МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп’ютерної інженерії

Баглей Валерій Ростиславович

Програмний модуль для інтеграції модуля відображення елементів віртуальної реальності із третьосторонніми системами / Software module for integration of the virtual reality elements display module with third-party systems

спеціальність: 123 – Комп’ютерна інженерія
освітньо-професійна програма – Комп’ютерна інженерія

Випускна кваліфікаційна робота

Виконав: студент групи КІ-41
Баглей Валерій Ростиславович

Керівник
Піцун О.Й.

2025

АНОТАЦІЯ

Баглей В. Програмний модуль для інтеграції модуля відображення елементів віртуальної реальності із третьосторонніми системами. – Дипломна робота.

Робота присвячена актуальній проблемі розробки програмного забезпечення для інтеграції модулів відображення елементів віртуальної (VR) та розширеної (XR) реальності з існуючими третьосторонніми системами. Розглянуто зростаючу важливість XR-технологій у різних галузях, таких як медицина, освіта та виробництво, та виклики, пов'язані із забезпеченням їхньої сумісності та розширенням функціональності через інтеграцію.

Робота написана обсягом 45 сторінок, містить 10 рисунків, 1 таблицю, 2 додатки та 20 джерел за переліком посилань. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою роботи є розробка підходу та програмного модуля, що забезпечує інтеграцію системи відображення елементів віртуальної реальності з третьосторонніми системами, зокрема з використанням стандарту OpenXR, для підвищення гнучкості та функціональності XR-додатків.

У ході роботи було проведено аналіз сучасних підходів до розробки XR-систем, досліджено структуру та принципи функціонування стандарту OpenXR, включаючи життєвий цикл сесії та механізми розширень. Розроблено алгоритм визначення переліку активних API слоїв та розширень OpenXR.

Результатом роботи є розроблений програмний модуль SlicerVirtualReality для інтеграції XR-функціоналу в медичну платформу 3D Slicer. Продемонстровано практичну реалізацію підтримки специфічного XR-обладнання (контролер Magic Leap 2) шляхом створення та інтеграції відповідних файлів прив'язки (binding files) для OpenXR, що забезпечує коректну взаємодію користувача в XR-середовищі.

Практичне значення полягає у створенні програмного модуля, що дозволяє інтегрувати сучасні XR-можливості в існуючі складні програмні продукти, такі як 3D Slicer, розширюючи їх функціональність для медичної візуалізації, планування та навчання. Запропонований підхід сприяє стандартизації процесу інтеграції та полегшує розробку кросплатформенних XR-рішень.

Ключові слова: вІРТУАЛЬНА РЕАЛЬНІСТЬ, РОЗШИРЕНА РЕАЛЬНІСТЬ, OPENXR, ІНТЕГРАЦІЯ СИСТЕМ, ПРОГРАМНИЙ МОДУЛЬ, 3D SLICER, SLICERVIRTUALREALITY, VTK, API, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

ANNOTATION

Bahlei V. Software module for integration of the virtual reality elements display module with third-party systems. — Diploma thesis

The work is devoted to the current problem of developing software for integrating virtual reality (VR) and augmented reality (XR) element display modules with existing third-party systems. The growing importance of XR technologies in various industries, such as medicine, education and manufacturing, and the challenges associated with ensuring their compatibility and expanding functionality through integration are considered.

The work is written in 45 pages, contains 10 figures, 1 table, 2 appendices and 20 sources according to the list of references. The volume of graphic material is 2 sheets of A3 format.

The purpose of the work is to develop an approach and a software module that ensures the integration of the virtual reality element display system with third-party systems, in particular using the OpenXR standard, to increase the flexibility and functionality of XR applications. During the work, an analysis of modern approaches to the development of XR systems was conducted, the structure and principles of the functioning of the OpenXR standard were studied, including the session life cycle and extension mechanisms. An algorithm for determining the list of active API layers and extensions of OpenXR was developed.

The result of the work is the developed software module SlicerVirtualReality for integrating XR functionality into the 3D Slicer medical platform. The practical implementation of support for specific XR equipment (Magic Leap 2 controller) was demonstrated by creating and integrating the corresponding binding files for OpenXR, which ensures correct user interaction in the XR environment.

The practical significance lies in the creation of a software module that allows integrating modern XR capabilities into existing complex software products, such as 3D Slicer, expanding their functionality for medical visualization, planning and training. The proposed approach contributes to the standardization of the integration process and facilitates the development of cross-platform XR solutions.

Keywords: VIRTUAL REALITY, AUGMENTED REALITY, OPENXR, SYSTEMS INTEGRATION, SOFTWARE MODULE, 3D SLICER, SLICERVIRTUALREALITY, VTK, API, SOFTWARE DEVELOPMENT.

ЗМІСТ

ВСТУП

У цифровому середовищі використання технологій XR (розширена реальність), VR (віртуальна реальність) та AR (доповнена реальність) набуває все більшої актуальності завдяки їх здатності створювати інтерактивний, занурювальний досвід для користувачів у різних сферах — від освіти та медицини до виробництва, маркетингу й розваг. Ці технології дозволяють значно підвищити ефективність навчання, покращити сприйняття інформації та забезпечити інноваційні підходи до візуалізації складних процесів або об'єктів. Водночас важливою є інтеграція XR-, VR- та AR-рішень із третьосторонніми модулями, такими як аналітичні системи, хмарні сервіси чи платформи штучного інтелекту. Така інтеграція дозволяє розширити функціональність, забезпечити персоналізований підхід, автоматизувати процеси взаємодії та зробити взаємодію з цифровим контентом більш гнучкою та динамічною. Усе це сприяє підвищенню загальної ефективності цифрових рішень і відкриває нові можливості для бізнесу, науки та суспільства.

Крім того, поєднання XR-, VR- та AR-технологій з зовнішніми модулями забезпечує більш глибоку аналітику користувацької поведінки та дозволяє адаптувати середовище в реальному часі, що особливо важливо для навчальних та тренувальних платформ, де персоналізація має вирішальне значення. Наприклад, інтеграція з модулями машинного навчання дає змогу адаптувати контент відповідно до рівня підготовки користувача, а підключення до систем управління підприємствами (ERP, CRM) дозволяє створювати комплексні рішення для бізнесу з повною візуалізацією процесів і даних. Такі можливості відкривають шлях до створення більш ефективних, масштабованих і гнучких рішень, які відповідають потребам сучасного ринку.

Мета та завдання дослідження. Метою цієї дипломної роботи є розробка підходу до інтеграції системи з третьосторонніми модулями, зокрема в галузі OpenXR.

Для досягнення мети роботи були поставлені такі завдання:

- провести аналіз досліджень в галузі розробки XR, для виділення сучасного стану підходів до розробки XR систем;
- проаналізувати сучасні тенденції до розробки XR систем та проаналізувати структуру OpenVR/XR;
- проаналізувати цикл життя сесії OpenXR;
- розробити алгоритм визначення переліку активних слоїв API та розширенню;
- розробити прикладне програмне забезпечення 3D Slicer, зокрема, програмний модуль SlicerVirtualReality, що додає XR функціонал.

Об’єкт дослідження – інтерфейс взаємодії системи з третьосторонніми модулями.

Предмет дослідження – підходи до розробки додатків для XR систем.

Практичне значення полягає у розробці програмного модулю для інтеграції третьосторонніх модулів в сфері віртуальної реальності.

Для розв’язання поставлених задач у кваліфікаційній роботі використано сучасні підходи до розробки тертьосторонніх додатків для XR систем.

У першому розділі було проаналізовано сучасні підходи до розробки систем в галузі XR, виділено тенденції розвитку XR технологій, що дозволило розробити концепцію інтеграції з третьосторонніми модулями.

У другому розділі було розроблено алгоритми роботи з API та виділено елементи специфікації стандарту OpenXR. Виділено базові концепти системи. Розроблено алгоритм роботи розширення.

У третьому розділі розроблено прикладне програмне забезпечення 3D Slicer, зокрема, програмний модуль SlicerVirtualReality, що додає XR функціонал.

1. АНАЛІЗ XR-ТЕХНОЛОГІЙ

1.1 Аналіз досліджень та розробок в галузі XR

1.1.1 Аналіз публікацій

Доцільність розробок і досліджень в області XR технології визначається поширеністю прикладного застосування XR технологій, що в свою чергу оцінюється дослідженнями, нижче наведені деякі з них. Огляд [1] узагальнює систематичний аналіз застосування віртуальної реальності (VR) в освіті, охоплюючи мотивації, проблеми та потенційні рішення. Дослідження аналізує наукові публікації, зосереджуючись на темах, пов'язаних з VR в освіті. Результати показують, що VR часто використовується для моделювання та навчання, особливо в медичній сфері. Автори підкреслюють обмежене обґрунтування з точки зору педагогіки та зосередженість на внутрішніх факторах, таких як зацікавленість студентів. Проблеми включають вартість, навчання, зручність використання програмного забезпечення та необхідність більшого реалізму. У документі розглядаються нові технології VR, такі як гарнітури, і їхній потенціал для вирішення проблем. Нарешті, робляться висновки про важливість інтеграції педагогіки, подолання проблем зручності використання та вивчення альтернативних підходів до реалізму у VR-освіті.

Дослідження [2] вивчало використання середовищ розширеної реальності (XR) для покращення знань про морську екологію. Дослідження базувалося на мобільній XR-інтервенції "принеси свій пристрій" (BYOD) в освітньому морському центрі Нової Зеландії. З'ясувалося, що XR може допомогти краще зрозуміти складні наукові концепції охорони морського середовища, сприяючи знанням, зміні ставлення та поведінці, пов'язаній з екологічною грамотністю.

