

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій
Кафедра прикладних інформаційних систем**

122 «Комп'ютерні науки»
(шифр і назва спеціальності)

«Прикладне програмування»
(назва освітньої програми)

Кваліфікаційна робота бакалавра

на тему: «Застосунок з використанням віртуальної реальності»

Виконав _____



(Підпис)

Філоненко Олексій Вячеславович

(прізвище, ім'я, по батькові)

Керівник Шолохов Олексій Вікторович

(прізвище, ім'я, по батькові)

(Резолюція «До захисту»)

Попередній захист:

(Висновок: "До захисту в екзаменаційній комісії")

Завідувач кафедри

Плескач В.Л.

(Підпис)

(Прізвище, ініціали)

(Дата)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____



(підпис)

Київ – 2021

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

Назва теми: «Застосунок з використанням віртуальної реальності»

Освітня програма: Прикладні інформаційні системи
Спеціальність: Комп'ютерні науки

ПІБ
Філоненко Олексій Вячеславович

Підпис

Назва роботи українською та англійською мовами

Застосунок з використанням віртуальної реальності

An application utilizing Virtual Reality

Мета бакалаврської кваліфікаційної роботи, завдання

Мета роботи: аналіз існуючих технологій віртуальної реальності, порівняння доступних засобів та методів їх використання та створення застосунку віртуальної реальності на основі цих технологій.

План роботи:

1. Проаналізувати існуючі пристрої та програмне забезпечення VR;
2. Порівняти API VR та 3D-рушії для конкретного VR-пристрою;
3. Ознайомитися з процесом створення VR-застосунку від створення прототипу до проведення тестування;
4. Розробити VR-застосунок з використанням сучасних технологій VR.

ПІБ, ступінь, звання наукового керівника роботи: _____

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	26.10.2020	
2.	Видача завдання кваліфікаційної роботи бакалавра	23.11.2020	заява
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	01.12.2020	
4.	Затвердження плану кваліфікаційної роботи бакалавра	18.02.2021	
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	25.02.2021	
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	05.03.2021	
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	09.04.2021	
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	07.05.2021	
9.	Подання роботи у першому варіанті	11.05.2021	
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	12.05.2021	
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	24.05.2021	
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедру	28.05.2021	
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	11.06.2021	
14.	Захист кваліфікаційної роботи бакалавра	25.06.2021	

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1
Завдання до дипломної роботи	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	1
Перелік скорочень, умовних позначень, термінів	1
Вступ	2
Розділ 1	18
Розділ 2	11
Розділ 3	12
Висновки	1
Перелік використаних джерел	2
Додатки	2

	ПІБ	Підп.	Дата	ДП ХХХХ 00.000.00		
Розробн.				Відомість дипломної роботи	Лист	Листів
Керівн.						
Н/контр.						
Зав.каф.	Плескач В.Л.					

АНОТАЦІЯ

Дипломна робота: 55 с., 30 рис., 21 джерел, 1 дод.

Дана дипломна робота присвячена аналізу технологій віртуальної реальності та їх використанню для створення VR-застосунку.

Метою дипломної роботи є аналіз існуючих технологій віртуальної реальності, порівняння доступних засобів та методів їх використання та створення застосунку віртуальної реальності на основі цих технологій.

Для досягнення поставленої мети треба вирішити такі **завдання**:

- проаналізувати існуючі пристрої та програмне забезпечення VR;
- порівняти API VR та 3D-рушії для конкретного VR-пристрою;
- ознайомитися з процесом створення VR-застосунку;
- розробити VR-застосунок з використанням сучасних технологій VR.

Об'єкт дослідження.

Програмні та апаратні технології віртуальної реальності.

Предмет дослідження.

Програмно-технічні, організаційні засади, принципи, підходи щодо створення застосунку з використанням віртуальної реальності.

Методи дослідження.

Дослідження всіх актуальних, на даний час, технологій, що дозволяють працювати з пристроями віртуальної реальності, порівняння найпопулярніших API для VR-пристроїв та 3D-рушіїв для VR-розробки, їх сумісності з широко розповсюдженим VR-пристроєм Oculus Rift S та простоти розробки з залученням основних функцій пристрою.

Ключові слова: VR, Unreal Engine, Oculus, SteamVR, OpenXR.

ABSTRACT

Thesis: 55 pages, 30 figures, 21 sources, 1 appendice.

This thesis is devoted to the analysis of Virtual Reality technologies and their usage for creating a VR application.

The purpose of this thesis is analysis of existing Virtual Reality technologies, comparison of available tools and methods for their usage and the creation of a VR application utilizing these technologies.

To achieve this goal you need to solve the following **tasks**:

- analyze existing hardware and software for Virtual Reality;
- compare VR APIs and 3D engines for a specific VR hardware kit;
- familiarize with the process of creation of a VR application;
- create a VR application utilizing modern VR technologies.

Object of study.

Software and hardware technologies of Virtual Reality.

Subject of study.

Software and technical, organizational principles, approaches to building an application utilizing Virtual Reality technologies.

Research methods.

Study of all modern technologies for working with Virtual Reality hardware, comparison of the most popular APIs for VR hardware and 3D engines for VR development, their compatibility with a widespread VR kit Oculus Rift S and the simplicity of VR development utilizing the essential hardware features.

Keywords: VR, Unreal Engine, Oculus, SteamVR, OpenXR.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ.....	8
ВСТУП.....	9
РОЗДІЛ 1: ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ДЛЯ VR.....	11
1.1 Аналіз сфер застосування VR.....	11
1.2 Доступне апаратне забезпечення для VR.....	15
1.2.1 VR-пристрої комерційного і бізнес-ряду.....	15
1.2.2 VR-пристрої споживацького ряду.....	17
1.3 Особливості розробки застосунків з використанням VR.....	22
1.3.1 Інтеграція з API апаратного забезпечення.....	23
1.3.2 “Повнокімнатна” і стаціонарна дизайн-парадигми у VR.....	25
1.3.3 Реагування на сигнали VR-середовища.....	27
1.4 Висновки до розділу.....	28
РОЗДІЛ 2: АНАЛІЗ ЗАСОБІВ І МЕТОДІВ СТВОРЕННЯ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ.....	30
2.1 Рушії для 3D-застосунків з підтримкою VR.....	30
2.1.1 Unity.....	30
2.1.2 Unreal Engine.....	31
2.2 Порівняння існуючих VR API на прикладі Oculus Rift S.....	36
2.2.1 Oculus SDK.....	36
2.2.2 OSVR.....	37
2.2.3 SteamVR (OpenXR).....	38
2.3 Висновки до розділу.....	39
РОЗДІЛ 3: СТВОРЕННЯ ЗАСТОСУНКУ З ВИКОРИСТАННЯМ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ.....	41
3.1 Створення проекту Unreal Engine та додання підтримки VR.....	41
3.2 Інтеграція основних функцій API VR-пристрою у SteamVR/OpenXR.....	43
3.2.1 Налаштування відстеження сенсорів і відповідних входних подій.....	43
3.2.2 Взаємодія з об’єктами за допомогою контролерів для рук.....	45
3.2.3 Переміщення користувача різними методами.....	45
3.3 Опис MVP проекту.....	46
3.4 Тестування проекту на декількох платформах.....	47
3.5 Публікація у Oculus Store.....	48
3.5.1 Прив’язка Oculus Online Platform до VR-застосунку.....	48
3.5.2 Перевірка відповідності критеріям VRC (Virtual Reality Checks).....	49
3.5.3 Відправлення збірки на перевірку до Oculus Store.....	49
ВИСНОВКИ.....	51
СПИСОК ЛІТЕРАТУРИ.....	52
ДОДАТКИ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

VR (BP) – Virtual Reality, віртуальна реальність

AR – Augmented Reality, доповнена реальність

XR – eXtended Reality, розширена реальність, надмножина VR та AR

Вендор (англ. vendor) — виробник, постачальник програмного чи апаратного забезпечення

OEM – Original Equipment Manufacturer, оригінальний виробник забезпечення

API – Application Programming Interface, інтерфейс програмування застосунку

MVP – Minimum Viable Product, мінімальний достатній продукт (прототип)

ВСТУП

Віртуальна реальність — сфера інформаційних технологій, що стрімко розвивається в останні десятиліття. За звуженим визначенням, VR — технології створення імітованої реальності за допомогою зорових, слухових, тактильних і інших відчуттів. Саме завдяки цим унікальним характеристикам VR-застосунки дозволяють “поринути” у віртуальний світ і відчутти себе його частиною.

