісності, аналіз орієнтації об'єкта та фіксацію компонента у правильній позиції. У разі успішного проходження перевірок система автоматично позиціонує об'єкт у відповідному слоті та оновлює логічний стан системи.

Для реалізації логіки взаємодії можуть використовуватись компоненти колізій для визначення фізичної взаємодії між об'єктами (Collider Components), тригерні зони для відстеження входу об'єкта у визначену область взаємодії (Trigger Zones), компонент Unity для автоматичної фіксації об'єктів у визначених точках встановлення (XR Socket Interactor), система фільтрації взаємодії між типами об'єктів (Layer Mask), об'єкти для централізованого збереження параметрів компонентів (Scriptable Objects) та система обробки подій рушія Unity (Unity Event System). Поєднання цих інструментів дозволяє створити гнучку архітектуру застосунку, яка може бути легко модифікована або розширена у майбутньому [18].

Особливу роль відіграє система станів компонентів. Наприклад, материнська плата може містити набір логічних прапорців: CPUInstalled, RAMInstalled, GPUInstalled та PowerConnected. Такі змінні дозволяють контролювати послідовність складання та перевіряти завершеність монтажу системи.

Таким чином, моделювання системних зв'язків між фізичними параметрами компонентів та логікою складання забезпечує правильне функціонування навчального симулятора та дозволяє реалізувати контрольований процес взаємодії користувача з VR-середовищем.

## 2.5 Моделювання механіки з'єднання деталей

### 2.5.1 Алгоритми перевірки сумісності конекторів

Одним із ключових механізмів VR-застосунку є система фіксації компонентів у визначених точках встановлення. Для реалізації такої взаємодії у Unity використовується механізм XR Socket Interactor.

Socket Interactor являє собою спеціальний компонент, який дозволяє автоматично приєднувати об'єкти до визначених позицій у сцені. У контексті симулятора складання ПК це дозволяє реалізувати встановлення процесора у сокет, монтаж оперативної пам'яті, підключення відеокарти, встановлення накопичувачів. Такий підхід значно спрощує взаємодію користувача із компонентами та дозволяє забезпечити точність встановлення деталей у VR-середовищі.

Основний принцип роботи Socket Interactor полягає у перевірці взаємодії між Socket-зоною, захопленим об'єктом та системою сумісності компонентів. Коли користувач наближає компонент до відповідної області встановлення, система виконує аналіз типу об'єкта, Layer Mask, допустимих параметрів взаємодії та відстані до точки фіксації. Після успішного проходження перевірки компонент автоматично фіксується у визначеній позиції.

У застосунку передбачається використання різних Layer для типів з'єднань, зокрема CPU Socket Layer, RAM Slot Layer та інш., кожен компонент містить набір допустимих Layer-взаємодій. Наприклад, процесор може взаємодіяти лише із CPU Socket Layer, а модуль оперативної пам'яті –

лише із RAM Slot Layer [19]. Такий підхід дозволяє значно спростити реалізацію системи перевірки сумісності та зменшити кількість логічних помилок під час роботи застосунку, процес роботи Socket Interactor на рисунку 2.4.

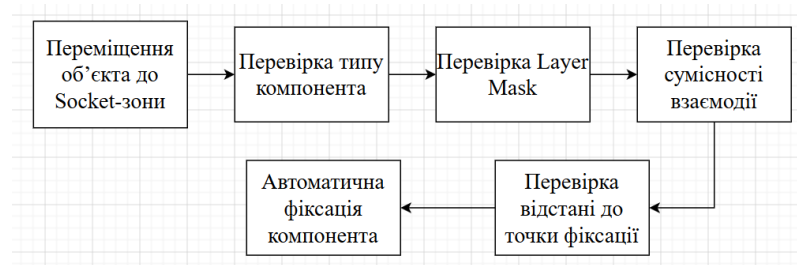


Рисунок 2.4 – Схема принципу дії компоненту Socket Interactor

Додатково Socket Interactor може використовуватися для реалізації візуальних підказок під час навчального процесу. Наприклад, у разі наближення сумісного компонента до сокета система може підсвічувати область встановлення або активувати анімацію позиціонування. Це дозволяє спростити процес навчання та зменшити кількість помилок під час взаємодії користувача з VR-середовищем.

Такий підхід дозволяє уникнути некоректного складання ПК, контролювати навчальний процес, реалізувати покрокову систему навчання та підвищити реалістичність взаємодії користувача із компонентами персонального комп'ютера.

### 2.5.2 Математичні моделі позиціонування об'єктів у тригерних зонах

Для забезпечення точного встановлення компонентів у VR-застосунку використовуються математичні моделі позиціонування об'єктів у Snap-зонах.

Основна задача Snap-системи полягає у плавному автоматичному вирівнюванні компонента відносно точки встановлення після входу у тригерну область.

Такий механізм позиціонування дозволяє зменшити кількість помилок під час встановлення компонентів та зробити взаємодію у VR-середовищі більш природною для користувача [20].

Позиціонування об'єкта може бути описане за допомогою векторної інтерполяції:

$$\vec{P}_{new} = \vec{P}_{current} + \alpha(\vec{P}_{target} - \vec{P}_{current}), \quad (2.6)$$

де  $\vec{P}_{current}$  – поточна позиція об'єкта;

$\vec{P}_{target}$  – цільова позиція сокета;

$\alpha$  – коефіцієнт інтерполяції.

Для вирівнювання обертання використовується сферична інтерполяція кватерніонів:

$$Q_{new} = Slerp(Q_{current}, Q_{target}, t), \quad (2.7)$$

де  $Q_{current}$  – поточне обертання;

$Q_{target}$  – необхідне обертання;

$t$  – коефіцієнт плавності.

Після входу компонента у Snap-зону система виконує аналіз допустимого типу об'єкта, перевірку відстані до точки встановлення, вирівнювання позиції, корекцію обертання та блокування подальшого фізичного руху.

Для визначення допустимої відстані використовується евклідова метрика 2.4.

Якщо значення  $d$  менше заданого порогу, система активує процес Snap-позиціонування.

Такий підхід забезпечує точність встановлення компонентів, зменшення кількості помилок користувача, плавність VR-взаємодії та покращення користувацького досвіду.

## 2.6 Теоретичні підходи до оптимізації тривимірних об'єктів у віртуальній реальності

Оптимізація тривимірних моделей є одним із ключових етапів розроблення VR-застосунків. Через високі вимоги до продуктивності VR-середовища навіть незначне перевантаження сцени може призводити до падіння частоти кадрів та виникнення дискомфорту у користувача.

Важливим елементом тривимірної графіки є Mesh – геометрична модель об'єкта, яка складається з вершин, полігонів та текстурних координат. Саме Mesh визначає форму тривимірного об'єкта та використовується рушієм Unity для його відображення у сцені. Кожен окремий Mesh потребує окремої обробки під час рендерингу, що безпосередньо впливає на навантаження графічної системи.

Однією з головних причин зниження продуктивності є велика кількість draw calls – окремих викликів рендерингу для кожного об'єкта сцени. Якщо сцена містить значну кількість окремих Mesh-об'єктів, Unity змушений багаторазово передавати дані до графічного процесора. Це створює додаткове навантаження на CPU та GPU, що особливо критично для VR-застосунків через необхідність рендерингу сцени одночасно для двох очей користувача.

Для вирішення цієї проблеми у VR-застосунку використовується підхід об'єднання геометрії за допомогою скрипту MeshCombiner.

MeshCombiner дозволяє об'єднувати кілька мешів в один, зменшувати кількість draw calls, оптимізувати рендеринг сцени, підвищувати стабільність FPS та зменшувати навантаження на центральний процесор [21]. Особливо

ефективним цей підхід є для статичних елементів сцени, декоративних об'єктів, меблів, елементів лабораторного середовища та компонентів корпусу ПК.

Принцип роботи MeshCombiner полягає у формуванні єдиного Mesh-файлу з кількох окремих моделей. Після об'єднання Unity виконує рендеринг групи об'єктів як одного елемента сцени, що суттєво зменшує навантаження на систему під час роботи VR-застосунку. Це особливо важливо у VR-середовищах, де необхідно підтримувати стабільну частоту кадрів для уникнення дискомфорту користувача.

Також важливими методами оптимізації VR-середовища є використання LOD-систем, baked lighting, occlusion culling, спрощення collision meshes, компресія текстур, atlas-текстур, статичного batching та GPU instancing.

LOD-системи (Level of Detail) – це метод оптимізації, при якому для одного об'єкта створюється декілька версій моделі з різною деталізацією. Чим далі об'єкт знаходиться від користувача, тим менш деталізована модель використовується для його рендерингу. Такий підхід дозволяє суттєво зменшити кількість полігонів у сцені та підвищити продуктивність VR-застосунку.

Baked lighting – це технологія попереднього обчислення освітлення сцени [22]. Під час використання цього методу інформація про освітлення зберігається у спеціальних lightmap-текстурах, що дозволяє уникнути складних обчислень освітлення у режимі реального часу. Це значно зменшує навантаження на графічний процесор та покращує стабільність FPS.

Occlusion culling – це метод приховування об'єктів, які не знаходяться у полі зору користувача або перекриті іншими об'єктами сцени. Unity не виконує рендеринг прихованих елементів, що дозволяє зменшити кількість draw calls та оптимізувати використання ресурсів системи.

Спрощення collision meshes полягає у використанні простіших форм колізій замість складної геометрії об'єкта. Для цього часто застосовуються

Box Collider, Sphere Collider або Capsule Collider. Використання спрощених колайдерів зменшує навантаження на фізичний рушій Unity та прискорює обчислення взаємодій між об'єктами.

Компресія текстур використовується для зменшення обсягу відеопам'яті, необхідної для зберігання графічних ресурсів. Стиснені текстури займають менше пам'яті та швидше завантажуються, що позитивно впливає на швидкодію VR-застосунку.

Atlas-текстури – це метод об'єднання кількох текстур в одну велику текстурну карту. Такий підхід дозволяє зменшити кількість звернень до GPU під час рендерингу та скоротити кількість draw calls у сцені.

Статичний batching – це механізм Unity, який автоматично об'єднує статичні об'єкти сцени у великі групи для спільного рендерингу. Це дозволяє зменшити кількість окремих викликів рендерингу та підвищити продуктивність системи.

GPU instancing – це технологія, яка дозволяє рендерити велику кількість однакових об'єктів за допомогою одного draw call. Даний метод особливо ефективний для сцен із повторюваними елементами, оскільки суттєво зменшує навантаження на процесор [23].

Комплексне використання таких методів дозволяє підтримувати високу продуктивність навіть у сценах із великою кількістю тривимірних об'єктів та інтерактивних елементів.

Окрему увагу необхідно приділити оптимізації освітлення. Використання real-time lighting у VR-сценах створює значне навантаження на GPU, тому доцільно застосовувати baked lighting для статичних об'єктів сцени.

Таким чином, використання MeshCombiner та інших методів оптимізації дозволяє забезпечити стабільну роботу VR-застосунку, підтримувати високу частоту кадрів та покращити загальний користувацький досвід у середовищі віртуальної реальності.

### **3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ VR-ЗАСТОСУНКУ ДЛЯ МОДЕЛЮВАННЯ СКЛАДАННЯ ПК**

#### **3.1 Обґрунтування вибору інструментальних засобів**

Під час розроблення VR-застосунку для моделювання процесу складання персонального комп'ютера було проаналізовано сучасні програмні платформи та інструментальні засоби створення віртуальної реальності. Основними критеріями вибору були підтримка VR-пристроїв, наявність готових засобів взаємодії користувача з об'єктами, можливість роботи з фізичними моделями та доступність інструментів оптимізації тривимірної графіки.

Створення VR-застосунку для навчального моделювання складання персональних комп'ютерів потребує використання програмних засобів, які забезпечують побудову тривимірного середовища, підтримку VR-пристроїв, фізичну взаємодію між об'єктами та високу продуктивність під час виконання застосунку. Крім цього, важливою вимогою є можливість швидкого прототипування та подальшого розширення функціональності системи.

Unity є одним із найпоширеніших інструментів для розроблення VR-застосунків завдяки підтримці сучасних стандартів віртуальної реальності, компонентній архітектурі та великій кількості готових засобів взаємодії. Використання Unity дозволяє реалізувати складні механіки роботи з тривимірними об'єктами без необхідності створення низькорівневих алгоритмів рендерингу та обробки пристроїв введення [24]. У межах даної роботи використовується Unity версії 6 (6000.0 LTS), яка забезпечує стабільну підтримку пакету XR Interaction Toolkit та стандарту OpenXR. Головне робоче середовище Unity, що використовується для створення сцени, налаштування компонентів, програмування логіки взаємодії та тестування застосунку, представлено на рисунку 3.1.

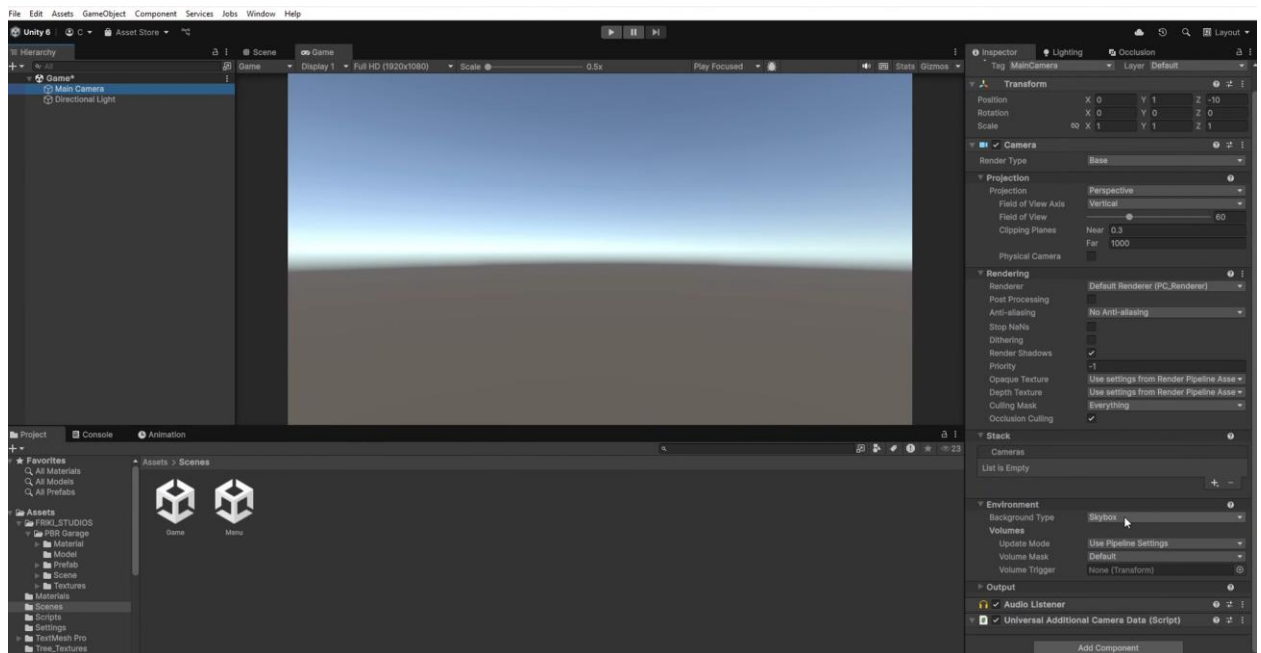


Рисунок 3.1 – Головне вікно середовища Unity

Для забезпечення сумісності із сучасними VR-пристроями у проєкті використовується стандарт OpenXR. Його застосування дозволяє працювати з різними моделями VR-гарнітур через єдиний програмний інтерфейс без необхідності окремого налаштування для кожного виробника. Налаштування OpenXR у середовищі Unity для активації підтримки VR-пристроїв та обробки даних від контролерів користувача зображено на рисунку 3.2.