У статті [3] проаналізовано останні розробки в галузі технологій доповненої реальності (AR) і віртуальної реальності (VR) в освіті за останні дванадцять років (2011-2022). Шляхом аналізу великої кількості наукових статей з бази даних Scopus, використовуючи методи інтелектуального аналізу тексту,

дослідження виявило ключові тенденції, прогалини, переваги та проблеми, пов'язані із застосуванням AR та VR в освітньому процесі. Результати показують значне зростання використання AR та VR в освіті в останні роки, особливо з використанням носимих пристроїв. Дослідження підтвердило збільшення уваги до AR і VR після спалаху Covid-19, але поточні дослідження не достатньо висвітлюють це питання.

Дослідження [4] розглядає "фізитальний робочий простір" - поєднання фізичних та цифрових елементів - як спосіб зменшення вуглецевого сліду від бізнес-поїздок і щоденних поїздок на роботу, особливо з урахуванням досвіду пандемії COVID-19. Автори проводять систематичний огляд літератури про впровадження технологій розширеної реальності (XR) в різних бізнес-середовищах, визначаючи сильні та слабкі сторони цього підходу. Мета - створити основу для нової типології робочого середовища, яка використовує XR для ефективного зменшення вуглецевого сліду, враховуючи технологічні, соціальні та екологічні аспекти. В рамках дослідження пропонується методологія оцінки екологічного впливу phygital-просторів, а також імплементуються нові технологічні рішення.

У статті [5] зазначається, що VR застосовується для лікування психічних розладів, зокрема фобій, шляхом моделювання безпечних ситуацій та сприяння подоланню пацієнтами важких для уяви ситуацій. VR-терапія також корисна при депресивних розладах. У сфері фізичної реабілітації VR використовується для відновлення рухових функцій після інсульту, пропонуючи різні завдання та зображення, що мотивують пацієнтів до регулярних вправ. Однак, попри корисність VR, наявні обмеження, такі як вартість, а також психологічні та ергономічні проблеми.

Стаття [7] розглядає застосування технологій розширеної реальності (XR), зокрема віртуальної реальності (VR), доповненої реальності (AR) та змішаної реальності (MR), в медицині, зокрема в області терапії захворювань хребта. Розглянуто внесок цифрової трансформації, пандемії COVID-19 та малоінвазивної хірургії хребта (MISS) у популяризацію XR. Розглянуто

можливості використання 3D-медичних зображень та голограм на різних етапах: VR забезпечує візуалізацію та моделювання хірургічних втручань, AR використовується для навчання та консультування, візуалізуючи анатомію та накладаючи цифрові образи на реальне операційне поле з застосуванням HMD (Microsoft HoloLens і Magic Leap).

1.1.2 Аналіз сучасного стану розробок в сфері XR.

Робота по розробці XR технологій ведеться у вигляді покращень наявних та створення нових проприєтарних технологій в середині компаній-виробників, або, якщо йде мова про відкриті стандарти, контрибуцій до відкритих стандартів. Публікаціями, що супроводжують таку активність, переважно, є супутня документація [8], або матеріали, що раціоналізують рішення щодо конструкції технічних засобів [10].

Також є публікації, що розглядають практичні аспекти застосування тих чи інших імплементацій, наприклад, дослідження [9] оцінює можливості відстеження очей гарнітурою віртуальної реальності Meta Quest Pro, використовуючи дані про рухи очей, зібрані у 78 учасників. Окрім класичних показників якості сигналу (просторова точність, плавність) в ідеальних умовах, досліджено вплив освітленості та ковзання гарнітури на продуктивність пристрою. Результати цього дослідження можуть застосовуватись розробниками для адаптації ПЗ або його елементів, наприклад, інтерфейсу, до більшої кількості можливих сценаріїв використання.

1.2 Аналіз тенденцій розвитку XR технологій

Об’ємне стереоскопічне зображення як ніша дослідження та розробки існує значний проміжок часу. Стереоскопія - це метод створення або посилення ілюзії глибини зображення шляхом представлення глядачеві двох дещо різних зображень. Ці зображення знімаються (або візуалізуються) з дещо різних точок зору, імітуючи природний паралакс людського зору. Коли мозок поєднує ці два зображення, він інтерпретує відмінності як глибину, створюючи 3D-ефект. Дисплей, що розташований перед очима користувача, та використовує стереоскопію, надаючи окремий дисплей для кожного ока, представляючи для нього злегка зміщені зображення називають Head Mounted Display (HMD).

Перші HMD розробки були проведені компанією Philco в 1961 році. Їх система використовувала дисплей, встановлений на голові, для моніторингу середовища в іншій кімнаті, використовуючи магнітне відстеження для моніторингу рухів голови користувача. Система Philco HMD відображала реальне відео з віддалено встановленої камери. Положення камери змінювалося відповідно до відстежуваних рухів голови, створюючи відчуття телеприсутності.

У 1980-х роках зростали зусилля з комерціалізації технології віртуальної реальності, що призвело до кількох помітних розробок. Починаючи з 1980-х років, NASA активно досліджувало VR, зосереджуючись на застосуванні для підготовки астронавтів і дистанційного керування роботизованими системами. Система VIEW, розроблена в Дослідницькому центрі NASA в Еймсі, досліджувала використання HMD для телеприсутності та взаємодії у віртуальному середовищі. Хоча це був насамперед дослідницький проект, він вплинув на подальший розвиток віртуальної реальності. VPL Research Джарона Ланьє була ключовим гравцем у ранній комерційній розробці віртуальної реальності. EyePhone (кінець 1980-х) був одним з перших комерційно доступних VR HMD, що пропонував стереоскопічні дисплеї, рудиментарне відстеження голови та введення в рукавичках через DataGlove. Однак EyePhone і пов'язані з ним продукти VPL були дорогими і пропонували обмежену продуктивність

порівняно з сучасними стандартами. Ці ранні спроби комерціалізації, хоч і були важливими, але обмежувалися наявними технологіями, що призводило до громіздкості та низької роздільної здатності зображення, тощо. Висока вартість також робила їх недоступними для масового ринку.

Незважаючи на початковий ажіотаж, в середині 1990-х ринок VR пережив період стагнації. Цьому занепаду сприяло кілька факторів:

- Висока вартість: Системи віртуальної реальності залишалися дорогими, що перешкоджало їхньому впровадженню споживачами і навіть багатьма комерційними організаціями.
- Обмежений контент: Брак цікавого і добре розробленого контенту для віртуальної реальності ще більше заважав широкому прийняттю.
- Технологічні обмеження: HMD все ще були громіздкими, важкими і пропонували обмежену роздільну здатність дисплея та поле зору.

Nintendo, випустивши Nintendo Virtual Boy у 1995 році, продемонструвала обмеження доступної технології. Червоний монохромний дисплей, відсутність справжнього відстеження руху голови та ергономічні проблеми призвели до її комерційного провалу.

У 21-му столітті відбулося поступове відродження інтересу до віртуальної реальності, зумовлене такими факторами, як покращення обчислювальних потужностей та удосконалення технологій виготовлення дисплеїв. Так, у 2012 Палмер Лакі очолив Kickstarer проект Oculus Rift, широко відомий тим, що відродив інтерес до віртуальної реальності. Rift продемонстрував потенціал сучасної технології віртуальної реальності для створення захоплюючих і доступних вражень від занурення, що викликало хвилю нових інвестицій і розробок.

Успіх Oculus Rift проклав шлях для виходу на ринок інших великих гравців, зокрема Sony (PlayStation VR), HTC (Vive) та Valve (Index). Крім того, зростаючі можливості мобільних пристроїв призвели до розробки мобільних VR і AR-рішень, розширюючи сферу застосування технології XR.

Отож, терміни Virtual Reality (VR), Mixed Reality (MR), Augmented Reality (AR), в рамках цієї роботи, означають групу технологій взаємодії із цифровим контентом. Переважно, апаратна частина цих технологій складається з HMD, при цьому точка зору на об’єкт, що відображається на дисплеї відповідає положенню пристрою у реальному просторі. Вище згадані технології відрізняються такою специфікою:

VR — контент стереоскопічно відображається, повністю перекриваючи оточення користувача. Прикладами таких пристроїв є Valve Index, HTC Vive.

MR — ідентично з попереднім, але із опцією застосування вбудованих у пристрій відеокамер та датчиків для трансляції зображення зовнішнього світу як частини контенту. Окрім забезпечення візуального орієнтування у просторі.

AR — контент відображається поверх оптично прозорого матеріалу, крізь який, в свою чергу, користувач бачить оточення.

Термін Extended Reality (XR), що буде часто використовуватись в рамках цієї роботи є узагальнюючим терміном, що поєднує в собі терміни Virtual Reality (VR), Mixed Reality (MR) та Augmented Reality (AR), та усе між ними.

1.3 Аналіз суспільної значимості дослідження та огляд існуючих технологій

Прогрес у сфері XR технологій можна описати як паралельний розвиток у трьох залежних сферах. Кінцеві користувачі, що, в особі приватних індивідів, освітніх установ, працівників різних галузей приймають та інтегрують ці технології для прикладного застосування.

Особливої уваги заслуговує сфера медицини, де технології XR знаходять застосування у цілому переліку сфер медичної практики, посилаючись на Розділ 1.1.1, сюди входять:

1. Медична освіта:

3D-голограми для анатомії: XR дозволяє студентам-медикам та ординаторам візуалізувати анатомічні структури за допомогою 3D-голограм. Це покращує розуміння анатомії та просторових взаємовідносин краще, ніж традиційні методи навчання.