Актуальність дослідження. Інтерес до віртуальної реальності не втрачався з початку 21 століття, але лише в останні роки VR почала масово розповсюджуватись не лише як нова платформа для комп’ютерних ігор, але і як інструмент презентації та онлайн-взаємодії в таких сферах, як дизайн, медицина, бізнес тощо. На жаль, інформаційний простір документації та навчального матеріалу по роботі з віртуальною реальністю є досить обмеженим, а оскільки індустрія стрімко просувається і технології для VR-розробки швидко змінюються, час від часу необхідно проводити повторний аналіз існуючих методів та засобів VR-розробки, аби визначити оптимальну комбінацію програмних та апаратних рішень для вирішення нових проблем та реалізації нових ідей. Нові VR-застосунки часто є реалізаціями існуючих процесів для ПК у новому вигляді, а актуальності їм додає можливість прозоро співпрацювати з існуючими сервісами і застосунками для ПК.

Метою дипломної роботи є аналіз існуючих технологій віртуальної реальності, порівняння доступних засобів та методів їх використання та створення застосунку віртуальної реальності на основі цих технологій.

Завдання дослідження:

- проаналізувати існуючі пристрої та програмне забезпечення VR;
- порівняти API VR та 3D-рушії для конкретного VR-пристрою;

- ознайомитися з процесом створення VR-застосунку від створення прототипу до проведення тестування;
- розробити VR-застосунок з використанням сучасних технологій VR.

Об'єктом дослідження у цій роботі є програмні та апаратні технології віртуальної реальності, а **предметом** – застосунок з використанням віртуальної реальності.

Бакалаврська робота складається зі вступу, трьох розділів, висновку та додатків.

РОЗДІЛ 1: ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ДЛЯ VR

Технології VR зазнали помітних змін з початком 21 століття. Шоломи і рукавиці були незручними і дуже дорогими, для них були потрібні найкращі ЕОМ - тепер вони замінені легкими і доступними шоломами (як з зовнішніми камерами, так і без них) та парними контролерами, що комфортно лежать у руці. Раніше віртуальна реальність була доступна лише деяким організаціям, а сьогодні будь-хто може придбати VR-комплект за ціною смартфона і використовувати його як портативно, так і з більшістю конфігурацій сучасних ПК.

В даному розділі проводиться аналіз наступних компонентів предметної області:

- сфери застосування віртуальної реальності, як розважальні, так і промислові;
- основні відмінності розробки для VR від звичайної 3D-розробки для ПК;
- найпопулярніші пристрої та комплекти VR.

1.1 Аналіз сфер застосування VR

- Відеоігри

Найперспективніший напрям. На сьогоднішній день займає ліву частку ринку. За деякими прогнозами, до 2025 року число VR-геймерів виросте до 216 млн.

VR дозволяє розробникам експериментувати та дати гравцям відчувати себе в досі незнаних обставинах. Наприклад, гра Lone Echo отримала нагороду Best VR Game of 2017 на конференції E3 зокрема за унікальний метод

пересування: персонаж гравця знаходиться у середовищі нульової гравітації, де для руху гравець хапається і відштовхується від об'єктів, а фізичний аспект гри базується на масі об'єктів, що взаємодіють.

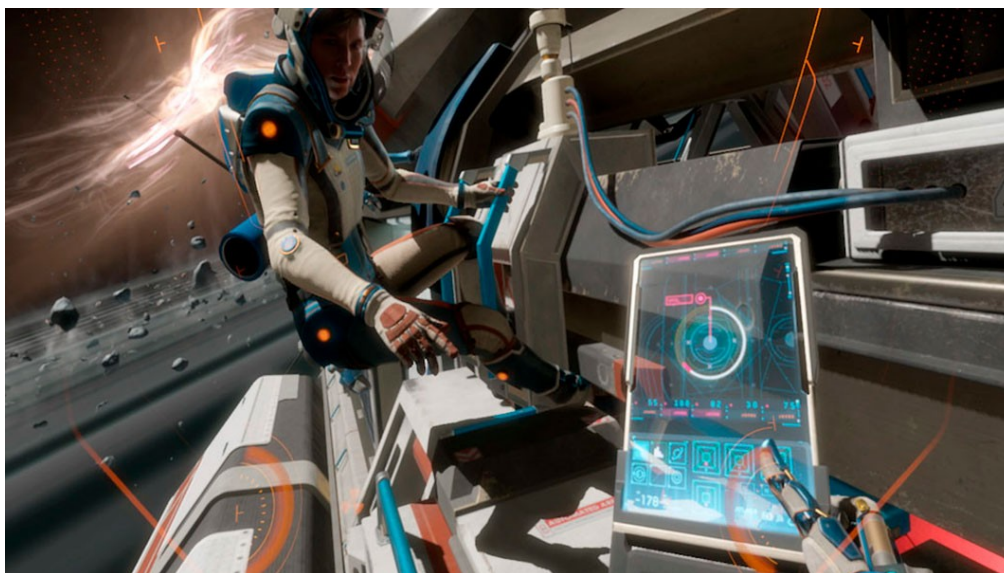


Рисунок 1 - Кадр з гри Lone Echo (зображення: ОнухGame)

Багато відомих ігор, які спочатку виходили лише для традиційних ПК, з часом додали підтримку VR. Платформа розповсюдження ігор Steam, що розробляється компанією Valve, у 2016 році додала підтримку SteamVR – API VR, що базується на OpenXR, завдяки якому ігри, що вже публікувалися в Steam, тепер могли додати підтримку VR на всіх пристроях, що працюють зі Steam. Цим скористалися такі ігри, як The Forest, Subnautica, TESV: Skyrim тощо.

- Парки віртуальної реальності

Найвідоміший приклад - віртуальний парк “The VOID”. Він дозволяє користувачам досліджувати своє фізичне оточення, взаємодіючи з динамічним віртуальним світом. Наприклад, можна відправитися в улюблений фільм і стати улюбленим персонажем.

- Заходи в прямому ефірі

Концерти, спортивні змагання. Завдяки VR користувачі зможуть відчутти ефект живої присутності - навіть перебуваючи за тисячі кілометрів.

- Кіно і серіали.

Швидше за все, вже в найближчі роки ми станемо свідками появи і розвитку нової галузі кінорозваг: глядачі зможуть занурюватися в фільм, а не дивитися його з боку.

- Продажі

Маркетингові рішення на основі VR/AR-технологій вже використовуються деякими провідними магазинами.

- Девелопмент

Перегляд житла в віртуальному просторі дозволить знизити витрати, залучити нових покупців і прискорити терміни укладання угоди.