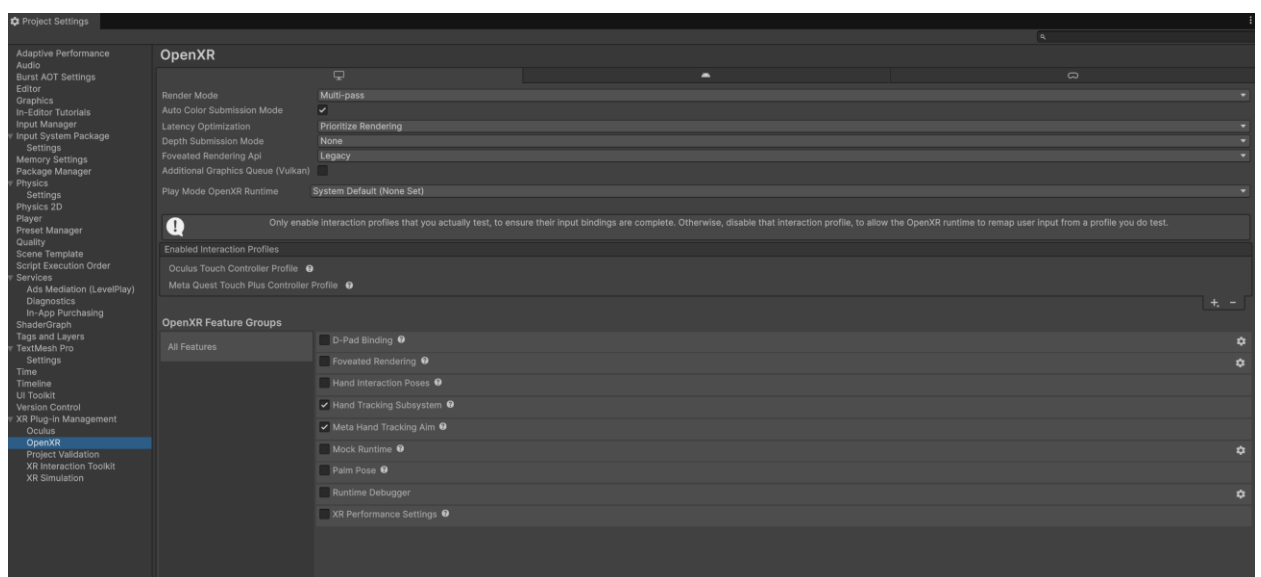


Рисунок 3.2 – Налаштування OpenXR у проєкті Unity

Важливим компонентом розроблення є XR Interaction Toolkit. Даний пакет містить набір готових механізмів для реалізації захоплення об'єктів, телепортації, роботи із сокетом, взаємодії через промені та інших функцій, необхідних для створення VR-середовища. У роботі використовується XR Interaction Toolkit версії 3.0, що є актуальною версією пакету для Unity 6 та повністю підтримує стандарт OpenXR. Підключений XR Interaction Toolkit у менеджері пакетів Unity наведено на рисунку 3.3.

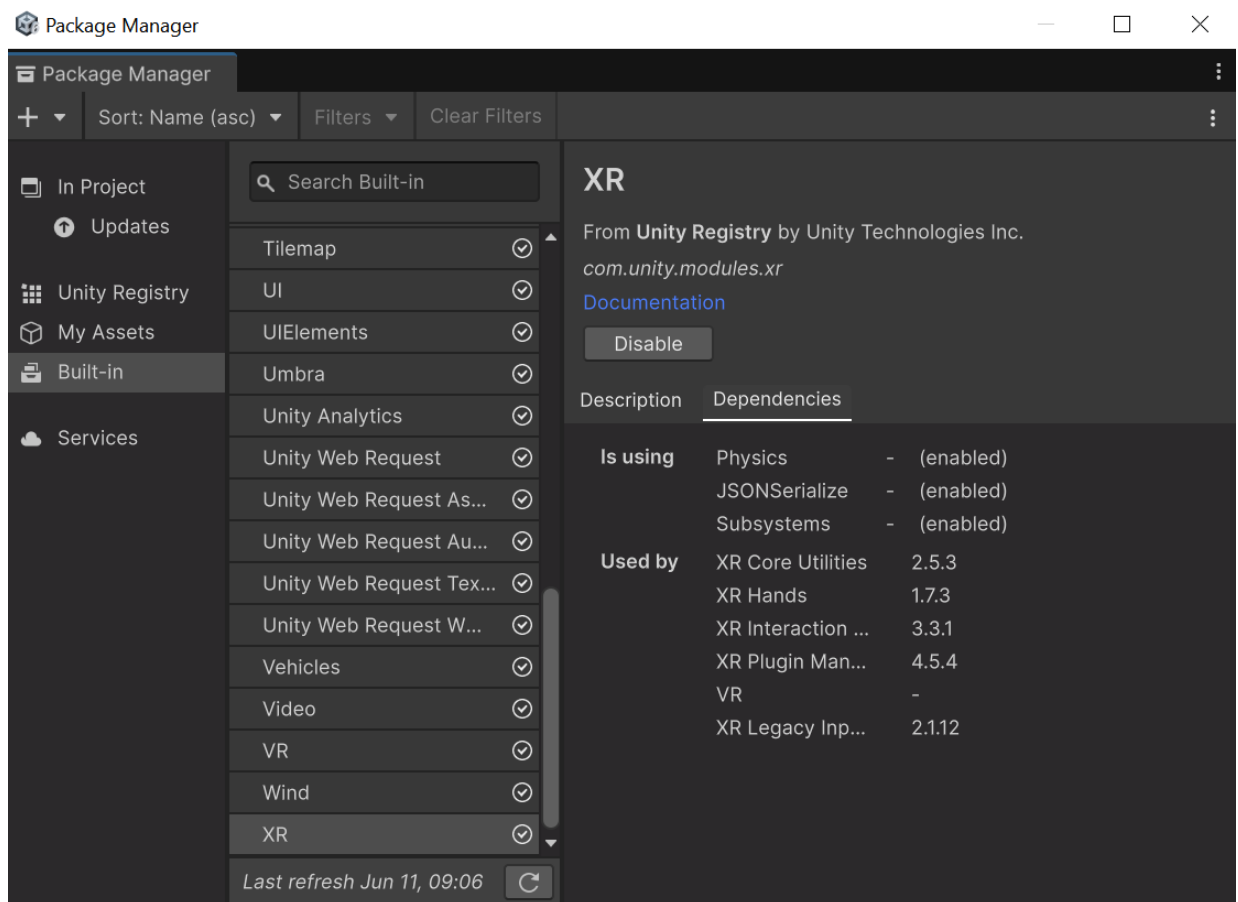


Рисунок 3.3 – Пакет XR Interaction Toolkit у менеджері пакетів Unity

Для створення та підготовки тривимірних моделей використовується програмний пакет Blender. За допомогою даного середовища виконується редагування готових моделей об'єктів, отриманих із зовнішніх джерел [25]. Blender також використовується для оптимізації геометрії моделей, налаштування матеріалів, текстур та експорту у формати, сумісні з Unity (рис. 3.4).

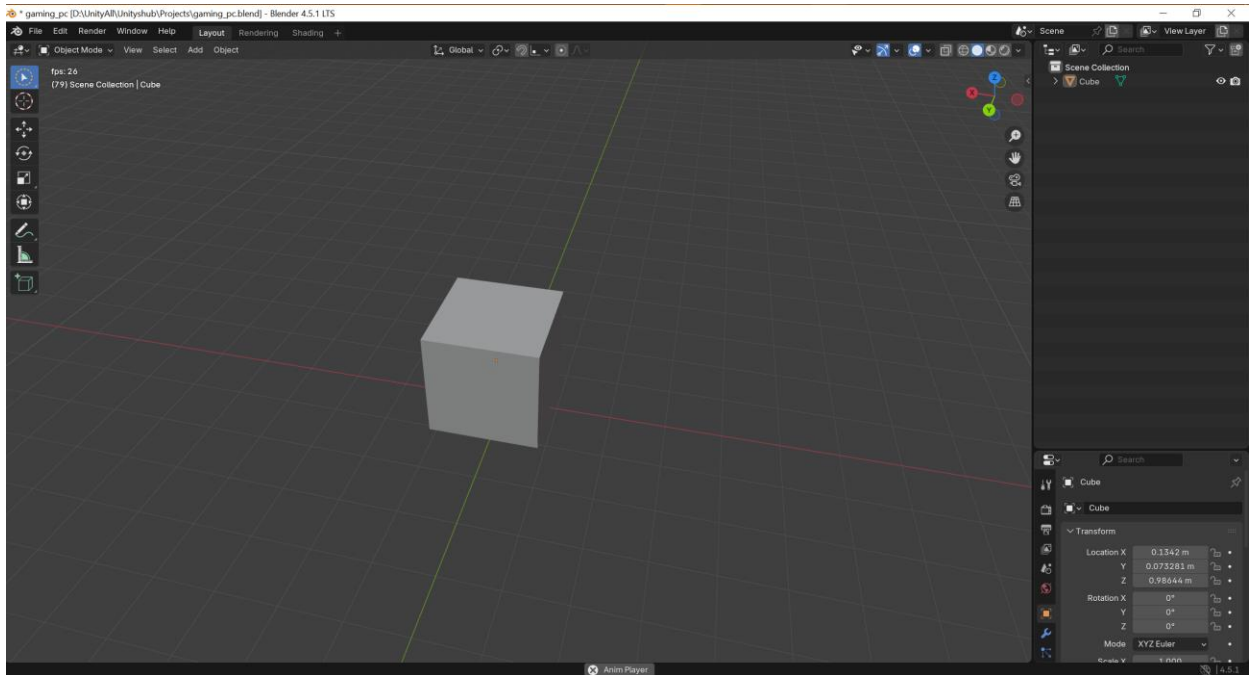


Рисунок 3.4 – Робоче середовище Blender

Основні характеристики використаних інструментів наведені у таблиці 3.1.

Таблиця 3.1 – Інструментальні засоби розроблення

| Засіб                  | Призначення                                                                                                                                                              |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unity                  | розроблення VR-застосунку; побудова тривимірної сцени, налаштування компонентів, реалізація логіки взаємодії та тестування застосунку                                    |
| Blender                | створення та редагування 3D-моделей комплектуючих ПК; оптимізація геометрії, налаштування матеріалів і текстур, запікання нормалей (Normal Baking), експорт у формат FBX |
| OpenXR                 | підтримка VR-пристроїв; забезпечення кросплатформної сумісності з гарнітурами різних виробників через єдиний програмний інтерфейс                                        |
| XR Interaction Toolkit | взаємодія користувача з об'єктами у VR-середовищі; реалізація механік захоплення, телепортації, роботи із сокетами та взаємодії через промені                            |

## 3.2 Організація тривимірного простору та підготовка графічних тривимірних асетів

### 3.2.1 Імпорт тривимірних моделей комплектуючих комп'ютера та організація тривимірного простору

Тривимірні моделі комплектуючих персонального комп'ютера імпортуються до Unity у форматі FBX. Перед імпортом кожен модель підготовлено у Blender: виконано очищення геометрії, застосування всіх трансформацій (Apply Transformations), центрування точки відліку (Origin) та налаштування орієнтації осей відповідно до координатної системи Unity.

Після імпорту до редактора Unity для кожної моделі виконується налаштування компонентів: додається Mesh Collider або спрощений примітивний колайдер (Box/Capsule Collider) для забезпечення фізичної взаємодії, компонент Rigidbody для моделювання фізичних властивостей об'єкта, а також компонент XR Grab Interactable, що дозволяє захоплювати об'єкт VR-контролером. Для комплектуючих, які мають фіксуватись у певних точках, додатково налаштовується шар взаємодії (Layer) відповідно до типу компонента.

Після створення проєкту у Unity було реалізовано базову структуру навчальної сцени. Одним із основних етапів розроблення є створення віртуального навчального середовища, у якому користувач виконуватиме складання персонального комп'ютера. Під час проєктування сцени враховувались вимоги до продуктивності VR-застосунків, зручності навігації та доступності всіх елементів взаємодії.

Основою сцени є навчальна лабораторія, представлена прямокутним приміщенням із відкритою верхньою частиною. Для спрощення структури сцени було обрано прямокутне приміщення відкритого типу без даху. Такий підхід дозволяє зменшити кількість полігонів, уникнути додаткового навантаження на систему освітлення та одночасно забезпечити достатній

рівень освітлення сцени. Загальний вигляд VR-сцени після базового налаштування представлено на рисунку 3.5.