Дистанційна освіта: Дисплеї XR HMD (Head Mounted Displays) можуть полегшити дистанційне навчання, дозволяючи студентам-медикам дистанційно брати участь в обходах палат і забезпечуючи хірургічне керівництво, яке приносить користь стажерам і досвідченим хірургам, які навчаються новим навичкам.

2. Медичне обстеження:

Телемедицина: XR інтегрує носимі датчики і високопродуктивні відеокамери для віртуальних консультацій і віддаленого моніторингу пацієнтів, особливо коли йдеться про проблеми з хребтом. Це допомагає оцінити ефективність реабілітації за допомогою точних показників.

3. Хірургія

Хірургічне моделювання та планування: XR, зокрема VR, забезпечує реалістичні хірургічні симуляції з використанням фантомів, що полегшує практику та вдосконалення навичок у безризиковому середовищі, покращуючи підготовку до реальних операцій. Висока точність завдяки використанню інтегрованих технологій, таких як 3D-камера.

Хірургічна навігація: Рентгенівські системи, які можуть проектуватися на HMD хірурга і накладати зображення на тіло пацієнта, забезпечують навігацію в хірургії хребта в реальному часі. Зменшує радіаційне опромінення.

4. Реабілітація:

Зняття болю (аналгезія): VR виявилася ефективною як техніка відволікання уваги для полегшення болю, відома як VR-аналгезія. Відновлення рухових функцій: VR та відеоігри використовуються для покращення реабілітації, особливо для верхніх кінцівок. Реабілітація вдома: Дозволяє пацієнтам, які проходять реабілітацію, проходити стандартну реабілітацію вдома, замість того, щоб залишати лікарню. Також окремо варто виділити

випадки, коли XR пристрої розглядаються як пристрої двостороннього зв’язку, а не як засоби одноосібної взаємодії з контентом, себто застосовуються вбудовані в пристрій або периферійно підключені засоби для просторового відеозв’язку та аудіо зв’язку.

В основному такі сценарії використання — дистанційне проведення індивідуальних тренінгів для працівників на підприємствах, дистанційна технічна підтримка, а також телемедицина.

На практиці має різні виявлення, наприклад:

- зчитування QR кодів для отримання інформації про розташування суб’єктів технічної підтримки[12]
- Створення 3D голограми співрозмовника для реалізації просторових відеоконференцій[13]

Виробники обладнання, що постійно прагнуть до покращення ключових характеристик XR пристроїв та розширення їх функціональності, орієнтуючись на потреби різних груп користувачів – від індивідуальних споживачів до великих корпорацій. Цей прогрес можна простежити за кількома напрямками:

- Збільшення обчислювальної потужності: Постійне вдосконалення процесорів та графічних прискорювачів дозволяє створювати більш реалістичні та складні XR середовища, підтримуючи вимогливі до ресурсів додатки для професійного використання (медицина, інженерія, дизайн)[8][9][10][14].
- Покращення якості дисплеїв: Виробники інвестують у розробку дисплеїв з вищою роздільною здатністю, частотою оновлення та ширшим полем зору, мінімізуючи ефект “екранних дверей" та покращуючи занурення користувача.
- Зменшення ваги та габаритів: Комфорт користувача є пріоритетом, тому виробники працюють над створенням легших та компактніших XR пристроїв, що зручніше використовувати протягом тривалого часу. Розвиток технологій акумуляторів також сприяє збільшенню часу автономної роботи.
- Вдосконалення систем відстеження: Точніше та стабільніше відстеження рухів користувача та навколишнього середовища є ключовим для

створення переконливого XR досвіду. Впроваджуються нові методи відстеження, включаючи inside-out tracking та використання штучного інтелекту.

Сторонні розробники програмного забезпечення, які беруть на себе ініціативу в розробці програмних XR та спеціальних частин програмного забезпечення XR.

Розробники досягають цього за рахунок:

1) Використання апаратного забезпечення, що надається постачальниками апаратного забезпечення.

2) Задоволення потреб кінцевих користувачів, як зазначено вище.

Таким чином, сферу XR технологій слід розглядати як повноцінну екосистему, де кількість кінцевих користувачів, програмного забезпечення та наявних апаратних платформ постійно росте.

Проведемо огляд стеку, що використовується для приведення в дію XR програми.

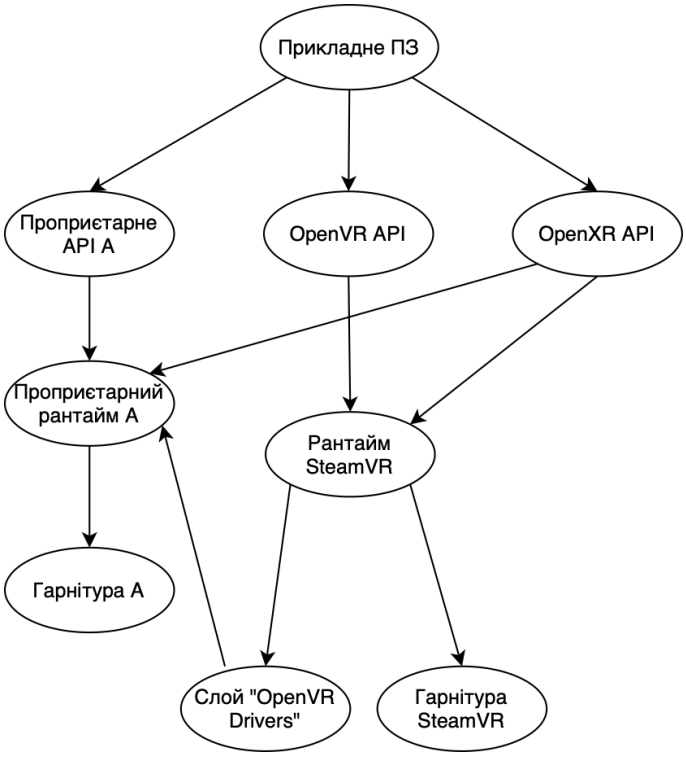

Рисунок 1.1 – Процес приведення в дію XR програми

Найвище, очевидно, прикладне ПЗ або комп’ютерна.гра. Найчастіше розробники ігор використовують готовий ігровий рушій для побудови проекту, хоча для побудови неігрових XR проектів ігрові рушії — також часте рішення.

Ігровий рушій - це компонент, який має на меті вирішити загальні задачі, що постають перед більшістю розробниками ігор - Unreal та Unity є найпоширенішими для XR;

Такі задачі включають:

- Реагування на введення користувача.
- Обробка ігрової фізики.
- Побудова кадру та застосування графічних АПІ (openGL, Vulkan)
- Взаємодія з API самого XR пристрою.

Інтерфейс прикладного програмування (API) - це мова, якою різні компоненти можуть спілкуватися між собою; якщо вони розмовляють однією мовою, вони можуть працювати разом.

Комплект для розробки програмного забезпечення (SDK) - ширший термін: іноді це просто інший спосіб сказати те ж саме, або це може бути набір декількох API та інших компонентів, корисних для розробників.

У 2013 році, коли був випущений оригінальний набір для розробників Oculus Rift (DK1), ігри використовували API Oculus для безпосереднього зв'язку з Oculus Runtime; це працювало лише з гарнітурами Oculus. Протягом 2010-х років інші виробники - Microsoft, Pimax, Varjo - представили свої власні подібні API, а ігрові рушії реалізовували пряму підтримку.

У 2015 році Valve Corporation випустили OpenVR, і хоча в подальшому під банером OpenVR, в якості окремого слоя API були додані апаратні абстракції, що дозволяють ПЗ OpenVR працювати з обладнанням від сторонніх виробників, «Драйвери OpenVR», сам OpenVR призначався як пряма заміна Oculus API для роботи з середовищем виконання SteamVR.

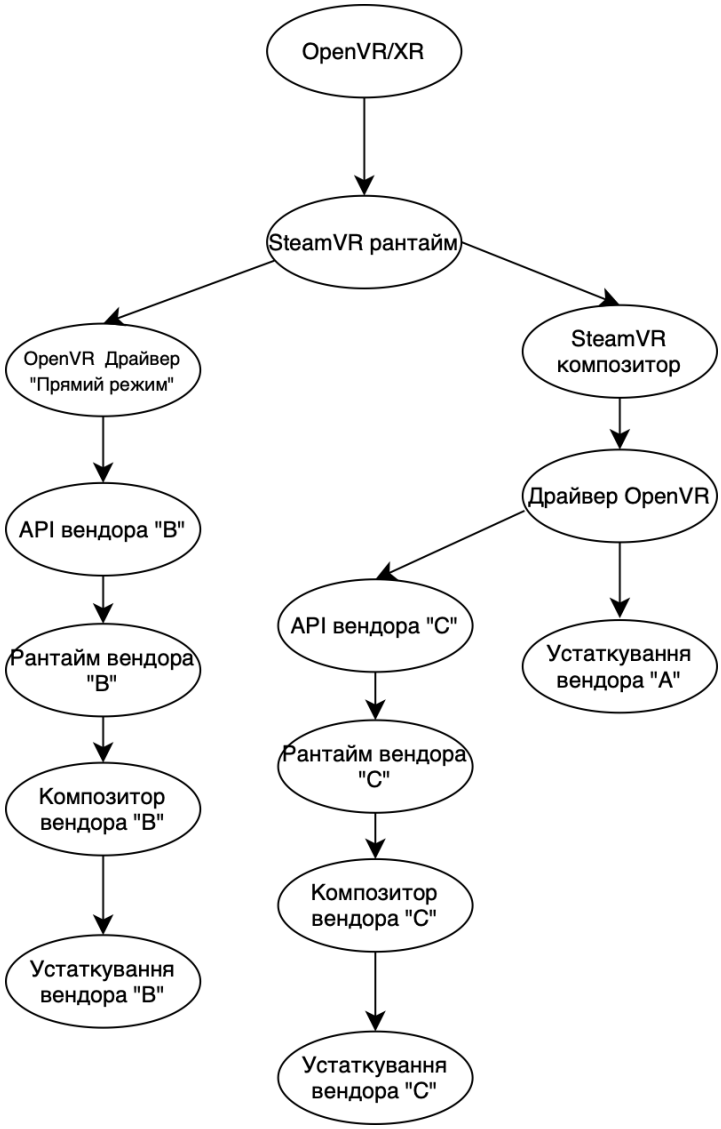

Рисунок 2 – Структура OpenVR/XR

Steam VR це середовище виконання або рантайм. У контексті XR застосунків, рантайм це фоновий сервіс, який обробляє введення з контролерів, зміну позиції у просторі, і передає ці дані прикладному ПЗ через API. При цьому для цілей сумісності, рантаймів у стеці може бути декілька, тоді вони

комунікують між собою через API та шари трансляції API. За таким сценарієм користувачу будуть доступні функції SteamVR, але компонувати та відсилати зображення до XR устаткування буде інший композитор[11]. Пояснення на схемі 2.