- Освіта

Взаємодія з предметами в віртуальному просторі - відмінний спосіб підвищити мотивацію учнів і, як наслідок, підвищити якість навчання.

- Медицина

Основне використання VR у медицині сьогодні — інтерактивна візуалізація анатомії людини. Завдяки широкій енциклопедії фактів і медики-початківці, і професійні лікарі, і просто зацікавлені користувачі можуть дізнатися більше про будь-які частини людського тіла, візуалізувати біологічні процеси діяльності людини та навіть поспостерігати чи прийняти участь в операціях.

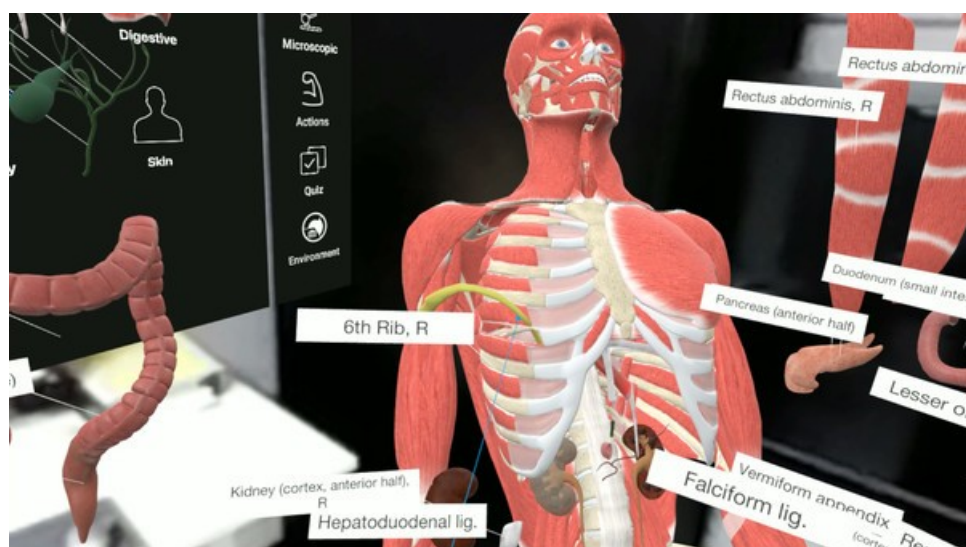


Рисунок 2 - ORGANON VR - застосунок для вивчення анатомії у ВР (зображення: Steam)

Також є потенціал психотерапії: лікування фобій, панічних атак, тривожних розладів, безсоння. На додачу, велике майбутнє у віртуальних консультацій з лікарями.

- Військова промисловість

Відпрацювання дій в надзвичайних ситуаціях, симуляція військових дій, підготовка військових фахівців - все це вже використовується арміями деяких країн.



Рисунок 3 - Використання ВР армією США

- **Проектування**

VR-технології полегшують створення цифрових креслень і уможливають тестування конструкцій у віртуальному просторі.

1.2 Доступне апаратне забезпечення для VR

Кількість різних VR-пристроїв за останні роки продовжує зростати. На ринку наявні пристрої для віртуальної реальності від різних виробників, таких як Oculus, HTC, Microsoft тощо. На зміну старим моделям приходять нові, з покращеними характеристиками і комфортом використання. Більшість з пристроїв, які можна легко придбати в інтернет-магазинах, є представниками споживацького ряду VR-пристроїв, де максимальний потенціал пристрою в контексті занурення користувача у віртуальний світ було зменшено задля доступності і меншої ціни VR-комплекту. Натомість, в бізнесі та інших сферах підприємницької діяльності, де стабільність і якість обладнання є невідмінними передумовами інновацій, використовуються VR-пристрої комерційного ряду, з рядом унікальних характеристик, але також вищою ціною і вузькими каналами розповсюдження.

1.2.1 VR-пристрої комерційного і бізнес-ряду

Для підприємств створюються VR-пристрої вищого класу, що дозволяють повністю перевести деякі робочі процеси у віртуальну реальність. На даний момент великими корпораціями активно впроваджуються програми навчання нових співробітників у VR.

1.2.1.1 Varjo VR-3



Рисунок 4 - Varjo VR-3 (зображення: Holographica)

Varjo VR-3 – ВР-комплект, створюваний компанією Varjo, спрямований на використання у професійному та індустріальному середовищі. Він є третьою моделлю ВР-комплектів від Varjo, і на даний момент має найбільшу роздільну здатність дисплею та кут огляду серед існуючих ВР-комплектів. Основним унікальним функціоналом VR-3 є система відстеження погляду і рук користувача без використання контролерів, що працює з частотою 200 Гц.

1.2.1.2 Pico Neo 3

Neo 3 працює на платформі Snapdragon XR2 від Qualcomm і використовує WiFi 6. У гарнітурі використовуються чотири ширококутні камери, які забезпечують "справжнє позиціонування та відстеження на міліметровому рівні", за словами виробника. Гарнітура використовує просторову локалізацію на основі SLAM (одночасна локалізація та відображення) та відстеження контролера на основі комп'ютерного зору. В поєднанні з 32 датчиками відстеження та контролерами, схожими на Oculus Touch, дана гарнітура є популярним вибором для комерційного застосування, де важлива точність і деталізація.

1.2.2 VR-пристрої споживацького ряду

1.2.2.1 Microsoft HoloLens

Microsoft HoloLens – пара розумних окулярів із змішаною реальністю, розроблених та виготовлених Microsoft. HoloLens є першим пристроєм-шоломом, на якому працює платформа Windows Mixed Reality під управлінням операційної системи Windows 10. Технологія відстеження, що використовується в HoloLens, також використовується у Kinect – доповненні для ігрової консолі Xbox від Microsoft, яка була представлена в 2010 році.



Рисунок 5 - Microsoft HoloLens (зображення: Inexhibit)

HoloLens має формат обруча з розташованими перед очима тонованими лінзами, що мають хвилеподібну структуру, що призначені для заломлення світла і відправлення таким чином зображення з бокових мікродисплеїв в очі користувача. HoloLens потребує калібрації міжзіничкової відстані для використання, а розмір пристрою регулюється за допомогою коліщатка на задній його частині. Зверху розташовані кнопки керування яскравістю та гучністю, які мають попарно різну форму (опуклу і увігнуту), що дає можливість розрізняти їх дотиком. На нижній частині пристрою розташовані

динаміки — завдяки такому дизайну користувач може краще чути звуки з оточуючого середовища, що є обов'язковим в деяких робочих обставинах.

На відміну від більшості інших пристроїв віртуальної, доповненої або змішаної реальності, HoloLens автономні і не вимагають підключення до ПК, смартфона або ігрової консолі.

1.2.2.2 HTC Vive та Valve Index

HTC Vive – перший VR-комплект, випущений компаніями HTC та Valve. HTC Vive був першим VR-комплектом масового виробництва, що використовував технологію повнокімнатного відстеження за допомогою двох базових станцій.



Рисунок 6 - VR-комплект HTC Vive (зображення: Amazon)

1.2.2.3 Oculus Rift S

Oculus Rift S – остання модель серії VR-комплектів для ПК Rift від Oculus, випущена в травні 2019 року. Вона є спадкоємницею оригінальної моделі Oculus Rift CV1 (вийшла у березні 2019 року), включаючи нову позиційну систему відстеження "навиворіт" із камерами, вбудованими всередину гарнітури, дисплей з більш високою роздільною здатністю та новий ремінець

для голови. Oculus Rift S стала першою моделлю VR-пристрою споживацького ряду, що могла використовуватись без базових станцій і додаткових пристроїв-трекерів.