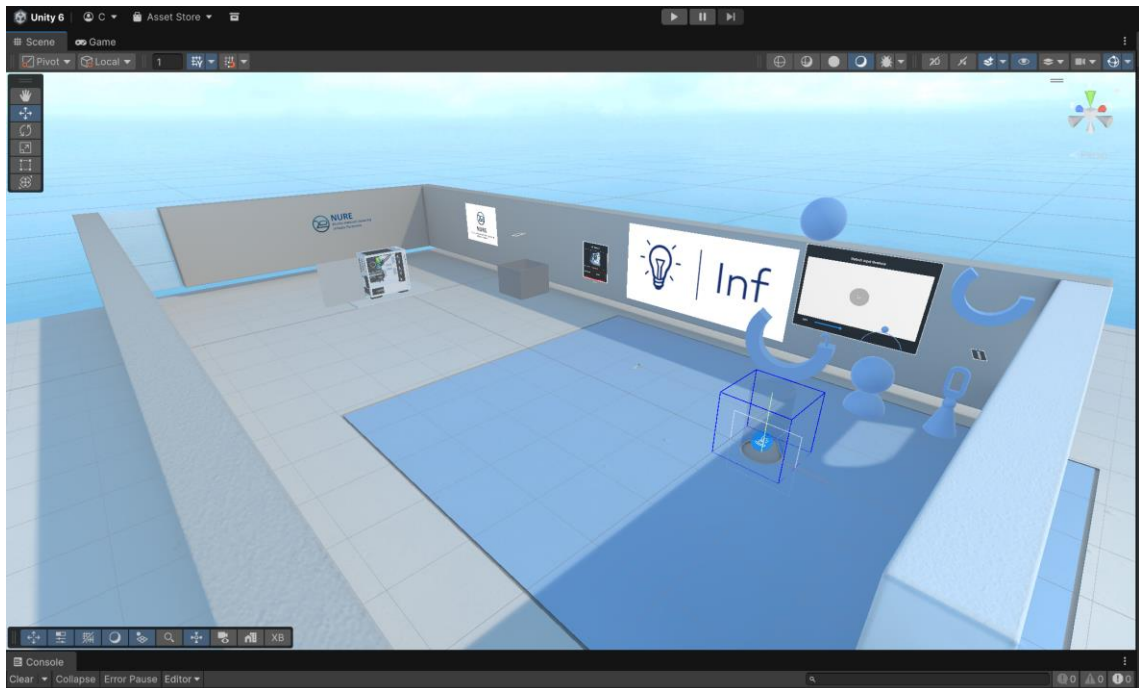


Рисунок 3.5 – Загальний вигляд VR-сцени

На стінах приміщення розміщено інформаційні матеріали та логотипи кафедри для формування навчальної атмосфери та візуальної ідентифікації проєкту. Такі об'єкти не беруть участі у взаємодії користувача, проте покращують візуальне сприйняття середовища (рис. 3.6).



Рисунок 3.6 – Розміщення інформаційних елементів сцени

Оскільки застосунок орієнтований на навчальний процес, важливим елементом сцени є інформаційна панель користувача. На ній відображаються інструкції щодо переміщення у VR-середовищі, способів захоплення об'єктів та використання контролерів. Інформаційний екран дозволяє швидко ознайомитися з основними принципами роботи системи без необхідності використання додаткових довідкових матеріалів (рис. 3.7).

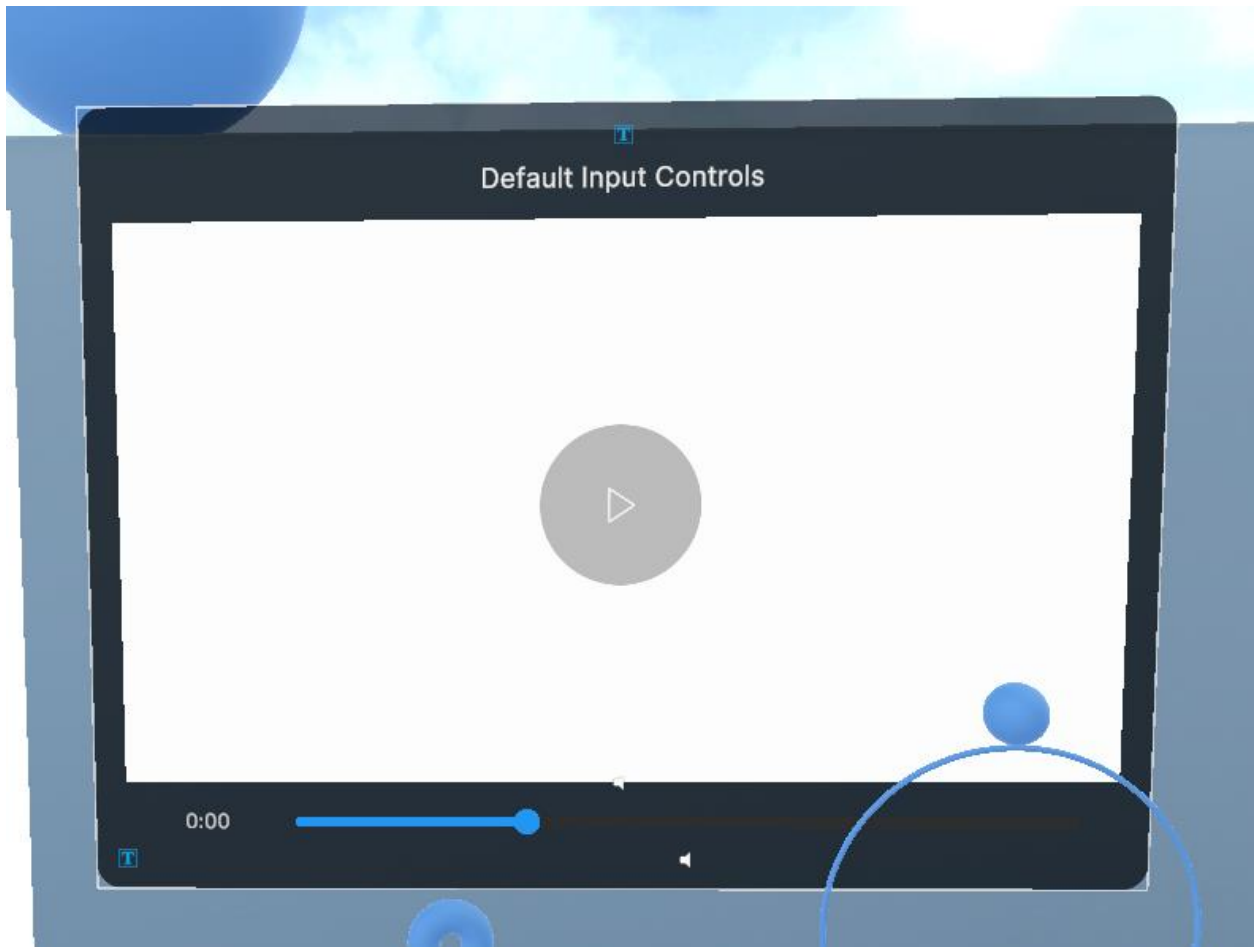


Рисунок 3.7 – Інформаційна панель користувача

Для перевірки механік взаємодії у процесі розроблення використовувався набір тестових тривимірних об'єктів. Серед них застосовувалися прості геометричні примітиви Unity: куби, сфери, циліндри та конуси. Використання таких моделей дозволяє швидко перевіряти коректність роботи систем захоплення, переміщення, обертання та фізичної взаємодії. На етапі тестування геометричні примітиви є більш зручними за

складні моделі комплектуючих ПК, оскільки дозволяють швидко виявляти помилки у логіці роботи контролерів та системі фізики (рис. 3.8).

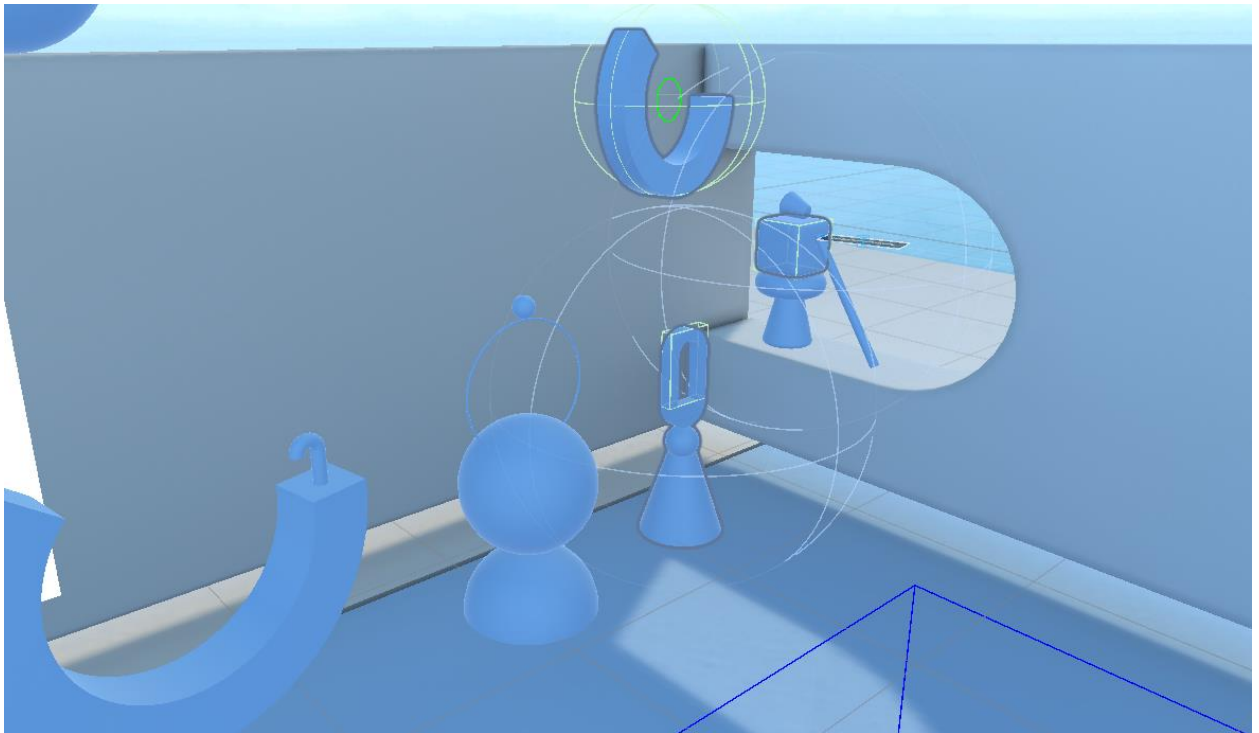


Рисунок 3.8 – Тестові об’єкти сцени

### 3.2.2 Оптимізація тривимірного простору та моделей

Оптимізація тривимірних моделей є обов’язковим етапом підготовки VR-застосунку. Через необхідність рендерингу сцени одночасно для двох очей VR-гарнітури та підтримки стабільної частоти кадрів (не менше 72–90 FPS) кожна модель проходить кілька стадій оптимізації у Blender та Unity.

У Blender виконується зниження кількості полігонів за допомогою модифікатора Decimate, видалення прихованих (невидимих) граней та об’єднання дублюючих вершин (Merge by Distance). Також для деталізованих компонентів (материнська плата, відеокарта) застосовується техніка запікання нормалей (Normal Baking): деталі геометрії переносяться з high-

poly моделі на normal map низькополігональної версії, що дозволяє зберегти візуальну якість без збільшення кількості полігонів [26].

У середовищі Unity для підвищення продуктивності застосовуються такі методи:

- Static Batching – об'єднання статичних об'єктів сцени (стіни, підлога) у спільний буфер рендерингу, що зменшує кількість draw calls;
- GPU Instancing – рендеринг великої кількості однакових об'єктів за один виклик;
- Baked Lighting – попереднє обчислення освітлення та збереження у lightmap-текстурах для статичних об'єктів сцени;
- Atlas Textures – об'єднання кількох текстур в один текстурний атлас для зменшення кількості звернень до GPU.

Для об'єднання статичних мешів сцени використовується скрипт MeshCombiner, який формує єдиний Mesh-файл із кількох окремих геометрій. Це суттєво зменшує навантаження на центральний процесор при рендерингу. Результати об'єднання мешів статичних об'єктів наведено на рисунку 3.9.

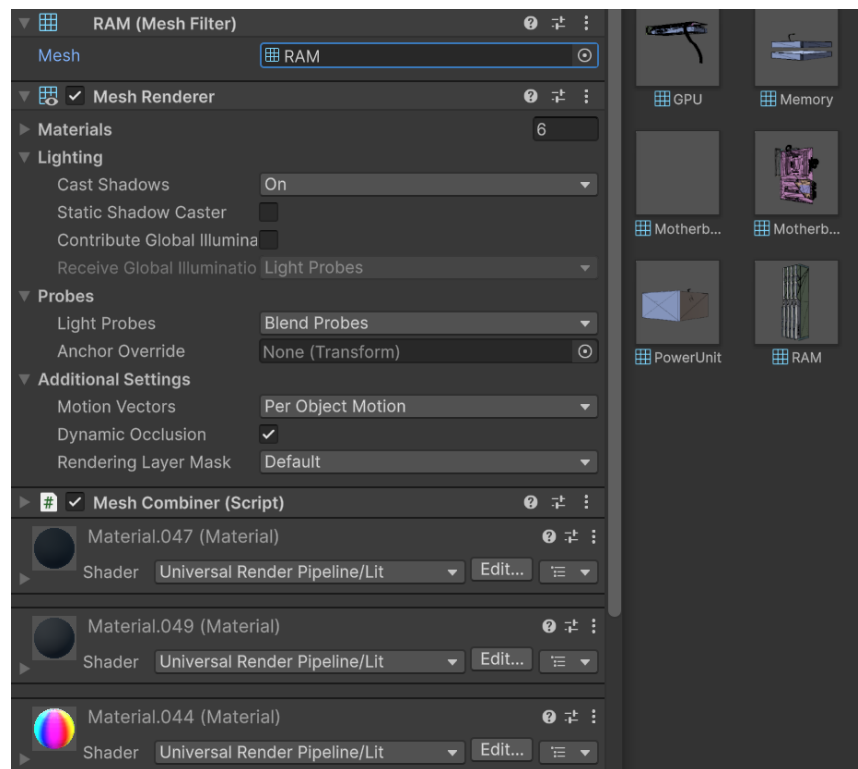


Рисунок 3.9 – Результат об'єднання мешів

Налаштування налаштування Static Batching у середовищі Unity представлено на рисунку 3.10.