Враховуючи, що росте кількість апаратних платформ, котрі, в свою чергу, можуть кардинально відрізнятися між собою способами імплементації функціоналу, виникає загроза значного ускладнення процесу розробки XR орієнтованого програмного забезпечення через необхідність додавання пропрієтарного програмного коду для підтримки кожного із наявних пристроїв. Відтак, виникає потреба у стандартизованому проміжному програмному забезпеченні для побудови кросплатформенних рішень, оптимізованих для охоплення більшої кількості клієнтів, водночас користуючись перевагами інноваційних функцій конкретних платформ за допомогою розширень.

Розробники ПЗ могли - і часто вибирали - використовувати OpenVR, оскільки комбінація OpenVR + SteamVR дозволяє відносно просто забезпечити базову підтримку для гарнітур більшості виробників без необхідності писати окремий код для кожного виробника, проте стандарт OpenVR не заповнив в повній мірі нішу такого універсального API, зокрема через такі недоліки:

- Упередженість стосовно виробників. OpenVR фактично підтримує не тільки ті пристрої, для яких першочергово створювався. Інструменти та документація для побудови сторонніх драйверів доступні[10], але інструменти та документація для запровадження підтримки сторонніх пристроїв зі сторони SteamVR — відсутня.
- Відсутність стандартизації. Немає формальних специфікацій, немає органу зі стандартизації, немає колаборативного процесу, як у OpenGL або Vulkan. Це ускладнює довгострокові інвестиції з боку інших компаній.
- Прив'язаність до Steam. OpenVR за своїм задумом передбачав використання SteamVR. Реалізація роботи OpenVR із сторонніми рантаймами передбачає використання обхідних шляхів, що є небажаним.

Відтак, можна впевнено зробити висновок що, попри наявність OpenVR, незважаючи на окремі зусилля окремих розробників, залишалась незаповнена ніша для універсального стандарту доступу до XR пристроїв. Цю нішу заповнив стандарт OpenXR, далі “стандарт”.

2. АЛГОРИТМИ ІНТЕГРАЦІЇ МОДУЛІВ

2.1 Алгоритми роботи з API

З точки зору розробника прикладного ПЗ, OpenXR - це набір функцій, які взаємодіють з рантаймом для виконання загальновживаних операцій, таких як доступ до стану контролера/периферії, отримання поточних та/або прогнозованих позицій відстеження та надсилання відрендерених кадрів.

Для реалізатора рантайму, OpenXR - це набір функцій, які керують роботою XR-системи та встановлюють життєвий цикл XR-додатку. Завданням реалізатора є надання програмної бібліотеки на хості, яка реалізує OpenXR API, при цьому зіставляючи роботу кожної функції OpenXR з можливостями пристрою, щодо специфіки самої реалізації стандарт не надає жодних вказівок чи обмежень.

Елементи специфікації стандарту OpenXR наведено у таблиці 2.1.

Таблиця 2.1 - Елементи специфікації стандарту OpenXR

Концепція	Опис
API (Інтерфейс)	OpenXR API - це набір команд, функцій і структур, які повинні пропонувати OpenXR-сумісні Рантайми (середовища виконання).
Рантайм	Середовище виконання (Рантайм) - це специфічна реалізація функціоналу OpenXR. Вона може бути надана виробником обладнання або як частина операційної системи пристрою; вона може бути надана виробником програмного забезпечення, щоб забезпечити підтримку OpenXR на певному обладнанні. Завантажувач знаходить відповідне середовище виконання і завантажує його під час ініціалізації OpenXR.

Loader (Завантажувач)	Завантажувач OpenXR — це спеціальна бібліотека, яка підключає вашу програму до будь-якого середовища виконання OpenXR, яке ви використовуєте. Завдання завантажувача полягає в тому, щоб знайти середовище виконання та ініціалізувати його, а потім дозволити вашій програмі доступ до версії API середовища виконання. Деякі пристрої можуть мати кілька доступних середовищ виконання, але лише одне може бути активним у будь-який момент часу.
Слої АПІ	Шари (слої) API – це додаткові компоненти, що доповнюють систему OpenXR. Шар може допомогти з налагодженням або фільтрувати інформацію між програмою та середовищем виконання. Шари API вибірково вмикаються під час створення екземпляра OpenXR.
Інстанс (Екземпляр OpenXR)	Інстанс є базовим об'єктом, який дозволяє вашій програмі взаємодіяти з рантаймом. Прикладне ПЗ просить OpenXR створити екземпляр під час ініціалізації підтримки XR у вашій програмі.
Графіка	OpenXR має з'єднатися з графічним API для відображення вигляду з гарнітури. Які саме графічні API підтримуються, залежить від рантайму та апаратного забезпечення.
Введення/виведення	OpenXR дозволяє програмам запитувати, які входи та виходи доступні. Потім їх можна прив'язати до дій, щоб програма знала, що робить користувач.
Дія (Action)	Семантично визначене введення або виведення для програми, яке можна прив'язати до різних апаратних введень та виведень за допомогою прив’язок (байдінгів)
Прив’язка (байндінг)	Встановлення відповідності (Маппінг) введень/виведень рантайму чи устаткування до семантичних дій
Поза	Позиція та орієнтація у 3D просторі.

На практиці, прикладне ПЗ, що використовує функціонал OpenXR та, відповідно, OpenXR функції, мусить підвантажити лише бібліотеку openxr_loader, одна з основних задач якої, при створенні інстансу, підвантажити динамічну бібліотеку з тією імплементацією OpenXR функцій, що надається активним рантаймом. Визначення шляху до цього , у випадку ОС Windows, DLL файлу, здійснюється через реєстр Winodws.

Отож, для використання OpenXR, прикладне ПЗ, в першу чергу, має створити OpenXR інстанс.

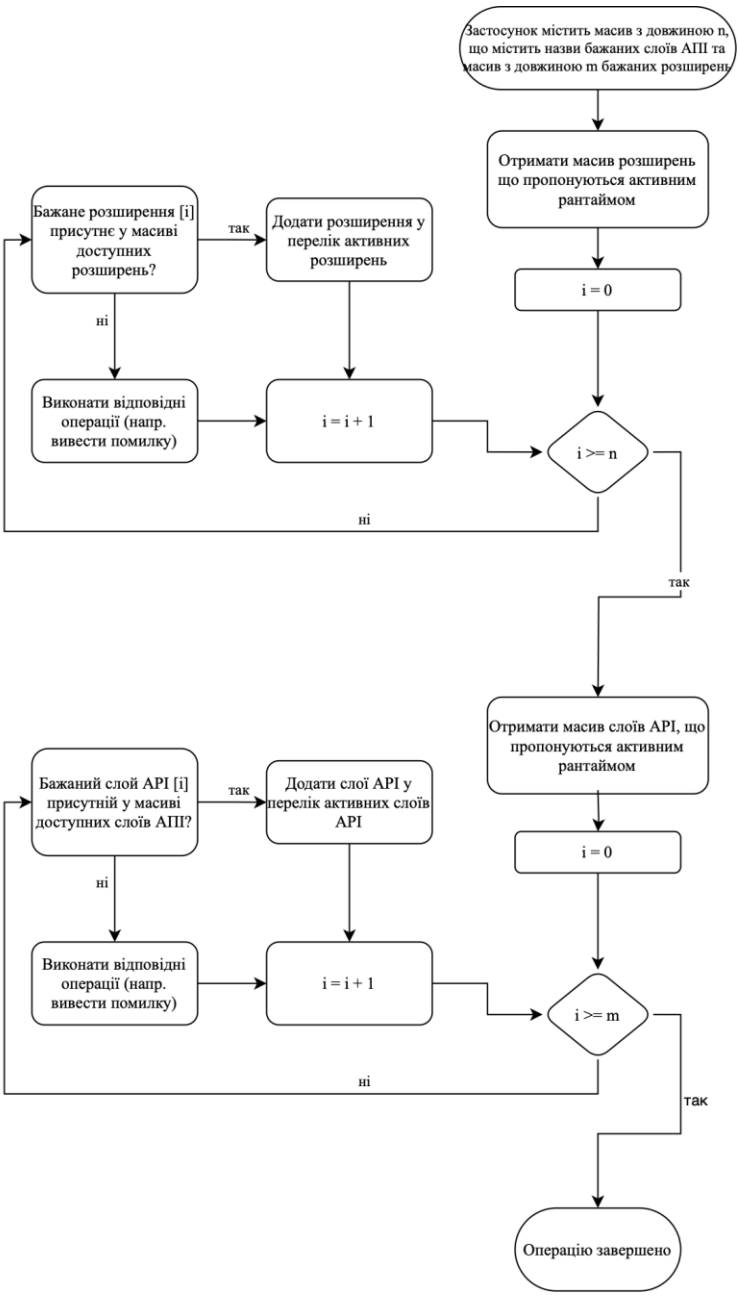

Рисунок 2.1 - Процес визначення переліку активних слоїв АПІ та розширенню

Інстанс OpenXR - це об'єкт, який дозволяє програмі OpenXR взаємодіяти з середовищем виконання OpenXR. Програма забезпечує цей зв'язок, викликаючи

xrCreateInstance і отримуючи дескриптор об'єкта XrInstance, що утворився у результаті.