Рисунок 7 - Oculus Rift S (зображення: Oculus)

Rift S отримав неоднозначні відгуки, критики високо оцінили поліпшення комфорту та простоту налаштування завдяки тримачу-обручу та новій системі відстеження, але охарактеризували Rift S як лише поступове оновлення порівняно з CV1, а також помітили регресії, такі як нижчий інтервал оновлення екрану та відсутність апаратного регулювання міжзіничної відстані (IPD).

Для даної роботи був використаний саме цей VR-комплект, до складу якого входить шолом-окуляри і два контролери Oculus Touch для рук. Rift S, на відміну від інших наборів VR для домашнього використання (HTC Vive, Valve Index), не потребує встановлення навколо робочої зони інфрачервоних камер для відстеження контролерів — на самому шоломі присутні 4 камери, що полегшує встановлення і використання набору в побутових та просторово обмежених умовах. На хост-комп'ютері необхідні підтримувані апаратні складові (на ПК для тестування програмного продукту з даної роботи встановлені CPU AMD Ryzen 3600X, GPU AMD RX 5700 XT) та Oculus Software.

Одним з граничних недоліків цього комплекту є його низька портативність: для підключення Rift S до ПК необхідний порт DisplayPort, який відсутній на багатьох сучасних комп'ютерах (особливо ноутбуках), а також він не має зворотної сумісності з HDMI – найпоширенішим стандартом підключення відеопристроїв на сьогоднішній день.

Оскільки Oculus Quest (2) на даний момент є найпопулярнішим VR-комплексом середнього класу на широкому ринку, продажі Rift S на даний момент призупинені, але програмна підтримка від Oculus запланована щонайменше до 2023 року.

1.2.2.4 Oculus Quest

Oculus Quest – це гарнітура віртуальної реальності, розроблена Oculus і випущена 21 травня 2019 року.



Рисунок 8 - Oculus Quest 1 (зображення: CNET)

Подібно до свого попередника, Oculus Go, це автономний пристрій, який може бездротово запускати ігри та програмне забезпечення в операційній системі на базі Android. Він підтримує позиційне відстеження із шістьма

ступенями свободи, використовуючи внутрішні датчики та безліч камер у передній частині гарнітури, а не зовнішні датчики. Камери також використовуються як частина функції безпеки "Passthrough", яка показує вигляд з камер, коли користувач виходить із призначеної зони межі. Пізніше оновлення програмного забезпечення додало "Oculus Link", функцію, яка дозволяє підключати Quest до комп'ютера через USB, що дозволяє використовувати сумісне з Oculus Rift програмне забезпечення та ігри.

Oculus Quest отримав похвалу за ціну та зручність, а також за те, що покращив графічну чіткість і відстеження в порівнянні з Oculus Go, але також отримав критику за свою надзвичайно складну збірку та зниження якості графіки порівняно з VR-іграми на ПК. Під час запуску пристрій також зазнав критики за те, що він містить обмежене програмне забезпечення, доступне у магазині Oculus, і не має зворотної сумісності з програмним забезпеченням Oculus Go. Пізніше введення Oculus Link призвело до повторного росту популярності Quest — критики високо оцінили підвищену гнучкість використання пристрою та зауважили, що такі пристрої, як Quest, швидше за все, замінять гарнітури лише для ПК (такі як Rift) в майбутньому.

13 жовтня 2020 року, як реакція на критику характеристик Quest, вийшов його наступник — Oculus Quest 2.



Рисунок 9 - Oculus Quest 2 (зображення: Habr)

Як і попередник, Quest 2 може працювати як автономна гарнітура з внутрішньою операційною системою на базі Android, так і з програмним забезпеченням VR, сумісним з Oculus, що працює на настільному комп'ютері при підключенні через USB або Wi-Fi. Це оновлення оригінального Oculus Quest з подібним дизайном, але з меншою вагою, оновленими внутрішніми характеристиками, дисплеєм з вищою частотою оновлення, роздільною здатністю на кожне око та оновленими контролерами Oculus Touch.

Quest 2 отримав переважно позитивні відгуки як наступна ітерація серії Quest, але деякі його зміни зазнали критики, включаючи його (для багатьох незручний) ремінець, зменшений доступний інтервал міжзіничкової відстані (IPD) та нову вимогу для користувачів — увійти в обліковий запис Facebook для використання гарнітури.

1.3 Особливості розробки застосунків з використанням VR

Хоча застосунки для VR часто створюються у поширених середовищах 3D-розробки (таких як ігрові рушії), розробникам VR-застосунків необхідно врахувати деталі та основні аспекти сучасної VR-розробки: інтеграція з API

апаратного забезпечення та реагування на сигнали, отримувані від VR-середовища.

1.3.1 Інтеграція з API апаратного забезпечення

Кожний VR-пристрій має набір функцій та параметрів, які він робить доступними для розробників. Це називається інтерфейсом розробника (API) VR-пристрою, і саме він відповідає за передачу VR-застосунку таких даних, як положення шолома у просторі, параметри лінз, натиснення кнопок на контролерах тощо. Окремі пристрої мають власні API низького рівня, на основі яких створюються більш високорівневі API, які дозволяють абстрагувати певні дії у VR, такі як захоплення об'єкту контролером, присідання чи вставання персонажа користувача, а також дозволити застосунку працювати з різними VR-пристроями, що підтримують даний API, без додаткового коду чи драйверів.

На даний момент існує дуальність в сфері уніфікації API для пристроїв віртуальної реальності. Деякі вендори (такі як Oculus) активно просувають власні API на основі існуючих загальнодоступних, що надають користувачам та розробникам ширший функціонал та розвинену взаємодію, але підтримуються лише пристроями від цієї компанії. Наприклад, Oculus SDK (API VR для пристроїв Oculus) сумісний лише з пристроями Oculus, але дозволяє використовувати розширені функції платформи Oculus Platform, такі як захисна система Oculus Guardian, соціальні функції через Facebook тощо.

Натомість, існують API з відкритим вихідним кодом, що спонсоруються та підтримуються низкою компаній та пристроїв. Спільний функціонал між пристроями є набагато меншим (зазвичай, лише підтримка сенсорів шолома і контролерів), але цей недолік переважає багатоплатформовість та широкий ряд моделей, що підтримують такий API. Найрозповсюдженішим стандартом на сьогоднішній день є OpenXR – нащадок OpenVR, створюється консорціумом

Khronos Group (розробником таких відомих стандартів, як Vulkan, OpenGL, OpenCL тощо) та підтримується такими компаніями, як Epic Games, Oculus (Facebook), HTC, Microsoft, Mozilla, Unity, Valve.

OpenXR є тришаровою системою абстракції:

- драйвер від виробника VR-пристрою забезпечує доступ до всіх функцій пристрою в уніфікованій формі;
- середовище платформи VR забезпечує можливість збільшення обсягу API за допомогою розширень API (на даний момент іде активне портування функцій API на кшталт Oculus SDK в формат розширень OpenXR, але жодне розширення ще не є готовим до використання);
- стандартні API Khronos VR призначені для використання кінцевими рушіями і застосунками задля абстрагування над апаратною складовою.

Є декілька реалізацій стандарту OpenXR, такі як Collabora MonoDVR Runtime (GNU/Linux), HoloLens та Oculus PC Platform, але єдиною багатоплатформовою (Windows/MacOS/Linux) реалізацією є SteamVR від Valve.

Саме через легкість використання і покриття всіх платформ Steam API VR SteamVR з дати реалізації стандарту OpenXR (лютий 2021 року) стабільно набирає користувачську базу, особливо він є популярним серед розробників ігор, які публікуються в онлайн-магазині Steam.