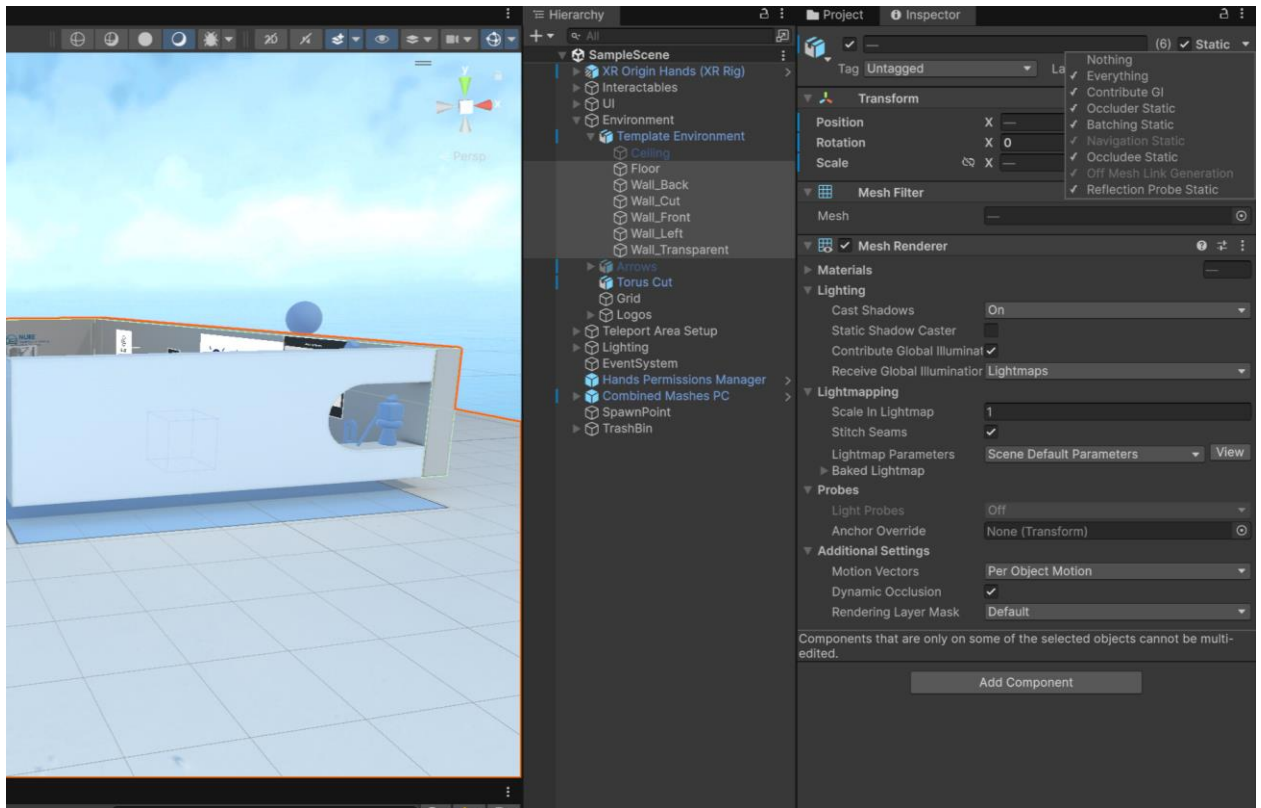


Рисунок 3.10 – Налаштування Static Batching

### 3.3 Програмна реалізація функціональних складових навчального симулятора

#### 3.3.1 Програмна реалізація контролера користувача у VR-середовищі

Основною складовою взаємодії користувача із VR-середовищем є система керування, побудована на базі XR Origin. Даний компонент виконує роль центрального вузла, який об'єднує камеру користувача, контролери та механізми переміщення у віртуальному просторі. XR Origin забезпечує автоматичне відстеження положення голови та рук користувача, що дозволяє створити природну взаємодію з об'єктами сцени. Ієрархію XR Origin у сцені Unity наведено на рисунку 3.11.

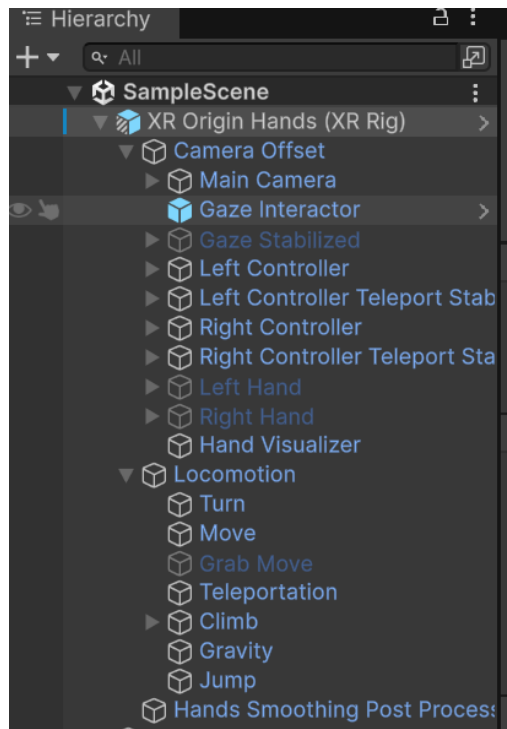


Рисунок 3.11 – Ієрархія XR Origin у сцені

До складу XR Origin входять такі елементи:

- головна камера користувача (Camera Offset / Main Camera);
- лівий та правий контролер (Left/Right Controller);
- система телепортації (Teleportation Provider);
- система фізичної взаємодії (Interaction Manager).

Структуру XR Origin та взаємозв'язки між його складовими представлено на рисунку 3.12.

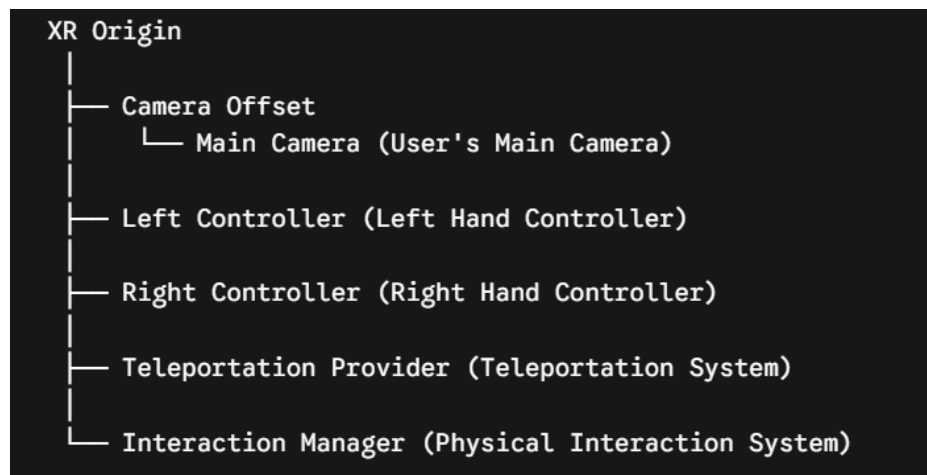


Рисунок 3.12 – Структура XR Origin

Для реалізації фізичної взаємодії із тривимірними об'єктами використовуються компоненти XR Direct Interactor та XR Grab Interactable. Вони дозволяють захоплювати об'єкти, утримувати їх у руках, переміщувати, обертати та використовувати під час складання комп'ютера. Після захоплення положення та орієнтація об'єкта автоматично синхронізуються з рухами VR-контролера, що забезпечує інтуїтивний процес взаємодії. Налаштування компонента XR Direct Interactor зображено на рисунку 3.13.

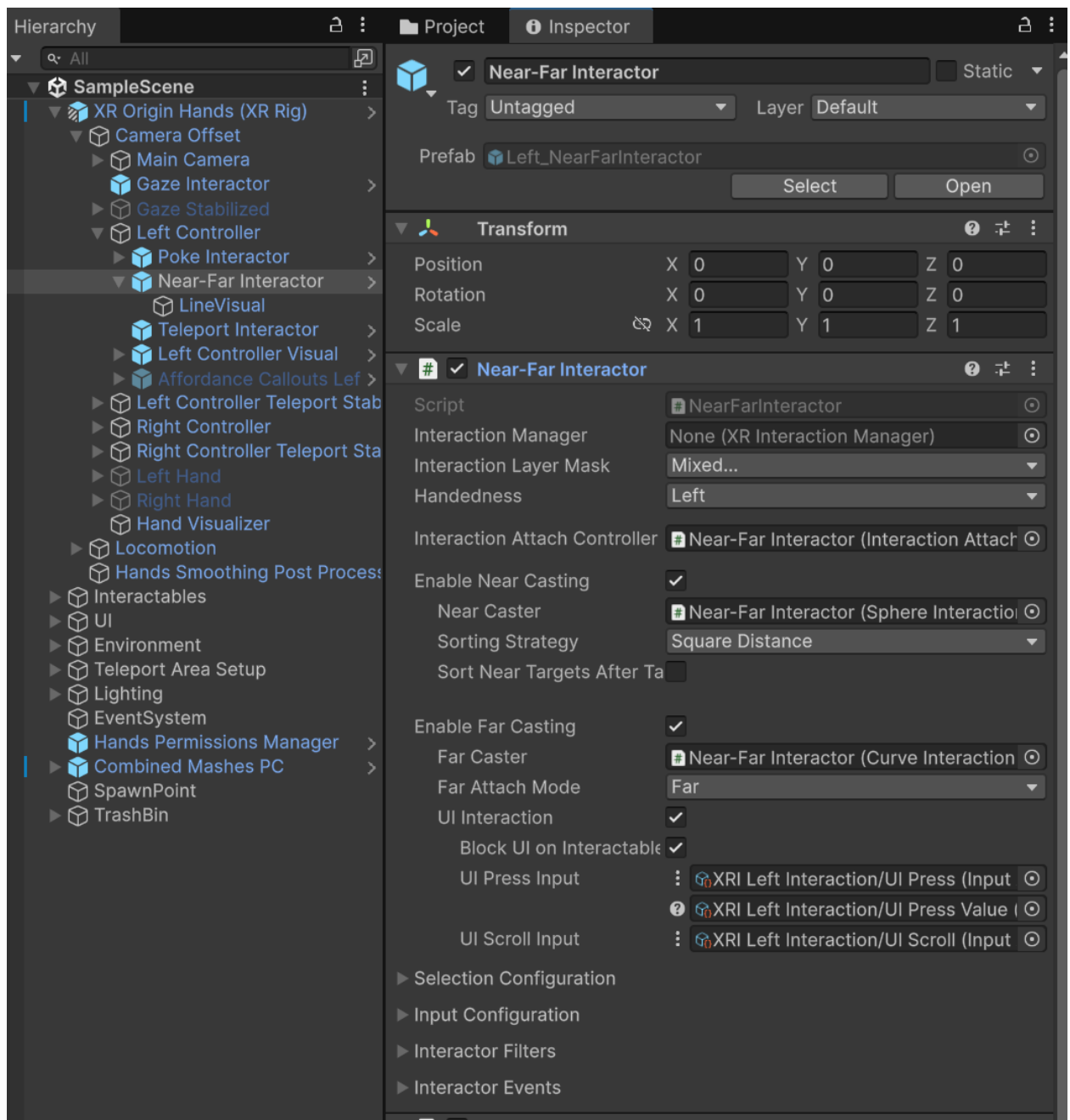


Рисунок 3.13 – Налаштування XR Direct Interactor

Після натискання кнопки захоплення об'єкт прив'язується до контролера та повторює його рухи. Приклад взаємодії з компонентом ПК у VR-середовищі наведено на рисунку 3.14.

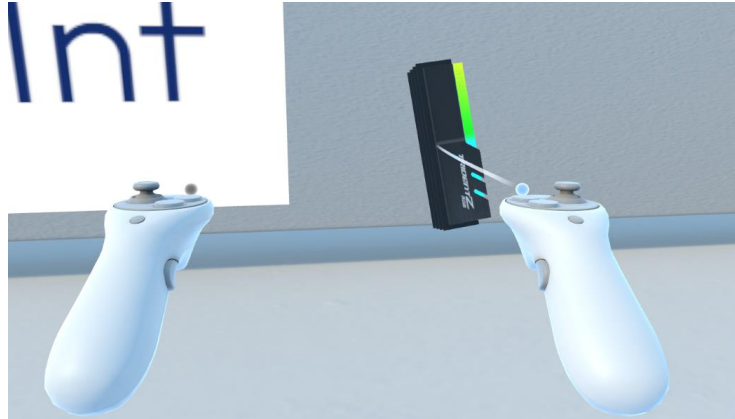


Рисунок 3.14 – Приклад взаємодії з компонентом ПК

Для навігації у VR-середовищі реалізовано механіку плавного переміщення (Continuous Locomotion). Даний підхід дозволяє користувачеві пересуватися по сцені за допомогою аналогового стіка контролера без необхідності фізичного переміщення у реальному просторі (рис. 3.15).

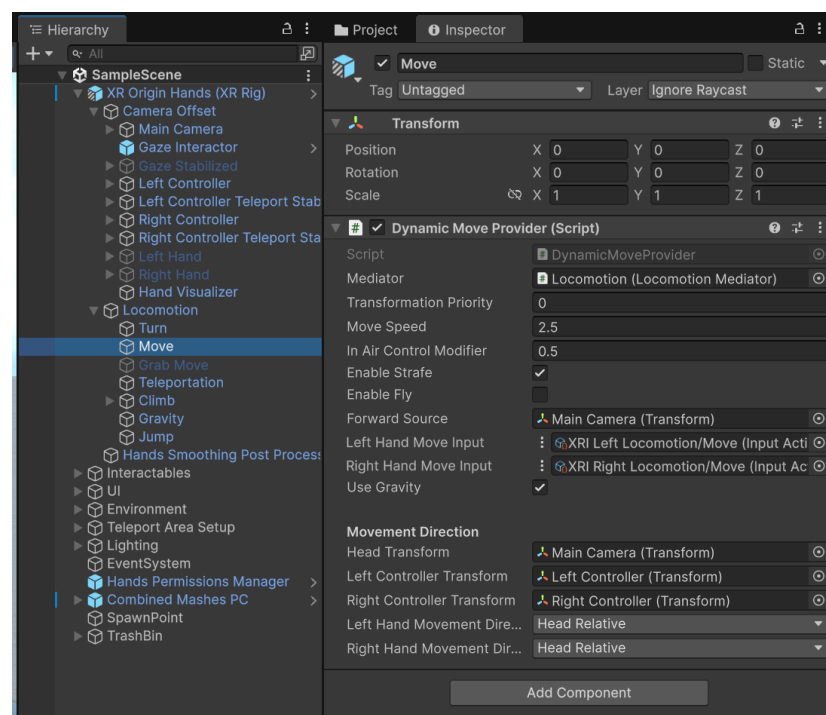


Рисунок 3.15 – Налаштування системи переміщення

Додатково реалізовано систему телепортації, яка є одним із найпоширеніших способів переміщення у VR. Використання телепортації дозволяє суттєво зменшити прояви кіберхвороби (VR sickness) та підвищити комфорт користувача під час тривалого використання застосунку (рис. 3.16).