Об'єкт XrInstance зберігає і відстежує стан програми, що стосується функцій OpenXR, без збереження такого стану у глобальному адресному просторі програми. Це дозволяє програмі створювати декілька екземплярів, а також безпечно інкапсулювати стан OpenXR, оскільки цей об'єкт є непрозорим для програми.

Фізично цей стан може зберігатися у будь-якому з шарів завантажувача OpenXR, OpenXR API або компонентів часу виконання OpenXR. Точне зберігання та розподіл цього збереженого стану залежить від реалізації.

Також на стадії створення інстансу, методу xrCreateInstance потрібно передати інформацію щодо версії та назви ПЗ, що створює цей інстанс, а також список бажаних розширень (extensions) та додаткових слоїв API, попередньо запитавши у активного рантайму, які саме додаткові слої API та розширення він підтримує, а які ні. Детальніше цей процес пояснюється на рисунку 2.1

Отримавши інстанс, наступним кроком є отримання XrSystemId. У специфікації OpenXR, розділ "System", наголошує на розмежуванні концепцій фізичної системи XR-пристроїв та логічних об'єктів, з якими безпосередньо взаємодіють програми. Система (System) представляє собою набір пов'язаних пристроїв у рантаймі, часто складається з декількох окремих апаратних компонентів, що працюють разом для забезпечення XR-досвіду.

XrSystemId може представляти як VR-гарнітуру, так і, наприклад, пару контролерів, або мобільний пристрій з video pass-through для AR. Отже, нам потрібно вирішити, який тип XrFormFactor ми хочемо використовувати, оскільки деякі рантайми підтримують декілька форм-факторів. Зазвичай, обирають XR_FORM_FACTOR_HEAD_MOUNTED_DISPLAY.

Після створення інстансу, наступним кроком є створення сесії (session). Сесія — це абстракція, яка дозволяє прикладному ПЗ взаємодіяти з графічним API через OpenXR, включаючи обробку рендерінгу, трекінгу та подій.

Сесія створюється через xrCreateSession, куди передаються параметри щодо графічного API, який буде використовуватися (наприклад, Vulkan чи DirectX).

Об’єкт XrSession зберігається прикладним ПЗ і використовується для подальших викликів API, пов’язаних з рендерінгом та керуванням станами.

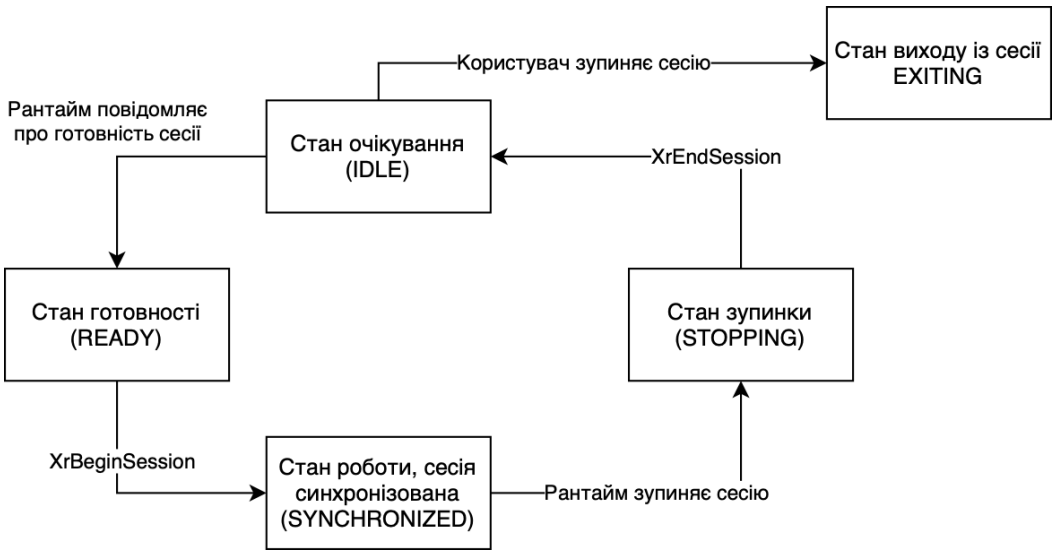

Рисунок 2.2 - Цикл життя сесії OpenXR

OpenXR використовує систему, засновану на подіях (event-based system) для повідомлення про зміни в XR-системі. Відповідальність за опитування цих подій та реагування на них лежить на застосунку. Опитування подій виконується за допомогою функції xrPollEvent. Застосунок повинен викликати цю функцію на кожному кадрі. Протягом одного XR-кадру застосунок повинен безперервно викликати xrPollEvent до тих пір, поки внутрішня черга подій не буде "спустошена"; в XR-кадрі може відбуватися декілька подій, і застосунок повинен обробити кожну з них.

OpenXR використовує концепцію swapchains. Swapchain - це серія зображень, які використовуються для відображення відрендереної графіки на дисплеї. У swapchain зазвичай знаходиться 2 або 3 зображення, що дозволяє платформі плавно їх відображати для користувача.

На відміну від звичайної розробки графіки, коли розробник сам створює swapchain, в OpenXR розробник звертається до OpenXR для створення swapchains, а XR compositor (який є частиною XR runtime) використовується для відображення відрендереної графіки. XR застосунки є унікальними тим, що часто мають декілька "views" (точок зору), до яких потрібно відрендерити графіку.

2.2 Базові концепти системи

OpenXR ставить перед собою ціль підтримувати необмежену кількість пристроїв, аби застосунки працювали (без модифікації) на апаратному/програмному забезпеченні, на якому це ПЗ не тестувалось, а також на апаратному забезпеченні, яке ще не було винайдено.

Для забезпечення роботи пристроїв введення/виведення із досягненням вищезгаданих цілей було запропоновано систему з такими основними концептами:

Дія(Action)

Натиски кнопок, рухи джойстика, гаптичні вібрації на контролері і т.д у OpenXR виражаються як “Дії” — суто семантичні об’єкти, що не обов'язково пов'язані до конкретного елементу керування: наприклад, розробник ПЗ визначає дію “вибрати”, а не дію “натиснути кнопку А”. визначає дію “ходити”, а не “положення лівого аналогового джойстика”.

Набір дій (Action Set)

У розробника ПЗ є можливість задавати контексти, у яких активні той чи інший перелік. Наприклад, у застосунку кнопка “А” на контролері означає дію “Створити об’єкт”, коли користувач відкриває меню, кнопку “А” потрібно переназначити до дії “Вибрати пункт меню”. Відповідна логіка також імплементується зі сторони рантайму.

Профіль взаємодій (Interaction profile)

Профіль взаємодії - це набір взаємодій, які підтримуються певним пристроєм та середовищем виконання і визначаються за допомогою текстових шляхів.

Кожна взаємодія визначається шляхом, що складається з трьох компонентів: шлях профілю, шлях користувача та шлях компонента. Шлях профілю має вигляд "/профілі_взаємодії/<назва_виробника>/<назва_типу>". Шлях користувача - це рядок, що ідентифікує пристрій контролера, наприклад, "/user/hand/left" або "/user/gamepad". Шлях компонента - це рядок, що ідентифікує конкретний вхід або вихід, наприклад, "/input/select/click" або "/output/haptic_left_trigger".

Наприклад, профіль простого контролера Khronos має такий шлях:

"/interaction_profiles/khr/simple_controller"

Для користувача він має шлях “/user/hand/left" та "/user/hand/right".

Шляхи до компонентів такі:

"/input/select/click"

"/input/menu/click"

"/input/grip/pose"

"/input/aim/pose"

"/output/haptic"

Поєднавши ці три частини разом, ми можемо визначити кнопку вибору на лівому контролері як “профіль + користувач + компонент” Відтак, маємо: "/interaction_profiles/khr/simple_controller/user/hand/left/input/select/click"

Прив’язка (Binding)

Binding це процес пов’язування шляхів компонента взаємодії (Interaction) до дії (Action), при тому що це пов’язування відбувається безпосередньо рантаймом, а розробник ПЗ може їх лише запропонувати, якщо бажає оптимізувати своє ПЗ під конкретний пристрій.

Концепт OpenXR Actions також дозволяє в реальному часі дізнаватися, чи можливо отримати данні про стан дії в принципі. На стадії опитування про дії, отримуючи структуру про стан конкретної дії, можна прочитати параметр isActive, котрий має значення true, якщо стан дії дійсно зчитується, якщо параметр false — опитування дії провалилося, наприклад, через раптове відключення контролера або втрату даних при зв’язку з контролером.