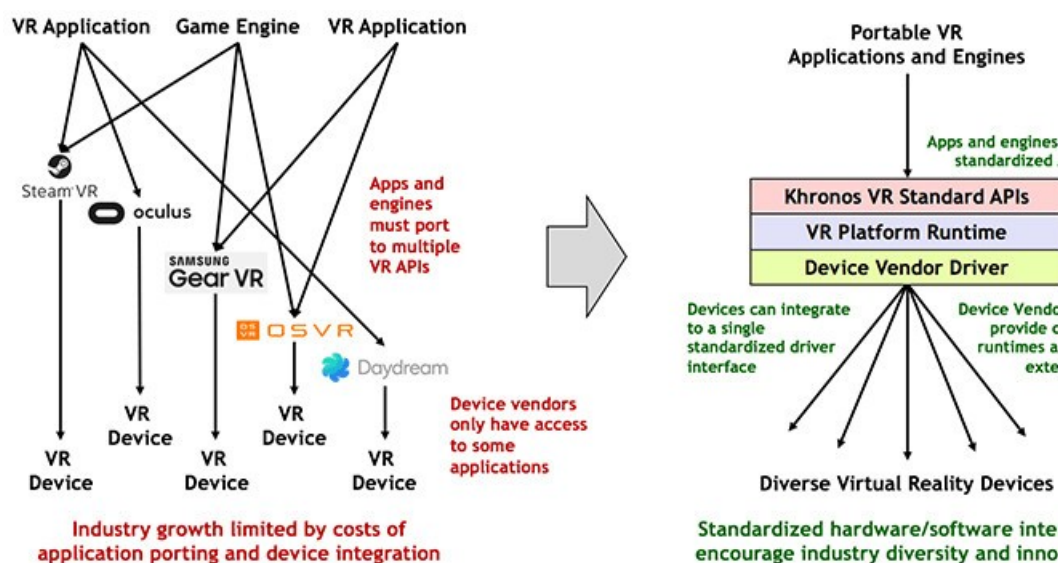


Рисунок 10 - Перехід від розділеної на сегменти індустрії до уніфікації через OpenXR (зображення: Overclockers.ua)

Варто зауважити, що використання декількох API в межах одного програмного продукту є досить розповсюдженим явищем. Наприклад, у розділі 3 створюваний VR-застосунок використовує SteamVR (OpenXR) для орієнтації користувача у просторі і взаємодії з контролерами, а також Oculus SDK для інтеграції функцій Oculus Platform. Таким чином, застосунок може бути опублікованим у Oculus Store з повним функціоналом (онлайн-покупки, соціальні функції тощо), а при портуванні на інші платформи, такі як WebVR, зберегти основний функціонал (взаємодію у 3D-просторі).

1.3.2 “Повнокімнатна” і стаціонарна дизайн-парадигми у VR

Під час дизайну VR-застосунку розробнику варто замислитись над тим, як користувач має розміщуватись у просторі. Існує два основних способи співвідношення координат реального і віртуального простору: шляхом виділення певної зони реального простору та інтерпретація переміщень у ній в VR-просторі — “повнокімнатна” (англ. room-scale) дизайн-парадигма, і шляхом

співвідношення положення пристрою відповідно до початкової точки — стаціонарна дизайн-парадигма.

“Повнокімнатна” дизайн-парадигма дозволяє користувачу більш вільно відчувати себе у віртуальному просторі, де для простих переміщень (крок вперед, назад тощо) немає потреби використовувати програмні засоби переміщення персонажа — можна просто перемістити своє тіло. Натомість, стаціонарна дизайн-парадигма дозволяє використовувати менший простір, підтримується більшою кількістю пристроїв (наприклад, Windows Mixed Reality та Oculus Quest) та дає користувачу можливість сидіти під час користування VR, що є важливою частиною комфорту при довгих сесіях використання.

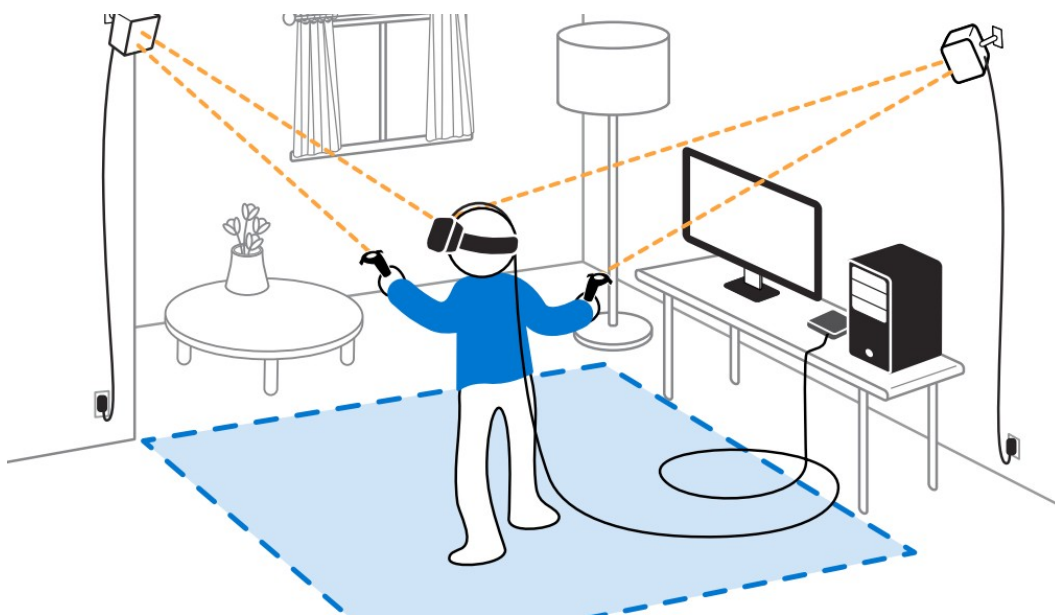


Рисунок 11 - Налаштування доступного простору з використанням базових станцій Lighthouse (зображення: LearnVR)

Для використання “повнокімнатної” дизайн-парадигми середовище користувача має бути підготовлено відповідними програмними продуктами, що поставляються з VR-комплект. Наприклад, VR-комплекти HTC Vive та Valve Index поставляються з двома станціями-датчиками Lighthouse, що відстежують положення шолому і контролерів у просторі та визначають доступний простір. Перевагою такого методу є надійне фіксування меж доступної зони між сесіями

використання, а недоліком — досить високі вимоги до мінімального розміру зони (зазвичай як мінімум 4x4 метри).

Натомість, на шоломах, які мають власні вбудовані камери (Oculus Rift S, Oculus Quest) користувач сам окреслює зону навколо себе перед використанням VR-застосунків за допомогою зображення з камер, а шолом намагається записати деталі навколишнього середовища (освітлення, великі об'єкти), аби відновити розмічену зону в майбутніх сесіях автоматично.