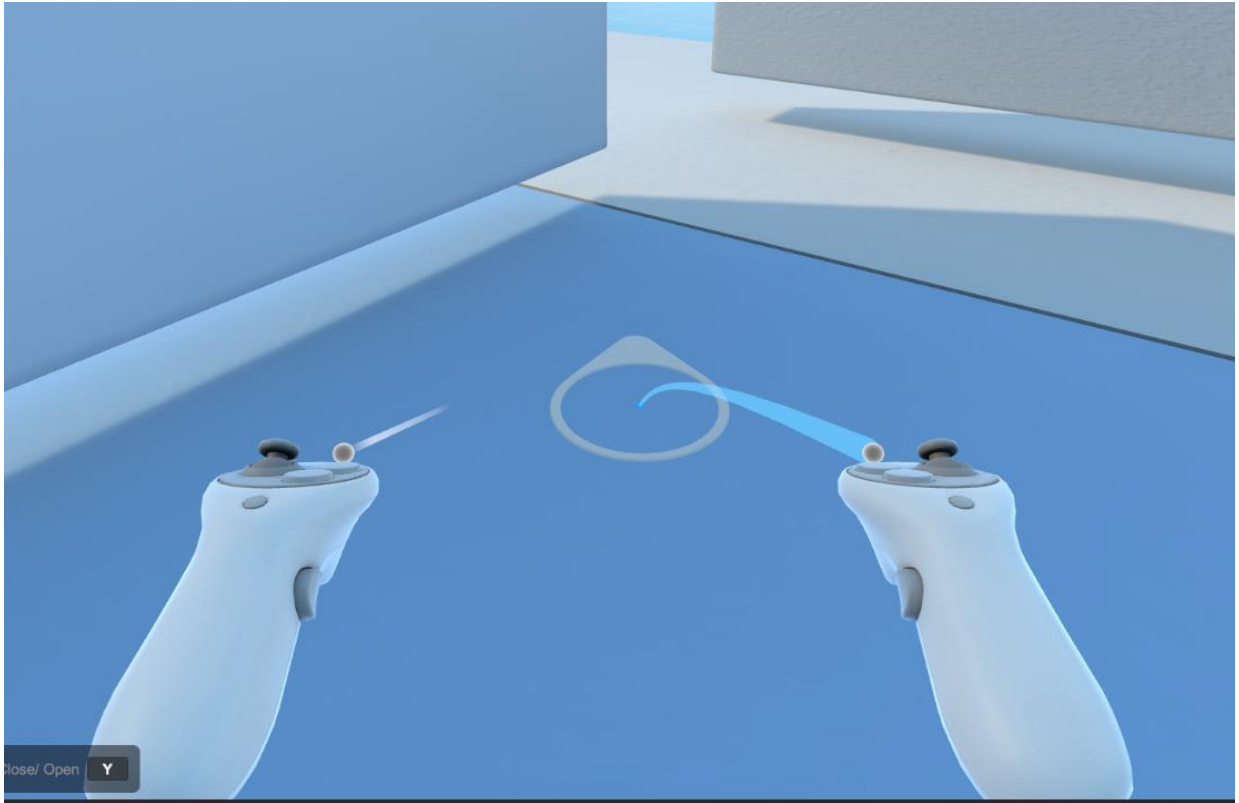


Рисунок 3.16 – Механіка телепортації

### 3.3.2 Програмування механік взаємодії з комплектуючими

Для забезпечення можливості багаторазового використання деталей реалізовано систему динамічного створення компонентів (Spawn System). Такий підхід дозволяє користувачеві самостійно формувати необхідний набір деталей для складання комп'ютера без потреби постійно зберігати усі об'єкти на сцені. На панелі користувача розміщено набір кнопок для створення окремих деталей, кожна кнопка викликає відповідний префаб із бібліотеки об'єктів (рис. 3.17).

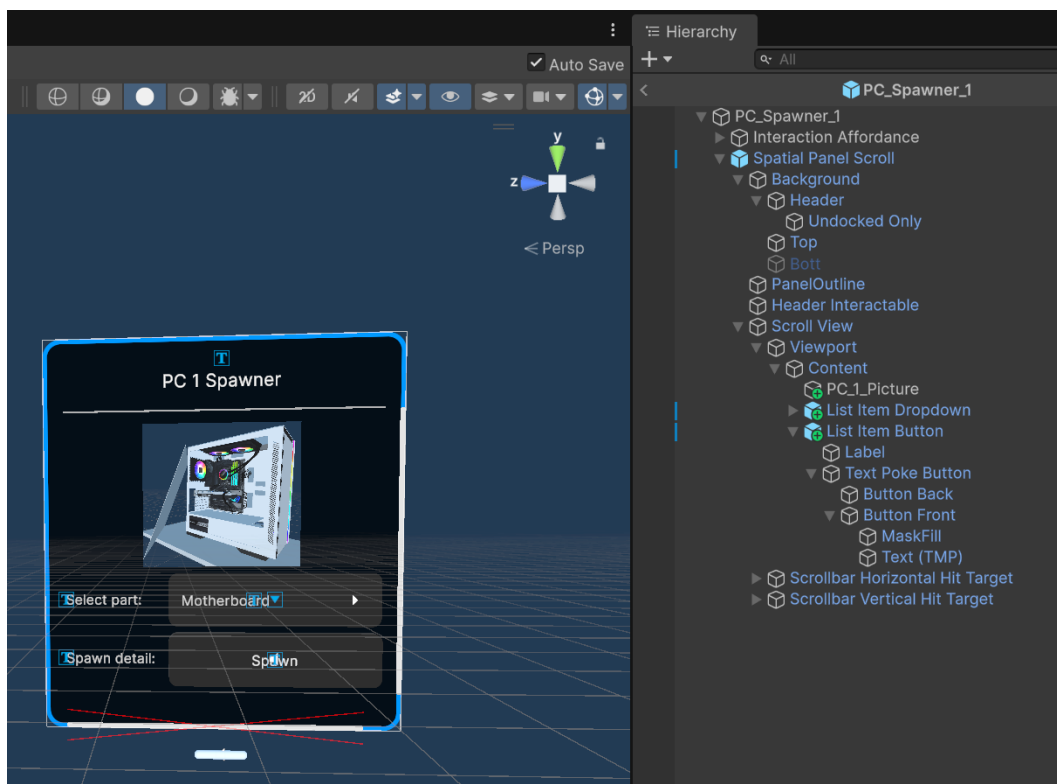


Рисунок 3.17 – Панель створення компонентів

Бібліотека префабів містить підготовлені моделі комплектуючих: процесори, модулі оперативної пам'яті, відеокарти, материнські плати та інші елементи системного блоку. Кожен префаб містить набір компонентів, необхідних для фізичної взаємодії та подальшого встановлення у відповідні сокети (рис. 3.18).

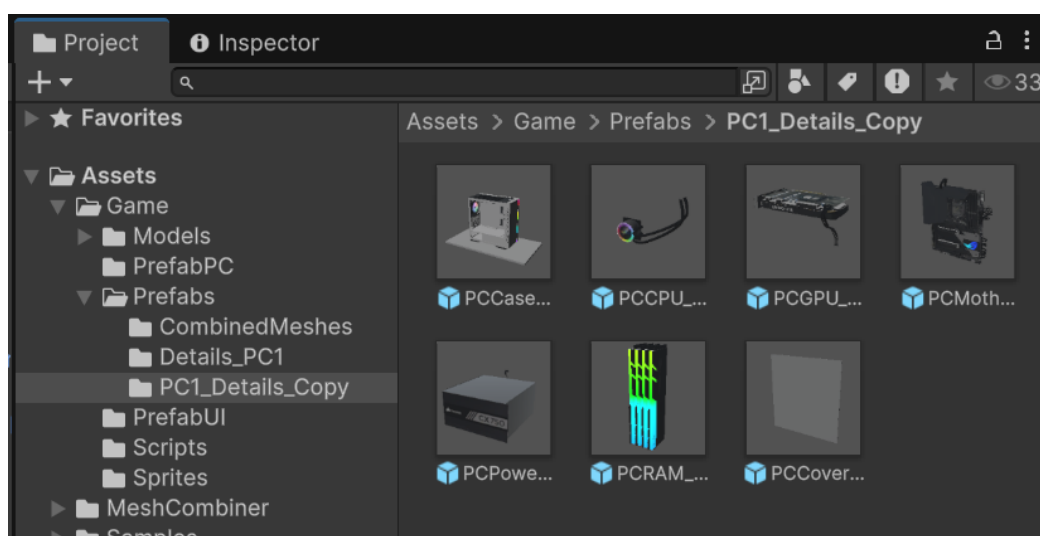


Рисунок 3.18 – Бібліотека префабів компонентів

Після натискання кнопки компонент створюється у визначеній точці сцени за допомогою стандартного методу Unity Instantiate [27]. Схему роботи системи динамічного створення компонентів наведено на рисунку 3.19.



Рисунок 3.19 – Схема створення компонентів

Центральною частиною навчального процесу є система встановлення комплектуючих у відповідні точки кріплення. Для реалізації даної функції використовується XR Socket Interactor, який дозволяє автоматично фіксувати

об'єкти у визначених позиціях після проходження перевірки сумісності. Такий підхід наближає процес взаємодії до реального складання персонального комп'ютера та спрощує позиціонування компонентів у VR-середовищі. Налаштування Socket Interactor наведено на рисунку 3.20.

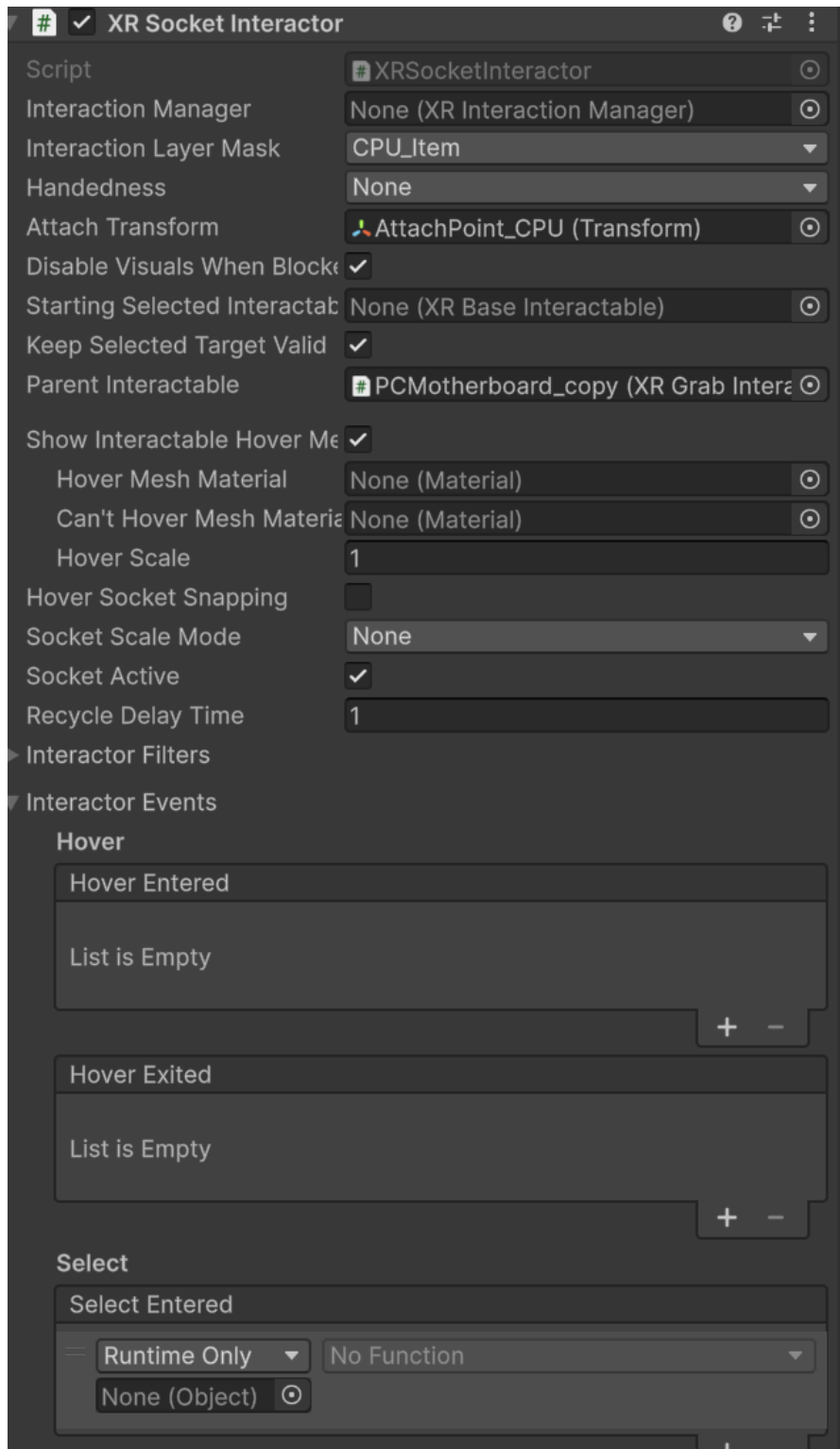


Рисунок 3.20 – Налаштування Socket Interactor

Для кожного слота материнської плати створено окрему Socket-зону з відповідним Layer Mask (рис. 3.21).

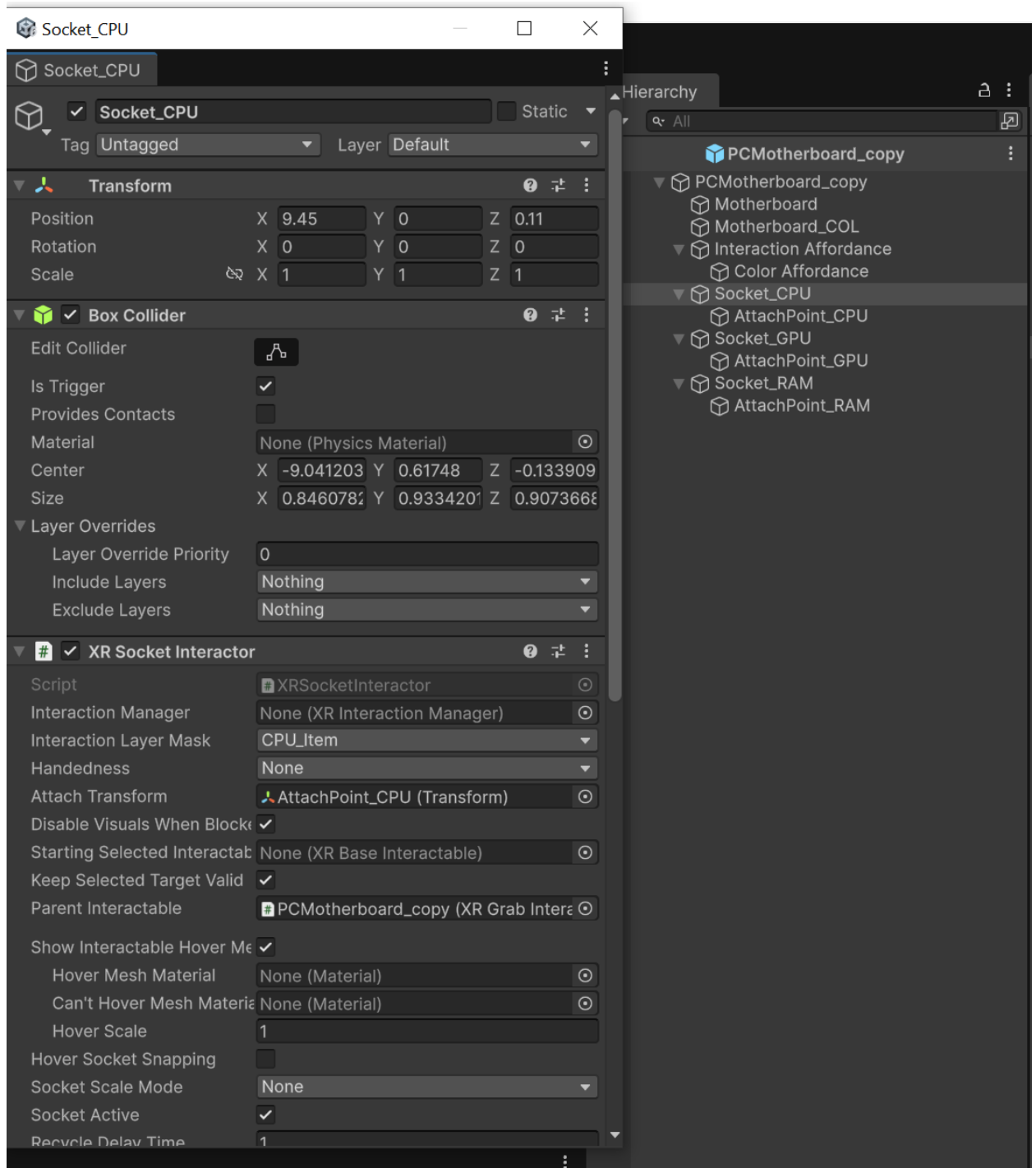


Рисунок 3.21 – Socket-зони материнської плати

Під час наближення компонента до сокета виконується перевірка сумісності за типом Layer: процесор може взаємодіяти лише із CPU Socket

Layer, модуль оперативної пам'яті – лише із RAM Slot Layer тощо. Процес перевірки сумісності компонентів зображено на рисунку 3.22.