Аналогічно, структура із даними про стан дії із значенням із плаваючою комою має значення changedSinceLastSync, яке є істинним, якщо значення змінилося між попереднім і поточним викликами xrSyncActions, а також вона має lastChangeTime, який є часом останньої зміни значення. Це дозволяє нам дуже точно визначити, коли користувач натиснув кнопку і як довго він її утримував. Це може бути використано для виявлення «довгих натискань» або подвійних кліків.

Ретельне використання цих метаданих опитування допомагає створювати інтуєтивно зрозумілі додатки у використанні.

2.3 Алгоритм роботи розширення

OpenXR має обширну підтримку розширень (extensions). Це можуть бути функції для роботи з унікальними функціями конкретного пристрою або ПЗ. Фактично не має обмежень на цільові призначення розширень. Якщо рантайм має намір імплементувати якийсь особливий функціонал, може існувати розширення для його інтеграці із застосунками, навіть якщо відповідну прикладну задачу вже вирішено іншими, не пов’язаними з OpenXR засобами.

Розширення, що пропонуватиметься нижче відноситься саме до такого класу розширень.

Запропонований концепт розширення XR_EXT_audio уможливлює інтеграцію просторового аудіо у OpenXR, дозволяючи рантаймам за бажанням керувати аудіопотоками, просторово їх локалізувати та синхронізувати, відстежуючи просторові данні джерел звуків.

На додатку А окреслено деякі функції та структури, що додаються цим розширенням.

У цьому розширенні, як бачимо, аудіо передається у рантайм у формі окремого композиційного шару (Composition Layer).

Композиційні шари OpenXR — це спосіб, яким XR-застосунок передає відрендерені зображення (кадри) OpenXR рантайму. Рантайм, у свою чергу, використовує ці шари для компонування остаточного зображення, яке бачить користувач на своєму XR-пристрої. Кожен шар може містити різні типи контенту (наприклад, 3D-сцени, 2D-елементи інтерфейсу), і рантайм може застосовувати до них різні ефекти та трансформації перед відображенням. Використання шарів дає змогу гнучко комбінувати контент з різних джерел і досягати більш якісного та комфортного XR-досвіду для користувача. Основні типи шарів - це проєкційні шари для рендерингу в 3D та плоскі шари для 2D елементів інтерфейсу, але є й інші можливості розширення базового функціоналу, як от у розширені XR_EXT_audio.

Порядок використання цього розширення розробником прикладного ПЗ виглядатиме, в загальних рисах, так:

1. Створення джерел звуку (фонова музика, кроки, голос).
2. Прикріплення джерела до відстежуваних просторів (персонажів, об'єктів).

На кожному кадрі:

1. Оновлюється просторове положення слухача.
2. Надсилається композиційний шар з активними джерелами аудіо.
3. Рантайм накладає просторові ефекти та міксує аудіо.
4. Загалом, використання такого розширення може дати такі переваги:

- Уніфікована крос-пристрійна звукова абстракція.
- Спрощене налаштування просторового звуку.
- Зменшує кількість зовнішніх залежностей.
- Дозволяє реалізувати апаратне прискорення звуку.

Також, розширеннями можна підвантажити специфічні профілі взаємодій, унікальні для одного конкретного рантайму, наприклад, розширення XR_ML_ml2_controller_interaction визначає профіль взаємодії для контролера Magic Leap 2. Зокрема такі шляхи:

…/input/menu/click
…/input/home/click
…/input/trigger/click
…/input/trigger/value
…/input/trackpad/y
…/input/trackpad/x
…/input/trackpad/click
…/input/trackpad/force
…/input/trackpad/touch
…/input/grip/pose
…/input/aim/pose
…/input/shoulder/click
…/output/haptic

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЮ

3.1 Структура модулю SlicerVirtualReality

Об’єктом роботи у цьому розділі є прикладне програмне забезпечення 3D Slicer, зокрема, програмний модуль SlicerVirtualReality, що додає XR функціонал до цього ПЗ.

3D Slicer - це потужна платформа з відкритим вихідним кодом для медичної інформатики зображень, обробки зображень і тривимірної візуалізації. Її архітектура є модульною та багаторівневою, що відображає складні потреби досліджень та клінічного використання медичних зображень. В основі 3D Slicer лежить Visualization Toolkit (VTK), який надає основні структури даних і алгоритми для тривимірної графіки, обробки зображень і чисельних обчислень. Більшість логіки рендерингу, взаємодії та конвеєра даних у Slicer побудовано на класах VTK, таких як vtkImageData, vtkPolyData, а також на різноманітних класах фільтрів та мапперів. Ці елементи VTK абстраговано і розширено у Slicer для забезпечення специфічних можливостей - наприклад, відображення медичних об'ємів, накладання сегментації та 3D-реконструкцій.

3D Slicer це також платформа для розробки продуктів, яка дозволяє компаніям швидко створювати прототипи та випускати продукти для користувачів. Розробники можуть зосередитися на розробці нових методів і не витрачати час на переробку базових функцій імпорту/експорту даних, візуалізації, взаємодії. Додаток розроблений таким чином, щоб його можна було легко кастомізувати (з власним брендингом, спрощеним користувацьким інтерфейсом і т.д.). 3D Slicer є повністю безкоштовним і немає ніяких обмежень щодо його використання - тільки дистриб'ютор програмного забезпечення повинен гарантувати, що розроблений додаток підходить для використання за призначенням.

Центральною концепцією Slicer є MRML, мова розмітки медичної реальності. MRML - це модель даних і граф сцени, що використовується для представлення та управління вмістом сцени Slicer - по суті, всіма даними, з якими працює користувач (об'єми, моделі, трансформації, сегментації тощо) та їх взаємозв'язками. Вузли MRML (похідні від базового класу vtkMRMLNode) інкапсулюють кожну логічну одиницю даних, а сцена (vtkMRMLScene) керує всіма вузлами та їх серіалізацією. MRML забезпечує високорівневу абстракцію над структурами вихідних даних VTK, що дозволяє послідовно спостерігати, скасовувати/повторювати та здійснювати міжмодульний зв'язок. Вузли MRML зберігають посилання на базові об'єкти VTK, але самі не відповідають за обробку або візуалізацію.

Для рендерингу та відображення даних MRML у інтерфейсі програми Slicer використовує менеджери відображення, зокрема, MRML Displayable Managers (MRMLDM). Ці компоненти відстежують зміни у вузлах MRML та відповідно оновлюють візуальне представлення, діючи як міст між MRML та стеком рендерингу VTK/Qt. Наприклад, якщо вузол об'єму змінено або до сцени додано новий вузол, відповідний менеджер відображення створить або оновить відповідний vtkActor або маппер фрагментів зображення у вікні перегляду. Кожен тип вузла MRML має один або декілька менеджерів відображення, які визначають, як він відображається у двовимірних і тривимірних поданнях. Ця архітектура чітко відокремлює логіку даних від логіки рендерингу і дозволяє користувацькому інтерфейсу відображати поточний стан сцени без необхідності модулів безпосередньо модифікувати вигляд.

Іншими рисами, процес відображення моделі в загальних рисах виглядає так:

1. Користувач завантажує модель — створюється vtkMRMLModelNode
2. Цей вузол додається до vtkMRMLScene
3. Менеджер відображення vtkMRMLModelDisplayableManager виявляє новий вузол і створює для нього:
   A vtkActor

4. A vtkPolyDataMapper
5. Підключає “актора” до рендерера.
6. Коли користувач змінює, наприклад, розмір чи положення vtkMRMLModelNode, тоді менеджер відображення оновлює положення vtkActor

Модулі Slicer - це, по суті, плагіни, які розширюють функціональність програми. Ці модулі можуть бути реалізовані на C++, Python або у вигляді виконуваних файлів CLI. Нативні модулі зазвичай взаємодіють зі сценою MRML для читання і запису даних, виконання операцій (таких як фільтрація, реєстрація або сегментація зображень), а також надають користувацькі інтерфейси за допомогою віджетів Qt. Коли модуль завантажується, він реєструється у програмі та оголошує свої входи, виходи та будь-які залежності від вузлів MRML. Більшість модулів не мають стану у тому сенсі, що вони покладаються на сцену MRML для всіх постійних даних, що дозволяє користувачам об'єднувати модулі у ланцюжки, зберігати та перезавантажувати сеанси, а також працювати з великими конвеєрами. Внутрішньо модулі часто обгортають або інтегрують конвеєри VTK, але з додатковою логікою для сумісності з MRML та інтеграції з користувацьким інтерфейсом. Така модульна конструкція дозволяє Slicer добре масштабуватися і підтримувати як експериментальні функції дослідницького рівня, так і стабільні клінічні робочі процеси.

Простіше кажучи, модулі 3D Slicer можуть:

- Створювати нові MRML вузли
- Створювати нові менеджери відображення MRMLDM
- Створювати можливості обробки VTK
- Додавати інший сторонній функціона

3.2 Структура ієрархії модулів для для забезпечення взаємодії з користувачем

Розширення Slicer Virtual Reality також використовує готові модулі VTK для забезпечення XR функціоналу. VTK підтримує як OpenVR, так і OpenXR, досягається це за рахунок абстрактності VR/XR фреймворку, що описаний у модулі VTK::RenderingVR. Цей модуль визначає набір потрібних для XR функціоналу компонентів та інтерфейсів, не прив'язуючись до рантайму. Імплементації цих абстрактних класів та методів відбуваються в окремому модулі, що буде розглянуто згодом.