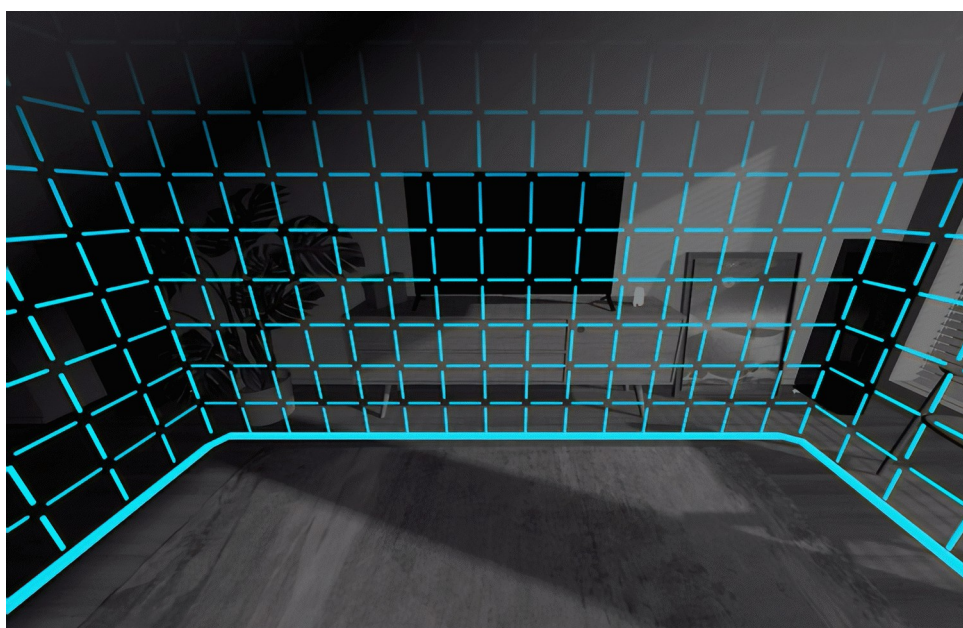


Рисунок 12 - Візуалізація межі Oculus Guardian
(зображення: UploadVR)

Перевагами такого методу є легкість встановлення нових зон невеликого розміру (для розваг, на робочому місці тощо) та невеликий мінімальний розмір зони (2x2 метри для Oculus Rift S та Oculus Quest), а недоліками — неточність розмежування та необхідність час від часу переналаштовувати зону.

1.3.3 Реагування на сигнали VR-середовища

Кожний VR-комплект щонайменше має в своєму складі VR-шолом чи обруч (англ. HMD – Head Mounted Device), який може відстежувати позицію, оберт та

нахил голови користувача відносно доступного простору (що залежить від обраної дизайн-парадигми). При виконанні таких операцій, як переміщення персонажу користувача у VR-просторі, важливу роль відіграє положення шолому відносно центру доступного простору (англ. VR Origin – вихідна точка VR), що дорівнює початковому положенню шолому під час запуску застосунку при використанні стаціонарної дизайн-парадигми, і центру прямокутника розміченої користувачем області (в кімнаті користувача) при використанні “повнокімнатної” дизайн-парадигми.

1.4 Висновки до розділу

Розробка під пристрої віртуальної (а також змішаної) реальності нині є надзвичайно актуальною: VR перестає бути просто “іграшкою для багатих дітей”, і все більше сфер діяльності людей в інформаційному просторі (комерційна, соціальна, розважальна) впроваджують рішення з використанням VR. Поріг входження у VR стає все нижчим, оскільки знижується як ціна VR-комплектів, так і вимоги до ПК.

Розробка для VR базується на стандартних процесах 3D-розробки, але потребує додаткового коду для обробки даних з сенсорів і розширеною взаємодією з функціями пристрою, які є унікальними для VR, за допомогою окремих API. Зараз йде активна стандартизація програмних рішень для розробки під VR в межах відкритих API, і ідея створення застосунку, який працюватиме на всіх популярних VR-платформах, швидко стає реальністю.

Верхня межа співвідношення “ціна-якість” для VR-пристроїв споживацького ряду активно покращується такими великими гравцями індустрії, як Oculus і Valve. Oculus Quest вже є безпроводним і мобільним замінником VR для ПК, а інтерес до нього посилений реаліями віддаленої роботи. Очевидно, що еволюція VR-пристроїв ще не досягла пікової точки, і в

майбутньому на нас чекають більш зручні, потужні та дешевші VR-пристрої. На жаль, піонерами створення нових апаратів є великі корпорації, а не широка спільнота розробників, тому існує небезпека монополізації індустрії та прив'язки до виробника (vendor lock-in); VR-комплекту, який мав би відкриті схеми збірки і був готовим до використання на рівні комерційних рішень, досі не існує.

Відповідно, технології вже є готовими для використання широким колом розробників, і екосистема програмних рішень є достатньо розвиненою як для комфортного прототипування, так і для публікації великих VR-продуктів.

РОЗДІЛ 2: АНАЛІЗ ЗАСОБІВ І МЕТОДІВ СТВОРЕННЯ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

Для написання даної роботи був використаний лише один апаратний комплект — Oculus Rift S – тому в даному розділі буде порівняна підтримка даного флагманського пристрою VR-ринку різними API VR та якість використання цих API основними 3D-рушіями, що підтримують VR-розробку. Тим не менш, за інформацією від провідних особистостей VR-індустрії, всі VR-пристрої споживацького ряду підтримуються основними 3D-рушіями на 100%, тому дані з даного розділу можна теоретично екстраполювати на інші пристрої цієї категорії.

2.1 Рушії для 3D-застосунків з підтримкою VR

2.1.1 Unity

Unity — багатоплатформове середовище розробки комп'ютерних ігор та 3D-застосунків, розроблена американською компанією Unity Technologies. Unity дозволяє створювати додатки, що працюють на більш ніж 25 різних платформах, на кшталт мобільних пристроїв, ігрових консолей, персональних комп'ютерів, браузерних застосунків тощо. Перша версія Unity стала доступною з 2005 року, і хоча з того часу продукт зазнав багатьох змін, розвиток середовища триває і досі.

Розробникам віртуальної реальності Unity пропонує підтримку багатьох платформ (PCVR, Android, iOS), але код для підтримки API відповідних платформ має бути інтегрований самим розробником. Unity лише пропонує легкий доступ до таких API, як Oculus SDK, SteamVR та OSVR.

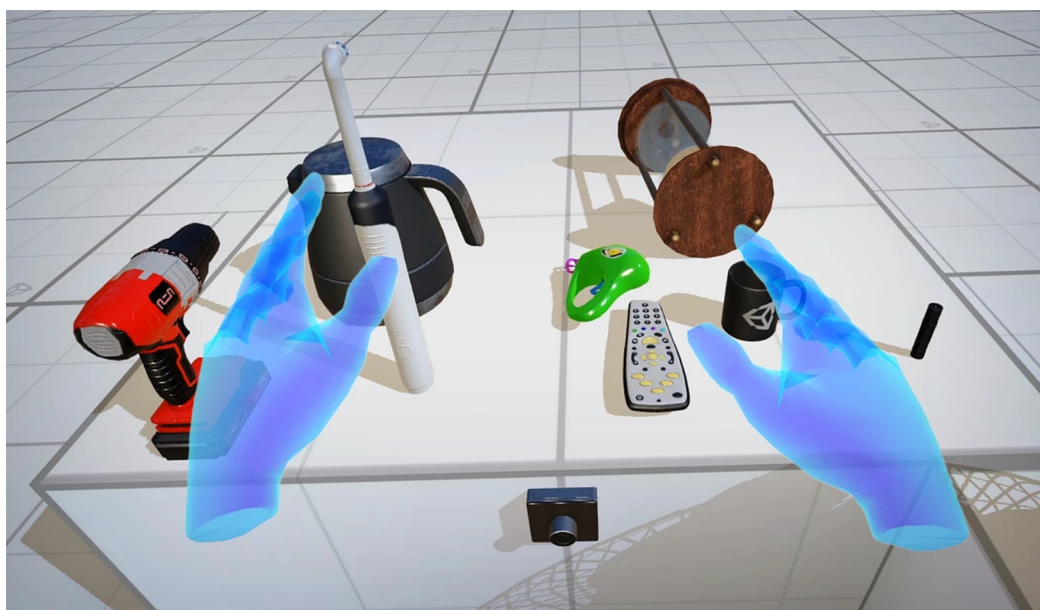


Рисунок 13 - VR-гра на Unity “The Escape Room” (зображення: Unity Learn)

Основними перевагами Unity є підтримка великої кількості сучасних платформ, наявність графічного інтерфейсу для розробки і компонентно-орієнтована система композицій рівнів. Як недоліки часто виділяють втрату швидкодії при роботі з акторами, що складаються з багатьох вкладених компонентів та ускладнений механізм інтеграції зовнішніх бібліотек (DLL/dylib) в застосунок.