Рисунок 3.22 – Процес перевірки сумісності

Після успішної перевірки система автоматично вирівнює компонент відносно точки встановлення та виконує його фіксацію. Подібний механізм використовується для всіх основних комплектуючих персонального комп'ютера. Приклади встановлення процесора, оперативної пам'яті та відеокарти наведено на рисунках 3.23–3.25 відповідно.

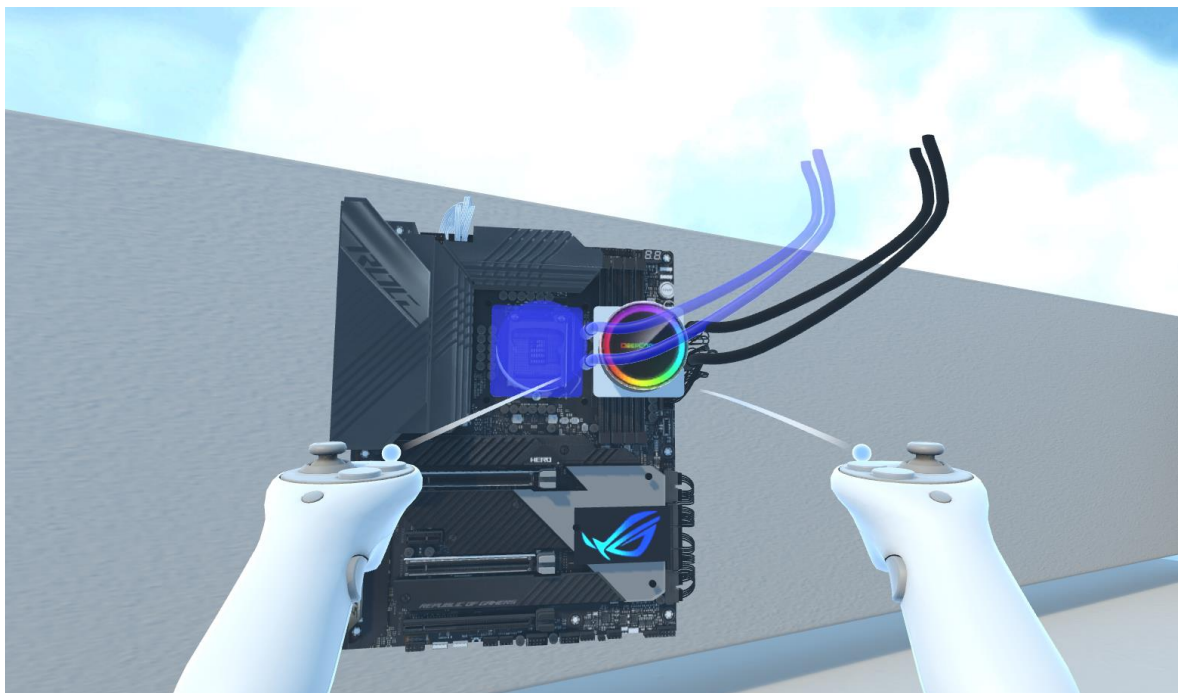


Рисунок 3.23 – Встановлення процесора у сокет

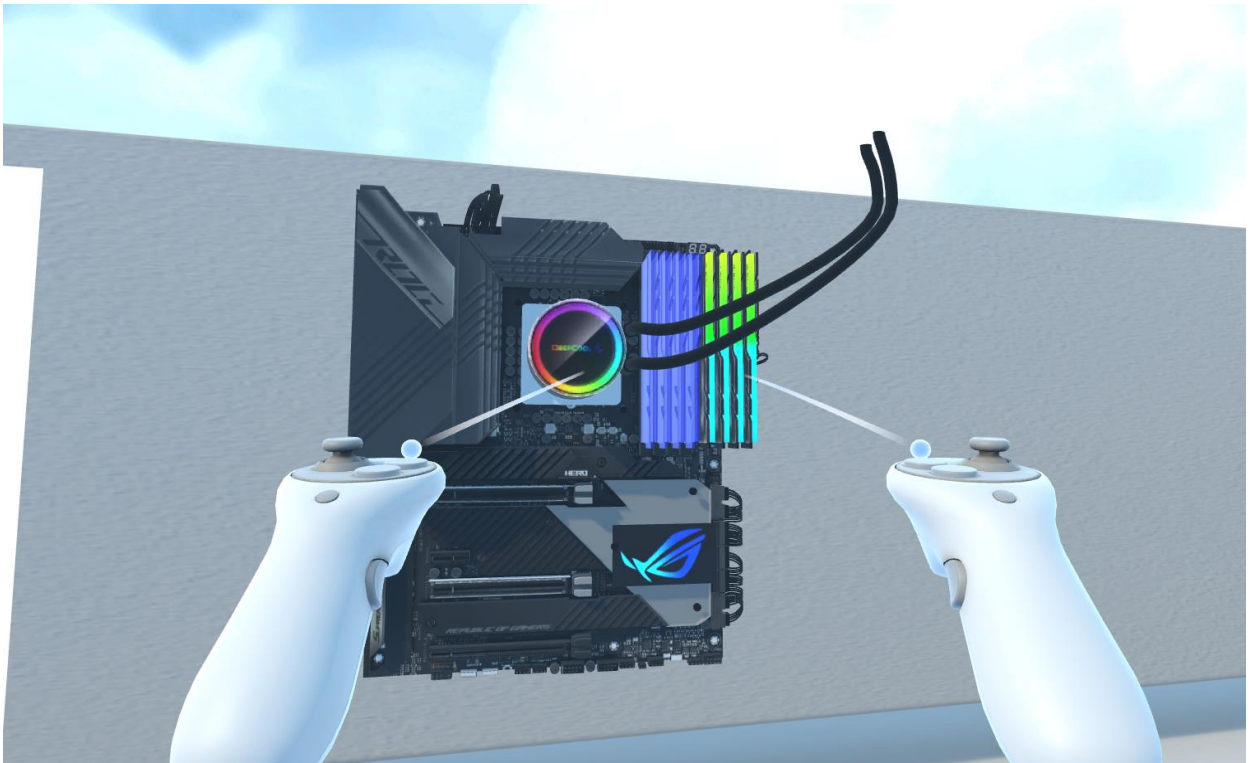


Рисунок 3.24 – Встановлення оперативної пам'яті

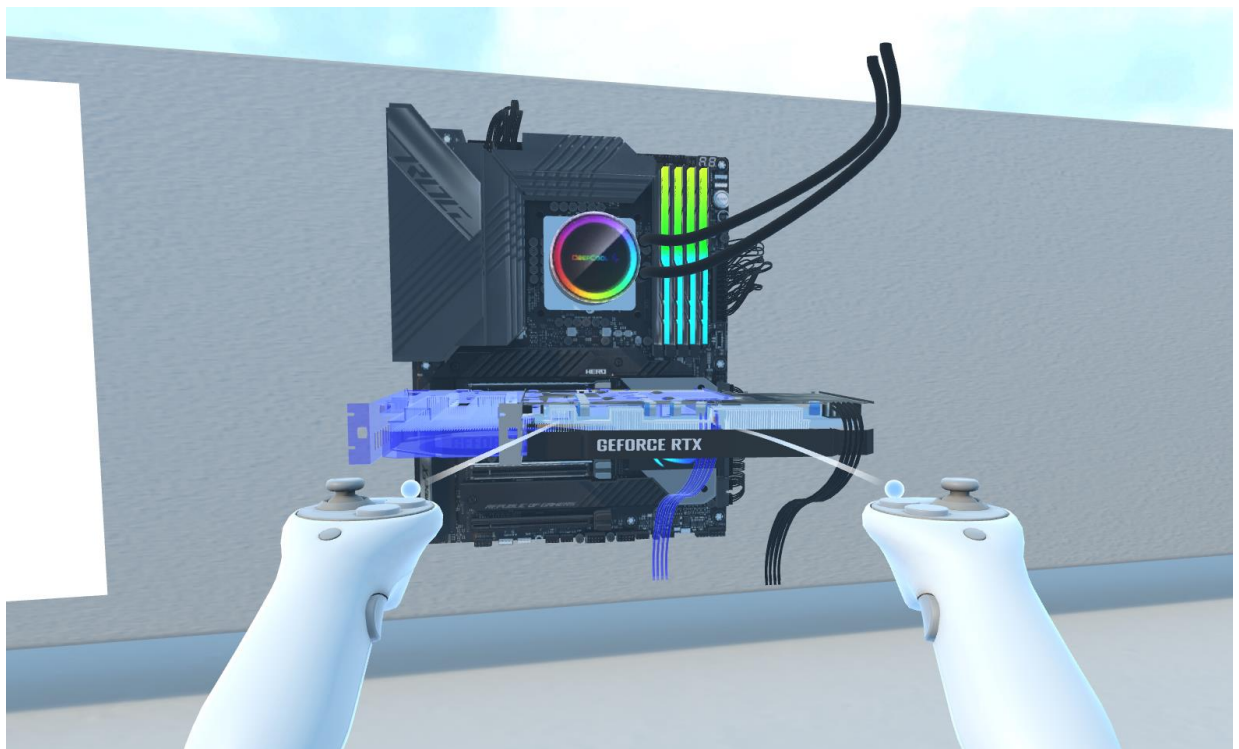


Рисунок 3.25 – Встановлення відеокарти

Для реалізації логіки перевірки сумісності та фільтрації взаємодій використовуються Layer Mask та Scriptable Objects (рис. 3.26).

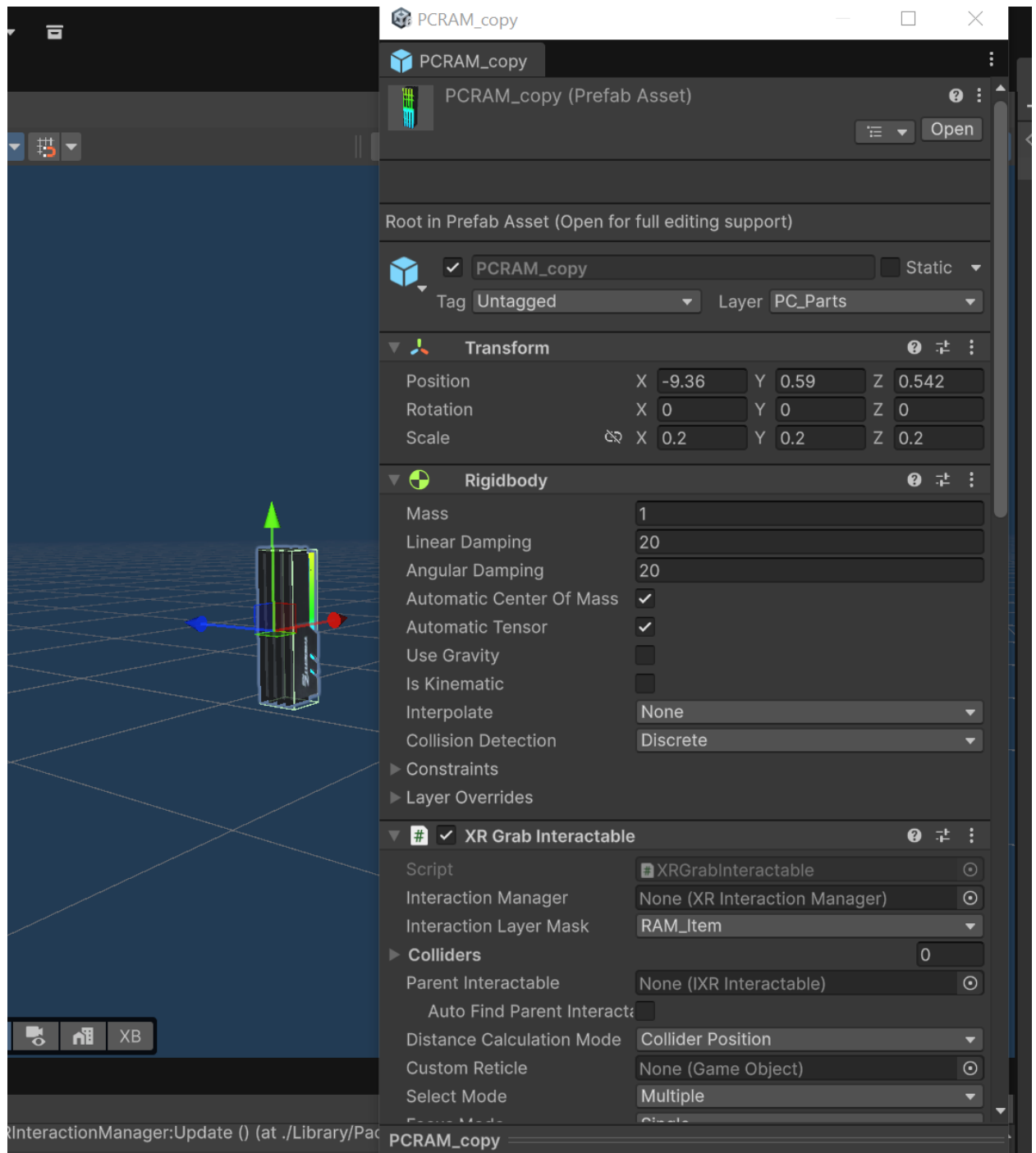


Рисунок 3.26 – Налаштування Layer Mask

Для запобігання перевантаженню сцени реалізовано механізм видалення непотрібних об'єктів. У VR-середовищі створено спеціальний контейнер для утилізації деталей: після потрапляння компонента до тригерної області виконується його автоматичне знищення (Destroy). Контейнер видалення компонентів зображено на рисунку 3.27, налаштування

тригерної зони – на рисунку 3.28, схему роботи системи видалення – на рисунку 3.29.