Зокрема розглянемо такі підмодулі модуля "VTK::RenderingVR":

- vtkVRRendererWindow: відповідає, в першу чергу, за перетворення між системою координат у якій знаходиться HMD пристрій (фізичною) у систему координат VTK сцени (система координат світу) із vtkRenderWindow.h:
  void SetPhysicalToWorldMatrix(vtkMatrix4x4* matrix);
  void GetPhysicalToWorldMatrix(vtkMatrix4x4* matrix) override;
  Цей клас вводить поняття дескрипторів пристроїв. DeviceHandle - це дескриптор типу uint32_t, який представляє пристрій у базовому VR API, такому як OpenVR або OpenXR. Реалізації цього класу відносно конкретного рантайму відповідають за представлення реальних пристроїв унікальними дескрипторами DeviceHandle.
- vtkVRRenderer адаптований під XR стандартний рендерер VTK. Керує окремим vtkOpenGLRenderer рендерером, використовуючи клас камери, що імплементується модулями vtkVRCamera та vtkVRHMDCamera. Модуль VTK::vtkRenderingVR також запроваджує фреймворк для обробкти взаємодій, зокрема vtkVRRenderWindowInteractor.
- vtkVRRenderWindowInteractor та vtkVRInteractorStyle: Для додаткової гнучкості, у VTK система взаємодії користувача зі сценою також побудована на ієрархії модулів. Спрощено, vtkRenderWindowInteractor3D відповідає за прослуховування необроблених подій (введення з клавіатури, кліки миші), а vtkInteractorStyle3D описує що пілся цих подій має відбутися. Абстрактні класи vtkVRRenderWindowInteractor та vtkVRInteractorStyle, відповідно,

додають обробку подій для XR. Так, vtkVRRenderWindowInteractor. вимагає від підкласів реалізації методу DoOneEvent() відносно циклу обробки подій конкретного рантайму, а також методу AddAction() для встановлення відповідності подям рантайму до функцій VTK.

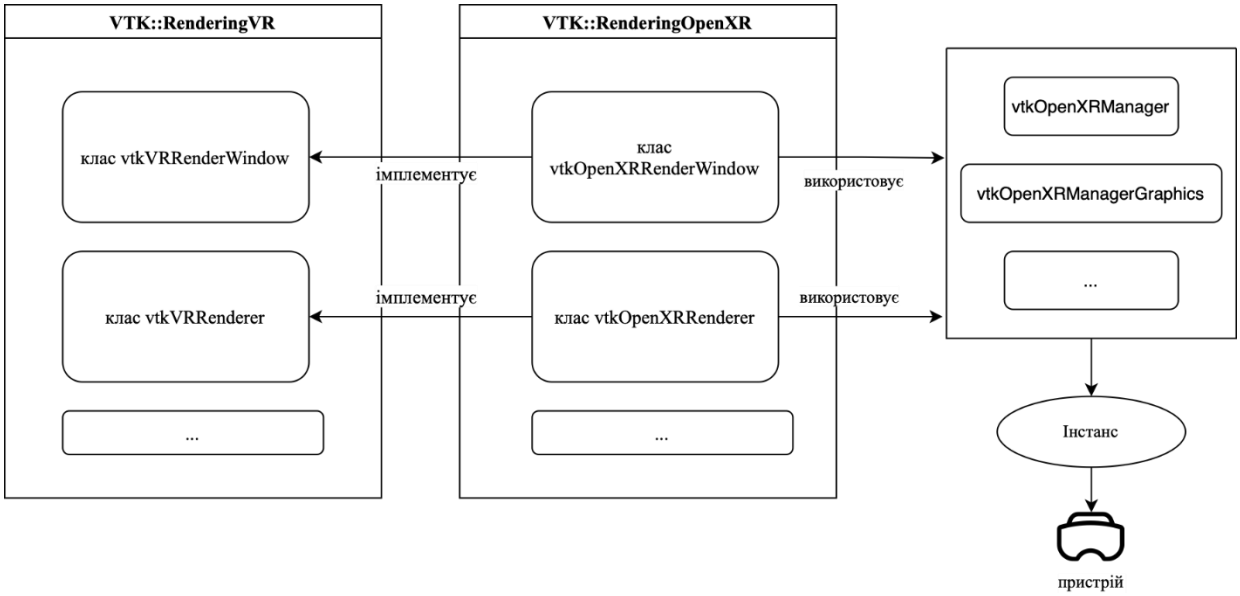

Рисунок 3.1 – Схема взаємодії модулів

Модуль VTK::RenderingOpenXR, в свою чергу, імплементує (реалізує) вище наведені абстрактні класи для роботи із OpenXR рантаймом зокрема. Також цей модуль має окремі, самостійні класи для безпосередньої взаємодії із інстансом OpenXR, . Наприклад, підмодуль vtkOpenXRRenderWindow — конкретна реалізація vtkVRRenderWindow. Окірм іншого, обробляє цикл рендерингу для OpenXR (очікування/початок/кінець кадру,

отримання/вивільнення зображення), для цього підмодуль використовує OpenXR функції у vtkOpenXRManager, включивши їх у своєму коді.

```
#include "vtkOpenXRManager.h"
//...
bool vtkOpenXRRenderWindow::GetSizeFromAPI()
{
  vtkOpenXRManager& xrManager = vtkOpenXRManager::GetInstance();

  std::tie(this->Size[0], this->Size[1]) = xrManager.GetRecommendedImageRectSize();

  return true;
}
```

Рисунок 3.2 – Приклад методу vtkOpenXRRenderWindow

На додатку Б, розглянемо детальніше частину ієрархії модулів для забезпечення взаємодії з користувачем.

vtkOpenXRRenderWindowInteractor імплементує цикл обробки подій в якості підкласу vtkVRRenderWindowInteractor, керує системою дій (Actions) OpenXR рантайму (через vtkOpenXRManager), передає об’єкту vtkOpenXRInteractorStyle методи для створення VTK дій, що відповідають переліку діям OpenXR.

Для розуміння, які OpenXR дії доступні у VTK (зміна позиції контролера, відкриття меню, дія при натисканні окремої кнопки взаємодії на контролері), і відповідно, які OpenXR дії потрібно оголошувати, існує маніфест дій OpenXR у форматі json: vtk_openxr_actions.json.

Файл vtk_openxr_actions.json також містить посилання на JSON файли із відповідністю взаємодій із профілів взаємодій конкретних OpenXR пристроїв до дій VTK, що описані у vtk_openxr_actions.json. Так звані binding файли. Як зазначено у розділі 2.2, це потрібно, аби вказати рантайму те, до яких фізичних елементів введення бажано прив’язати конкретні OpenXR дії, визначені прикладним ПЗ, тим самим забезпечуючи повноцінну підтримку рантайму.

3D Slicer та його розширення SlicerVirtualReality, як прикладне ПЗ, що використовує VTK, обгортає функціональні можливості VTK VR, вбудовує їх у свій фреймворк і налаштовує взаємодію відповідно до потреб візуалізації медичних зображень і взаємодії у віртуальній реальності. Зокрема інтеграція із MRML: vtkMRMLVirtualRealityViewNode - вузол перегляду віртуальної реальності для зберігання та керування конфігураціями VR. Це дозволяє зберігати і завантажувати VR-налаштування зі сценами Slicer. Він також оновлює вузли перетворення MRML з живими позиціями пристроїв віртуальної реальності, роблячи їх легко доступними у всьому додатку Slicer. Інтеграція з інтерфейсом користувача: Стандартний модуль Qt Slicer (qSlicerVirtualRealityModule) та віджет (qSlicerVirtualRealityModuleWidget) надають користувацькі елементи керування для VR, які взаємодіють з вузлом MRML. Slicer призначений для умовної компіляції та вибору під час виконання (через vtkMRMLVirtualRealityViewNode::XRBackend) між різними бекендами VTK VR (OpenVR, OpenXR).

3.3 Структура ієрархії модулів для для забезпечення взаємодії з користувачем

Переходимо до постановки практичної задачі та реалізації. Модуль VTK::RenderingOpenXR забезпечує пряму підтримку лише невеликої низки XR пристроїв, адже налічує обмежену кількість binding файлів (див Рисунок 3.3). Одним з таких пристроїв, що не має прямої підтримки, є AR гарнітура Magic Leap 2. Це означає, що не гарантується коректне зв’язування елементів керування із діями VTK, і, відповідно, функціями SlicerVirtualReality, а також неповноцінне використання можливостей апаратного забезпечення, такі як сенсорна панель на пульті керування MagicLeap2.

Відтак, основним етапом додавання цієї прямої підтримки буде створення та інтеграція відповідного файлу openxr_binding.

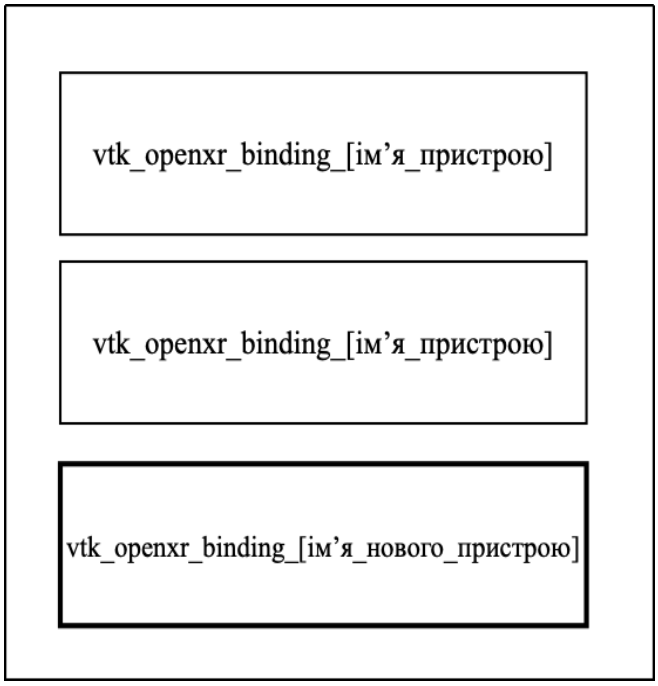

Рисунок 3.3 – Блоки vtk

Перший практичний крок для додавання підтримки для контролера ML2 — модифікувати вихідний код VTK та створити відповідний json файл та розмістити його у папці /Rendering/OpenXR/.