На даний момент найактивнішою сферою розвитку середовища Unity є мобільний ринок, в якому Unity є абсолютним лідером як серед менших незалежних студій, так і багатомільярдних корпорацій.

Для даної роботи Unity не був обраний через недостатньо стабільну підтримку SteamVR (OpenXR), через що підтримка пристроїв Oculus на Linux є вкрай важкою чи навіть неможливою. Плагіни для API VR підтримуються або розробниками з Unity Technologies, або громадою, що регулярно спричиняє конфлікт версій з ПЗ деяких VR-пристроїв.

2.1.2 Unreal Engine

Unreal Engine – ігровий рушій, розроблюваний компанією Epic Games. Спочатку рушій використовувався лише для внутрішніх проектів компанії, але з часом став програмним продуктом з відкритим вихідним кодом і почав використовуватися для ігор різних жанрів та перспектив. Також помітною відмінністю є система ліцензування: на відміну від Unity, що безкоштовно надає обмежену версію рушія і потребує дорогої ліцензії для професійного використання, Unreal Engine є безкоштовним з повним функціоналом аж до досягнення розробником застосунку \$1,000,000 на продажах, після чого Epic Games одержує 5% від продажів щоквартально. Така система є зручною для розробників-початківців та невеликих студій, але потребує додаткових фінансових зусиль від великих компаній.

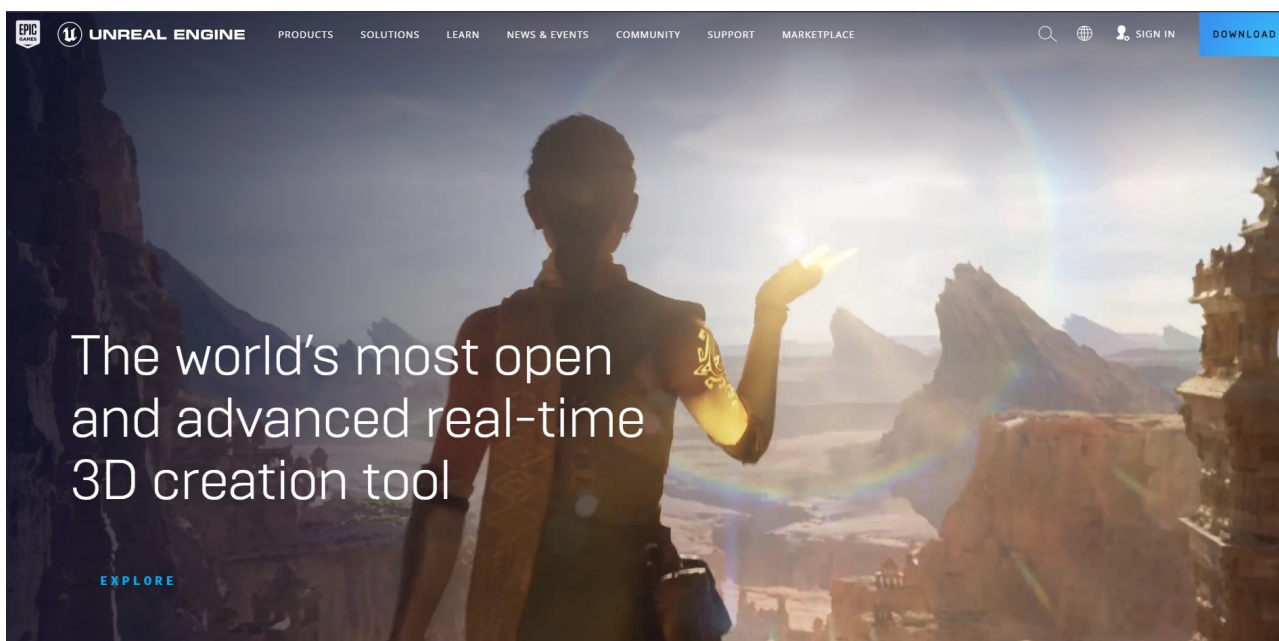


Рисунок 14 - Офіційний сайт Unreal Engine

Рушій використовує стандартну систему компонентів (також наявну в Unity та Godot), але відмінністю є агресивне використання наслідування в стандартних класах Blueprint. Наприклад, клас Actor дає лише можливість

розміщення в світі (позиція, оберт, розмір), його наслідувач Pawn має функціонал для переміщення та базових взаємодій як гравця, так і неігрових персонажів, а наслідувач Pawn – Character дає можливість гравцю керувати Pawn за допомогою стандартних засобів вводу (WASD, геймпад тощо). З цього слідує, що при розробці в Unreal Engine частіше доводиться використовувати перевизначення функцій (overloading), ніж, наприклад, в Unity, де частіше використовується компонування. Тим не менш, інтерфейс Unreal Engine та мова Blueprint реалізують основний функціонал ООП в доступній формі, що полегшує роботу з цією парадигмою без використання C++.

Unreal Engine підтримує основні API VR, зокрема Oculus SDK і SteamVR завдяки системі плагінів, які підтримуються самими вендорами, тому завжди підтримують останні зміни в API. Таким чином досягається швидкість розробки та стабільність VR-застосунків, розроблених у Unreal Engine. Дані плагіни інтегруються з низькорівневими загальними функціями VR, надаваними Unreal Engine, що дозволяє вільно змінювати плагіни під час розробки.

Русій написаний на мові C++, і саме її дозволяє використовувати під час розробки. Але для полегшення прототипування та порогу входження в розробку Unreal Engine також надає доступ до візуальної мови програмування Blueprint. Blueprint – графічна мова, що використовує елементи графів (nodes) та зв'язки між ними для описання входів і виходів функцій, а також потоку виконання (execution flow). Blueprint не є обмеженою у функціоналі, адже компілюється в нативний C++, але має деякі недоліки:

- певні патерни написання коду (цикли, ООП тощо) є незручними в Blueprint і потребують багато елементів і зв'язків для опису;
- код C++, згенерований з Blueprint, має машинно-задані імена та класи, незручні для інтеграції з власним кодом C++;

- також ефективність Blueprint в деяких випадках є в 5-10 разів повільнішою за ідентичний C++ - через це такі ресурсомісткі операції, як обробка шейдерів, освітлення тощо, зазвичай пишуться на C++.

В останніх версіях Unreal Engine 4 є функція під назвою VR Editor, що дозволяє редагувати рівні, освітлення, графі Blueprint, матеріали тощо, не виходячи з інтерфейсу VR-пристрою. Це дозволяє скоротити інтервал впровадження невеликих змін в застосунок і попередньо переглянути їх, навіть не запускаючи застосунок. Також дана функція реалізує цікаві механіки взаємодії з користувачем, використовуючи оберт кистей рук, відстань між руками тощо для операцій над об'єктами у 3D-просторі.