Рисунок 3.27 – Контейнер видалення компонентів

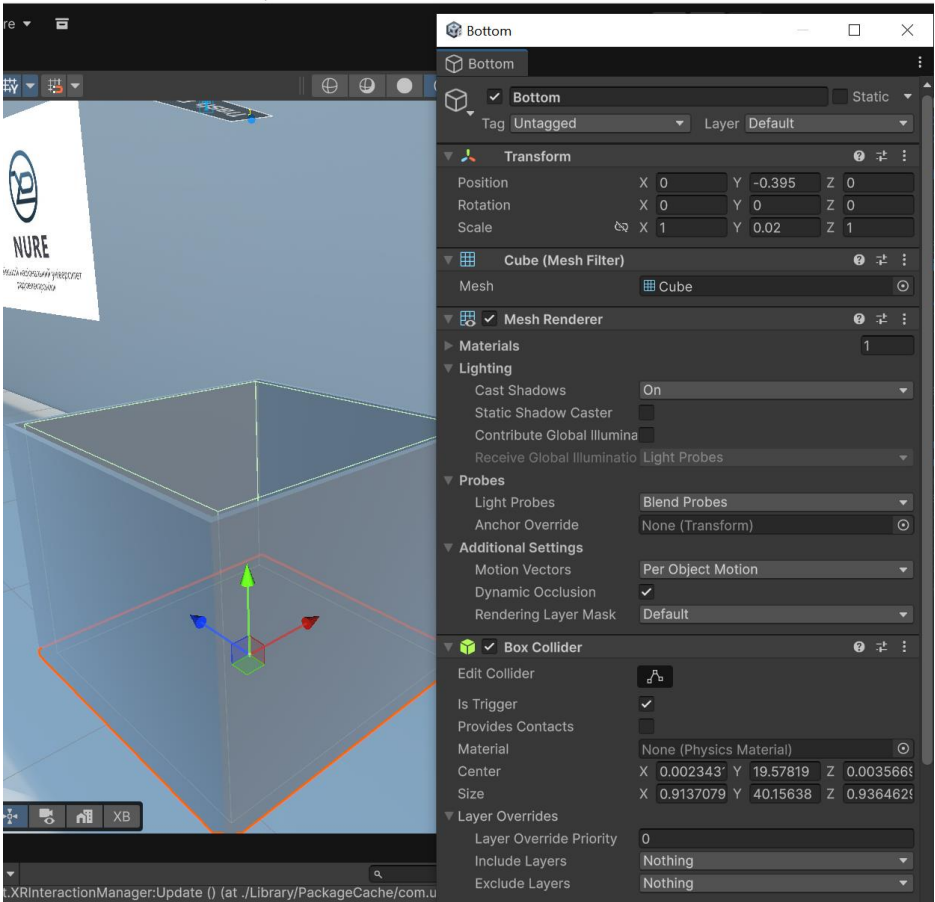


Рисунок 3.28 – Налаштування тригерної області



Рисунок 3.29 – Схема роботи системи видалення

### 3.3.3 Реалізація системи інтерактивних підказок, UI-панелей та аудіовізуального зворотного зв'язку

Для підтримки навчального процесу у застосунку реалізовано систему інтерактивних підказок та аудіовізуального зворотного зв'язку. Система орієнтована на те, щоб користувач міг самостійно освоїти процес складання ПК без зовнішнього супроводу викладача.

Візуальні підказки реалізовані за допомогою компонентів Canvas та TextMeshPro у World Space. Підказки відображаються у вигляді UI-панелей, які розміщені безпосередньо у просторі сцени поруч із відповідними

об'єктами взаємодії. Це дозволяє користувачеві отримувати контекстні інструкції, не виходячи з режиму VR.

Підсвічування сумісних Socket-зон під час наближення компонента реалізовано через зміну матеріалу та кольору Renderer компонента. Коли сумісний об'єкт потрапляє у зону дії Socket Interactor, колір підсвічування змінюється з нейтрального на голубий, що є візуальним сигналом для користувача про можливість фіксації деталі. У разі несумісності – колір підсвічування залишається нейтральним.

Аудіовізуальний зворотний зв'язок реалізовано за допомогою компонента AudioSource Unity. Для кожної ключової дії (успішне захоплення компонента, встановлення деталі у сокет, спроба встановлення несумісного компонента) відтворюється відповідний звуковий сигнал. Це підвищує імерсивність взаємодії та надає користувачеві чіткий зворотний зв'язок без необхідності стежити за текстовими повідомленнями. Загальну структуру UI-панелей та систему підказок у сцені зображено на рисунку 3.30.

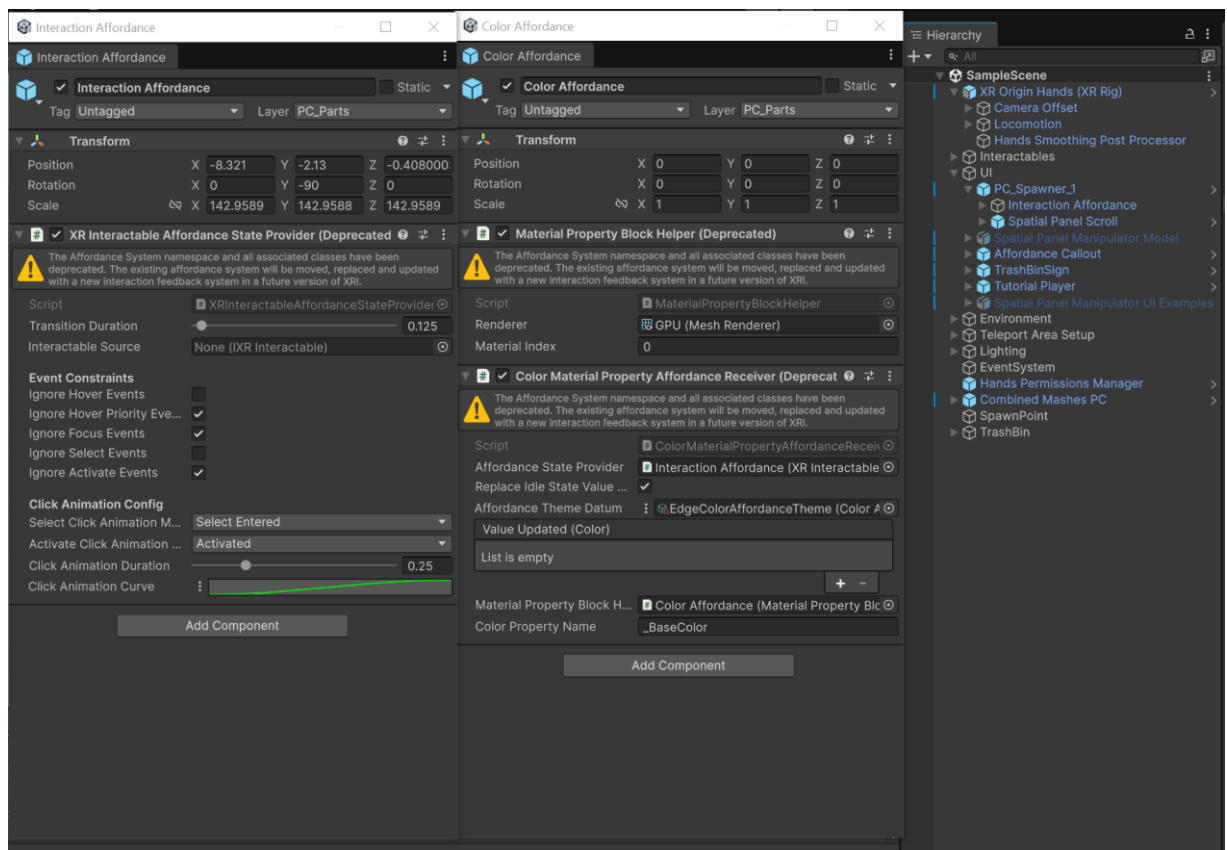


Рисунок 3.30 – UI-панелі та система підказок у сцені

### 3.4 Тестування розробленого застосунку та оцінювання продуктивності

Для перевірки працездатності системи та відповідності розробленого застосунку висунутим вимогам проведено функціональне тестування та оцінювання продуктивності.

У ході функціонального тестування перевірялась коректність роботи таких підсистем:

- динамічне створення компонентів (Spawn System);
- захоплення та переміщення об'єктів за допомогою VR-контролерів;
- переміщення користувача (Continuous Locomotion);
- телепортація у VR-середовищі;
- встановлення компонентів у Socket-зони;
- перевірка сумісності компонентів (Layer Mask);
- видалення компонентів через тригерну зону.

Результати функціонального тестування наведено у таблиці 3.2.

Таблиця 3.2 – Результати функціонального тестування

| Функція, що тестується              | Результат | Примітки                              |
|-------------------------------------|-----------|---------------------------------------|
| Динамічне створення компонентів     | Успішно   | Компонент з'являється у заданій точці |
| Захоплення та переміщення об'єктів  | Успішно   | Об'єкт відстежує рух контролера       |
| Переміщення (Continuous Locomotion) | Успішно   | Рух відповідає напрямку стіка         |
| Телепортація                        | Успішно   | Миттєве переміщення без артефактів    |
| Встановлення компонентів у сокет    | Успішно   | Фіксація та вирівнювання коректні     |
| Перевірка сумісності компонентів    | Успішно   | Несумісні деталі не фіксуються        |
| Видалення компонентів               | Успішно   | Об'єкт знищується при вході у зону    |

Тестування встановлення материнської плати зображено на рисунку 3.31, тестування встановлення кришки корпусу – на рисунку 3.32.



Рисунок 3.31 – Тестування материнської плати

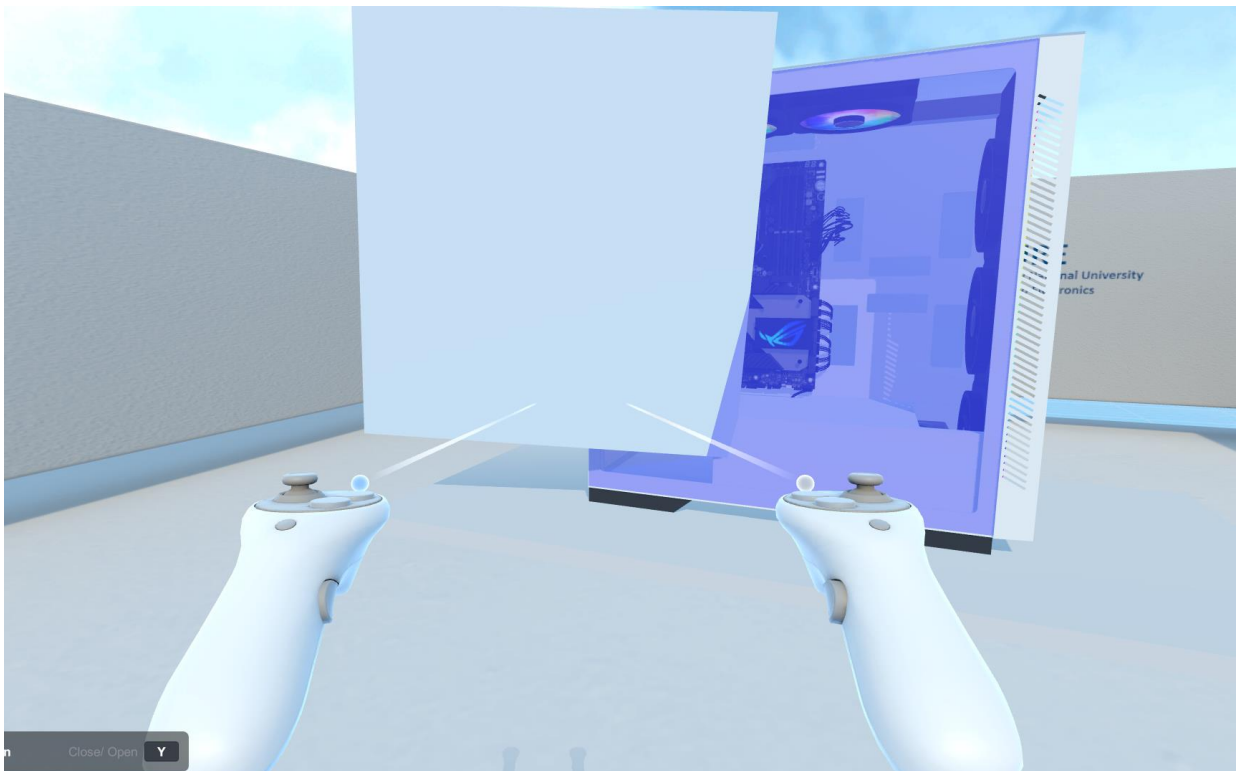


Рисунок 3.32 – Тестування встановлення кришки корпусу

Тестування роботи системи видалення компонентів наведено на рисунку 3.33.



Рисунок 3.33 – Тестування роботи системи видалення

Для оцінювання продуктивності застосунку використовувався вбудований інструмент Unity Profiler. Аналіз виконувався у двох режимах: без застосування методів оптимізації та після їх застосування (рис. 3.34–3.35).