На рисунку 3.5 наведено розташування розробленого json файлу в списку.

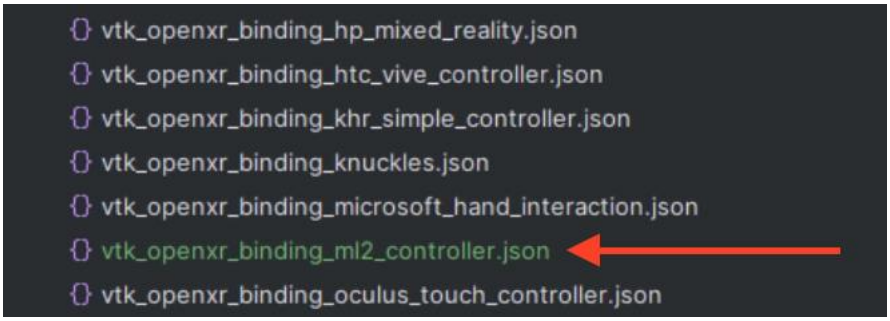

Рисунок 3.4 - Розташування розробленого json файлу

Наповнити цей файл потрібно по аналогії з іншими подібними, прив'язавши VTK дії до фізичних елементів введення, що представлені семантичними шляхами. Набір семантичних шляхів до елементів введення, що

доступні на пристрої уже згадувався у розділі 2.3. В результаті, маємо такий вміст файлу vtk_openxr_binding_ml2_controller.json:

- *"interaction_profile": "/interaction_profiles/magicleap/ml2_controller",*
- *"bindings": {*
- *"vtk-actions":{*
- *"sources": [*
- *{*
- *"inputs": {*
- *"click": {*
- *"output": "triggeraction"*
- *}*
- *},*
- *"path": "/user/hand/right/input/trigger"*
- *},*
- *{*
- *"inputs": {*
- *"touch": {*
- *"output": "startmovement"*
- *},*
- *"position": {*
- *"output": "movement"*
- *}*

Наступний етап — зазначити модифікувати файл vtk_openxr_actions.json та зазначити там про створений нами файл у масиві default_bindings.

Рисунок 3.5 – Вміст масиву default_bindings

Далі потрібно перевірити код підмодуля vtkOpenXRInteractionStyle, чи усі VTK дії, що описані вище, мають відношення до відповідної VTK команди, адже за замовчуванням не всі OpenXR дії, оголошені у vtk_openxr_actions.json мають відношення до відповідної VTK команди.

Далі потрібно скомпілювати, по черзі, VTK (із внесеними змінами до коду), 3D Slicer, SlicerVirtualReality. І одержати результат.

На зображені зліва (до розробки програмного модуля для інтеграції модуля відображення елементів віртуальної реальності із третьосторонніми системами) видно, як натискання на сенсорну панель контролера не приводить до руху у просторі сцени, в той час як на зображені зліва (після проведеної роботи), видно, як що при відповідному введені відповідна дія відбувається у прикладному ПЗ.

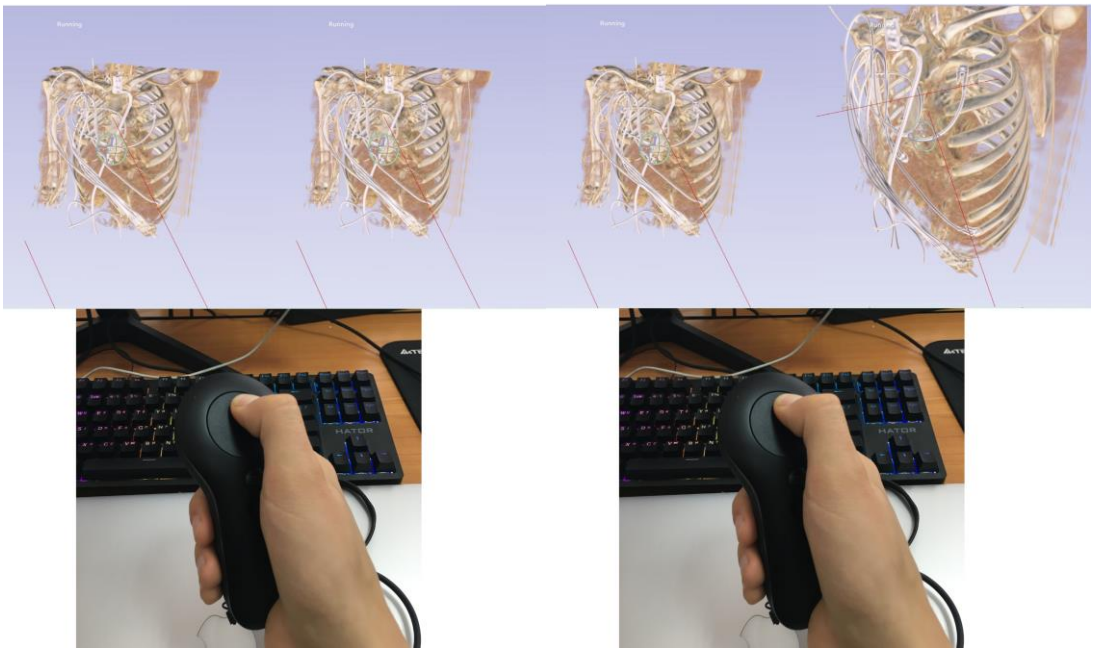

Рисунок 3.6 — Демонстрація роботи програмного модуля

ВИСНОВКИ

На основі аналітичного підходу проведено аналіз досліджень в галузі розробки XR, для виділення сучасного стану підходів до розробки XR систем, що дозволило виділити їх переваги та недоліки.

На основі аналітичного підходу, проаналізовано сучасні тенденції до розробки XR систем та проаналізувати структуру OpenVR/XR, що дозволило визначити концепції для розробки інтеграції з третьосторонніми модулями.

Проаналізовано цикл життя сесії OpenXR, що дозволило виділити основні компоненти системи для розробки модулю інтеграції з третьосторонніми модулями.

На основі алгоритмічного підходу розроблено алгоритм визначення переліку активних слоїв АПІ, що дозволило сформувати порядок завантаження об’єктів для розробки програмного інтеграційного модулю.

Додатково розроблено програмну реалізацію методу vtkOpenXRRenderWindow, який є частиною третьостороннього програмного модулю.

На основі підходів до розробки програмних модулів для XR додатків, розроблено структуру ієрархії модулів для для забезпечення взаємодії з користувачем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kavanagh, S., Luxton-Reilly, A., Wuensche, B., & Plimmer, B. (2017). A systematic review of Virtual Reality in education. Themes in Science and Technology Education, 10(2), 85-119.
2. Aguayo, C., & Eames, C. (2023). Using mixed reality (XR) immersive learning to enhance environmental education. The Journal of Environmental Education, 54(1), 58–71. https://doi.org/10.1080/00958964.2022.2152410
3. Abdullah M. Al-Ansi, Mohammed Jaboob, Askar Garad, Ahmed Al-Ansi, Analyzing augmented reality (AR) and virtual reality (VR) recent development in education, Social Sciences & Humanities Open, Volume 8, Issue 1, 2023, 100532, SSN 2590-2911,
4. Lo TTS, Chen Y, Lai TY and Goodman A (2024), Phygital workspace: a systematic review in developing a new typological work environment using XR technology to reduce the carbon footprint. Front. Built Environ. 10:1370423. doi: 10.3389/fbuil.2024.1370423
5. XR and Workers ’safety in High-Risk Industries: A comprehensive review. Joana Eva Dodoo a, Hosam Al-Samarraie b,* , Ahmed Ibrahim Alzahrani c, Tang Tang b
6. Mazurek, J., Kiper, P., Cieślik, B., Rutkowski, S., Mehlich, K., Turolla, A., Szczepańska-Gieracha, J. (2019). Virtual reality in medicine: a brief overview and future research directions. Human Movement, 20(3), 16-22. https://doi.org/10.5114/hm.2019.83529
7. Morimoto, T.; Kobayashi, T.; Hirata, H.; Otani, K.; Sugimoto, M.; Tsukamoto, M.; Yoshihara, T.; Ueno, M.; Mawatari, M. XR (Extended Reality: Virtual Reality, Augmented Reality, Mixed Reality) Technology in Spine Medicine: Status Quo and Quo Vadis. J. Clin. Med. 2022, 11, 470. https://doi.org/10.3390/jcm11020470
8. https://registry.khronos.org/OpenXR/specs/1.1/html/xrspec.html

9. Aziz, S., Lohr, D. J., Friedman, L., & Komogortsev, O. (2024, June). Evaluation of eye tracking signal quality for virtual reality applications: A case study in the meta quest pro. In Proceedings of the 2024 Symposium on Eye Tracking Research and Applications (pp. 1-8)
10. https://github.com/ValveSoftware/openvr/wiki/Driver-Documentation
11. https://github.com/ValveSoftware/openvr/wiki/IVRCompositor_Overview
12. https://www.magicleap.care/hc/en-us/articles/12130709541517-Assist-User-Guide
13. https://www.webex.com/us/en/solutions/hologram.html
14. http://media.steampowered.com/apps/valve/2016/Alex_Vlachos_Advanced_VR_Rendering_Performance_GDC2016.pdf