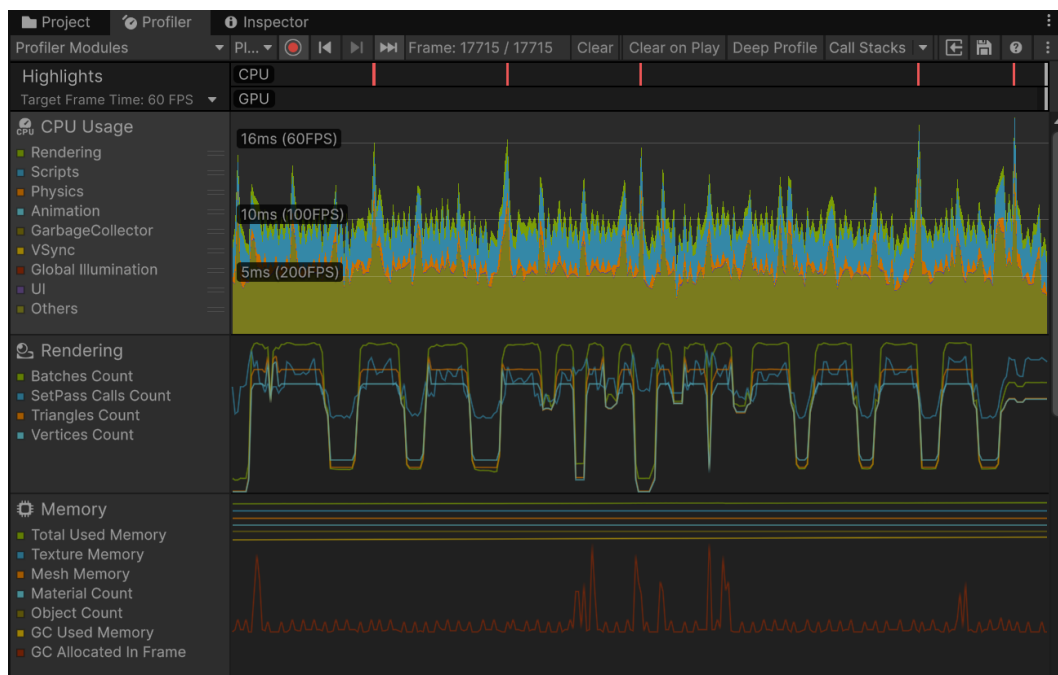


Рисунок 3.34 – Вікно Unity Profiler до оптимізації

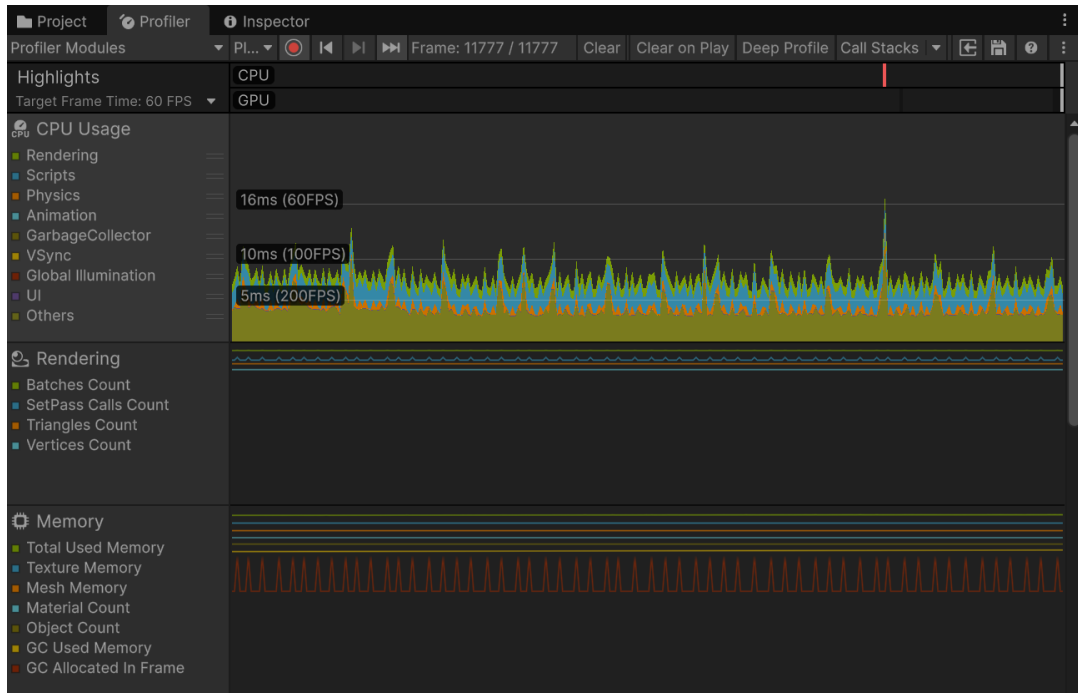


Рисунок 3.35 – Вікно Unity Profiler після оптимізації

За результатами профілювання встановлено, що без застосування оптимізацій частота кадрів у VR-сцені зі всіма завантаженими моделями комплектуючих становила в середньому 58–65 FPS, що є недостатнім для комфортної роботи у VR [28]. Після застосування комплексу методів оптимізації (Static Batching, Occlusion Culling, MeshCombiner, компресія текстур) частота кадрів стабілізувалась на рівні 90–110 FPS, що відповідає вимогам до VR-застосунків. Усі перевірені функції працюють відповідно до поставлених вимог.

### 3.5 Аналіз результатів практичного застосування та перспективи подальшого вдосконалення системи

Розроблений VR-застосунок для моделювання складання персонального комп'ютера пройшов функціональне тестування та підтвердив свою практичну цінність як навчальний інструмент. Усі ключові механіки –

від динамічного створення та захоплення компонентів до перевірки їх сумісності та фіксації у відповідних сокетах – функціонують коректно та забезпечують природну взаємодію користувача з VR-середовищем.

Використання XR Interaction Toolkit та стандарту OpenXR забезпечує сумісність застосунку із сучасними VR-гарнітурами різних виробників, що підвищує потенційну аудиторію користувачів. Система перевірки сумісності на основі Layer Mask ефективно запобігає некоректному складанню, формуючи у користувача правильні практичні навички. Впровадження аудіовізуального зворотного зв'язку та системи підказок підтвердило свою доцільність: навіть без попередньої підготовки тестові користувачі успішно виконували базові операції складання ПК протягом перших 5–10 хвилин роботи з застосунком.

Застосовані методи оптимізації тривимірної графіки (MeshCombiner, Static Batching, Occlusion Culling, компресія текстур) дозволили досягти стабільної частоти кадрів на рівні 90–110 FPS, що відповідає вимогам до VR-середовищ та забезпечує комфортну роботу без виникнення дискомфорту [29].

Подальший розвиток програмного засобу може бути спрямований у кількох напрямках:

- розширення бібліотеки комплектуючих персонального комп'ютера – додавання нових моделей процесорів, материнських плат різних форм-факторів, відеокарт, накопичувачів та блоків живлення, що дозволить створювати більшу кількість конфігурацій ПК та наблизити навчальний процес до реальних умов;

- впровадження покрокових навчальних сценаріїв – реалізація режиму, у якому система супроводжує користувача під час виконання окремих етапів складання, надає рекомендації щодо послідовності дій та контролює їх правильність;

- реалізація системи оцінювання дій користувача – автоматичний аналіз правильності встановлення компонентів, фіксація кількості допущених

помилки, визначення часу виконання завдання та формування підсумкових результатів навчання;

- впровадження голосових підказок та розширеного аудіосупроводу – голосові інструкції для пояснення етапів складання та повідомлення про помилки підвищують доступність застосунку;

- використання адаптивних механізмів навчання – на основі аналізу попередніх дій система зможе автоматично змінювати складність завдань та формувати персоналізовані навчальні сценарії.

Таким чином, розроблений VR-застосунок є функціональним навчальним засобом, який успішно вирішує поставлені завдання та має значний потенціал для подальшого масштабування і впровадження в освітній процес вищих та професійно-технічних навчальних закладів.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений та реалізований імерсивний VR-застосунок у середовищі Unity для моделювання навчального процесу зі складання персональних комп'ютерів. Робота охоплює повний цикл створення програмного продукту – від аналізу предметної області до програмної реалізації інтерактивних механік та тестування оптимізованого застосунку.

Для досягнення поставленої мети були вирішені такі завдання:

- проведено аналіз сучасного стану використання VR-технологій в освіті та існуючих симуляторів, що дало можливість виявити їхні обмеження (зокрема нестачу просторової взаємодії у традиційних 2D-симуляторах) та сформулювати вимоги до нового рішення;

- здійснено моделювання архітектури застосунку, а також змодельовано системні зв'язки між фізичними властивостями об'єктів та логікою збирання комп'ютера;

- розроблено алгоритми перевірки сумісності моделей комплектуючих персонального комп'ютера та математичні моделі позиціонування об'єктів у тригерних зонах;

- створено тривимірне навчальне середовище, імпортовано та оптимізовано моделі комп'ютерних комплектуючих із застосуванням теоретичних і практичних підходів до оптимізації геометрії;

- розроблено повноцінний VR-застосунок, що поєднує механіки захоплення деталей, систему інтерактивних підказок та аудіовізуального зворотного зв'язку.

Розглянуто існуючі застосунки зі схожим функціоналом та проаналізовано літературні джерела, що стосуються задач створення VR-застосунків.

Проведено тестування розробленого застосунку, зокрема перевірку логіки відпрацювання алгоритмів сумісності компонентів, точності роботи

системи колізій та оцінювання продуктивності візуалізації. Застосування методів оптимізації тривимірної графіки забезпечило стабільну частоту кадрів, необхідну для комфортної роботи користувача у VR-гарнітурі.

Розглянуто перспективи подальшого масштабування розробленого рішення, які включають додавання режиму діагностики апаратних несправностей, розширення бази комп'ютерних комплектуючих, створення системи завдань та перевірки їх виконання.

Результати кваліфікаційної роботи можуть бути корисними і можуть бути впроваджені в освітній процес закладів вищої та професійно-технічної освіти для підготовки фахівців [30]. Розроблений інструмент дозволяє безпечно відпрацьовувати просторові навички роботи з апаратним забезпеченням, знижує ризики пошкодження реального дороговартісного обладнання та підвищує загальну ефективність засвоєння матеріалу завдяки ефекту повної присутності.

Результати роботи апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «Радіoeлектроніка і молодь у XXI столітті» [31], Міжнародної науково-практичної студентської конференції «ІТ-простір сьогодення: тенденції, іновації та перспективи розвитку» [32].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & education*, 147, 103778.
2. Mazuryk, T., & Gervautz, M. (1996). History, applications, technology and future. *Virtual Reality*, 72(4), 486-497. Vienna University of Technology.
3. LaValle, S. M. (2023). *Virtual reality*. Cambridge university press.
4. Mystakidis, S. M. (2022). Encyclopedia 2022, 2, 486–497.
5. Makransky, G., & Lilleholt, L. (2018). A structural equation modeling investigation of the emotional value of immersive virtual reality in education. *Educational Technology Research and Development*, 66(5), 1141-1164.
6. Brookes, J., Warburton, M., Alghadier, M., Mon-Williams, M., & Mushtaq, F. (2020). Studying human behavior with virtual reality: The Unity Experiment Framework. *Behavior research methods*, 52(2), 455-463.
7. Bondarenko, Y., Kuts, V., Pizzagalli, S., Nutonen, K., Murray, N., & O'Connell, E. (2023, June). Universal XR framework architecture based on open-source XR tools. In *International XR Conference* (pp. 87-98). Cham: Springer Nature Switzerland.
8. Erleben, K., Sparring, J., Henriksen, K., & Dohlmann, H. (2005). *Physics-based animation* (Vol. 79). Hingham: Charles River Media.
9. Gregory, J. (2018). *Game engine architecture*. AK Peters/CRC Press.
10. Freina, L., & Ott, M. (2015, April). A literature review on immersive virtual reality in education: state of the art and perspectives. In *The international scientific conference elearning and software for education* (Vol. 1, No. 133, pp. 10-1007).
11. Irregular Corporation. (2018). PC Building Simulator [Video game]. Irregular Corporation. URL:

[https://store.steampowered.com/app/621060/PC\\_Building\\_Simulator/](https://store.steampowered.com/app/621060/PC_Building_Simulator/) (дата звернення 11.04.2026)

12. PCPartPicker. (2024). PCPartPicker – Build guides and system builder. URL: <https://pcpartpicker.com> (дата звернення 10.04.2026).

13. Tvoroshenko, I., Gorokhovatskyi, V., Kobylin, O., & Tvoroshenko, A. (2023). Application of deep learning methods for recognizing and classifying culinary dishes in images. *International Journal of Academic and Applied Research*, 7(9), 57–70.

14. Unity Technologies. (2023). Unity XR Interaction Toolkit documentation (version 2.5). URL: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.5> (дата звернення 05.04.2026).

15. Hocking, J. (2022). *Unity in action: multiplatform game development in C*. Simon and Schuster. Manning Publications.

16. Schell, J. (2008). *The Art of Game Design: A book of lenses*. CRC press.

17. Rebenitsch, L., & Owen, C. (2016). Review on cybersickness in applications and visual displays. *Virtual reality*, 20(2), 101-125.

18. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description. *Advances in Electrical and Electronic Engineering*, 21(1), 19-27.

19. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Tools for fast metric data search in structural methods for image classification. *IEEE Access*, 10, 124738-124746.

20. Гороховатський, В. О., & Творошенко, І. С. (2021). *Методи інтелектуального аналізу та оброблення даних: навч. посібник*. ХНУРЕ.

21. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2023). Statistical data analysis models for determining the relevance of structural image descriptions. *IEEE Access*, 11, 126938-126949.

22. Gorokhovatskyi, V., & Tvoroshenko, I. (2024). Transforming image descriptions as a set of descriptors to construct classification features. *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), 113–125.
23. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O., Hudáková, M., & Gorokhovatskyi, O. (2024). Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set. *IEEE Access*, 12, 73376-73385.
24. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks. *International Journal of Academic Information Systems Research*, 7(7), 25–36.
25. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Handwritten character recognition models based on convolutional neural networks. *International Journal of Academic Engineering Research*, 7(9), 64–72.
26. Gorokhovatskyi, V., Chmutov, Y., Tvoroshenko, I., & Kobylín, O. (2025). Reducing computational costs by compressing the structural description in image classification methods. *Advanced Information Systems*, 9(1), 5-12.
27. Tvoroshenko, I., Pomazan, V., Gorokhovatskyi, V., & Kobylín, O. (2023). Application of video data classification models using convolutional neural networks. *International Journal of Academic and Applied Research*, 7(11), 134–145.
28. Bohdan, N., Tvoroshenko, I., Gorokhovatskyi, V., & Kobylín, O. (2025). Development of a hybrid method to enhance context memory for a chatbot application based on large language models. *International Journal of Academic Information Systems Research*, 9(10), 7–18.
29. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O., & Hudáková, M. (2026). Feature space quantization and application of metrics in structural methods of image recognition. *IEEE Access*, 14.

30. Гороховатський, В. О., & Творошенко, І. С. (2026). Продуктивні моделі аналізу даних у методах розпізнавання зображень: монографія. ХНУРЕ.

31. Косовський С.Б., (2025) Використання ігрового рушію Unity для створення AR та VR освітньо-ігрових програм. *IT-простір сьогодення: тенденції, інновації та перспективи розвитку. Тези II міжнародної науково-практичної студентської конференції (15 жовтня 2025). Харків: ХНУ імені В. Н. Каразіна*, С. 389-392.

32. Косовський С.Б., (2026) Використання ігрового рушію Unity для створення AR та VR освітньо-ігрових програм. *Радіoeлектроніка і молодь у XXI столітті. Тези тридцятого міжнародного молодіжного форуму (22-24 квітня 2026). Харків: ХНУРЕ*, С. 83-85.