Рисунок 15 - Інтерфейс редагування VR-рівнів в UE4 (зображення: RoadToVR)

Впродовж 2021 року планується вихід наступної версії Unreal Engine 5. Окрім покращення продуктивності, запланований, зокрема, такий функціонал, який може вплинути на індустрію VR-застосунків:

- динамічна система непрямого освітлення Lumen, яка дозволить відображати реалістичне непряме освітлення (дифузія світла) в реальному часі, що до цього було дуже ресурсомісткою операцією;
- система мікрополігонального рендерингу Nanite, завдяки якій вхідні 3D-ресурси (моделі, текстури) можуть бути максимальної якості з мільйонами полігонів, а рушій в реальному часі адаптує якість, рівень деталізації на відстані та рендеринг об'єктів поза кадром для оптимального результату;
- графова система редагування звуку MetaSounds, що натхнена існуючою системою редагування матеріалів Material Editor;
- повна заміна фізичного рушія Nvidia PhysX на власний рушій Chaos Physics, що додасть реалістичності симуляціям фізики руйнування, рідини та волосся.

2.2 Порівняння існуючих VR API на прикладі Oculus Rift S

2.2.1 Oculus SDK

Офіційне API VR від Oculus – Oculus SDK. Oculus SDK (також відомий як Oculus PC Platform) – набір програмних інструментів від Oculus та відповідна документація до інтеграції цих інструментів в такі програмні продукти, як Unity, Unreal Engine та SteamVR. Oculus SDK надає розробникам повний функціонал платформи Oculus, до якого, окрім стандартних функцій відстеження сенсорів та контролерів, що надаються багатьма API, входять такі функції платформи Oculus, як:

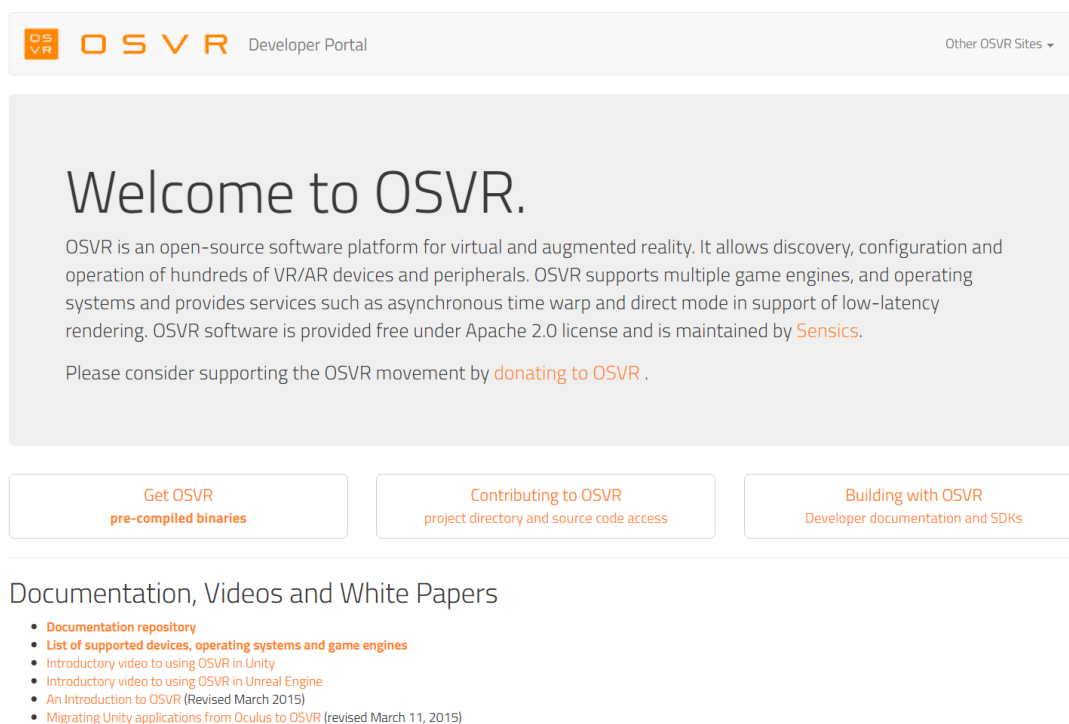
- інтеграція аватару користувача Oculus в VR-застосунок;
- доступ до унікального алгоритму просторового звуку Oculus Spatial;
- соціальні функції: список друзів, голосовий чат, списки лідерів тощо;
- синхронізація прогресу і збереження великих файлів у хмарному сховищі Oculus Cloud;
- перевірка володіння застосунком (Oculus Entitlement Check) та платежі через Oculus Store.

Oculus Rift S повністю підтримується API Oculus SDK і надає користувачам даного пристрою повний доступ до всіх функцій платформи Oculus. Oculus SDK API є стабільним: інтерфейс існуючих функцій змінюється виключно при викладенні нової великої версії (наприклад, 2.0; на даний момент API знаходиться на версії 1.43) і всі оновлення до нього, такі як нові функції платформи Oculus, додаються в API як розширення і не впливають на існуючий функціонал, тому функціонал існуючих застосунків ніколи не порушується.

2.2.2 OSVR

Віртуальна реальність з відкритим кодом (OSVR) - це проект програмного забезпечення з відкритим кодом, розроблений Razer та Sensics, який спрямований на те, щоб гарнітури та ігрові контролери всіх постачальників могли використовуватися з будь-якими іграми.

OSVR має дві основні та незалежні частини: апаратне забезпечення з відкритим кодом та програмне забезпечення з відкритим кодом. Проект в основному фінансується Razer та Sensics. Партнерами проекту є розробники ігор Ubisoft та виробники обладнання Vuzix. Програмна платформа з відкритим кодом дозволяє розробникам віртуальної реальності виявляти, налаштовувати та керувати пристроями віртуальної реальності в широкому діапазоні операційних систем. Вона надається за ліцензією Apache 2.0. З серпня 2018 року доступні вихідні файли електротехнічного обладнання.



OSVR Developer Portal Other OSVR Sites ▾

Welcome to OSVR.

OSVR is an open-source software platform for virtual and augmented reality. It allows discovery, configuration and operation of hundreds of VR/AR devices and peripherals. OSVR supports multiple game engines, and operating systems and provides services such as asynchronous time warp and direct mode in support of low-latency rendering. OSVR software is provided free under Apache 2.0 license and is maintained by [Sensics](#).

Please consider supporting the OSVR movement by [donating to OSVR](#).

[Get OSVR pre-compiled binaries](#) [Contributing to OSVR project directory and source code access](#) [Building with OSVR Developer documentation and SDKs](#)

Documentation, Videos and White Papers

- [Documentation repository](#)
- [List of supported devices, operating systems and game engines](#)
- [Introductory video to using OSVR in Unity](#)
- [Introductory video to using OSVR in Unreal Engine](#)
- [An Introduction to OSVR \(Revised March 2015\)](#)
- [Migrating Unity applications from Oculus to OSVR \(revised March 11, 2015\)](#)

Рисунок 16 - Офіційний веб-сайт OSVR

Підтримка конкретних пристроїв в OSVR реалізується шляхом написання драйвера для цього пристрою. Драйвери для багатьох існуючих VR-пристроїв зараз знаходяться на різному рівні стабільності і завершеності, де флагманськими реалізаціями є драйвери для пристроїв Vuzix та комплектів розробника OSVR HDK. На даний момент існує драйвер OSVR для Oculus Rift, але він не оновлюється з 2017 року. Спроба скомпілювати і встановити драйвер для Oculus Rift S в рамках цієї роботи не була успішною, тому детальний аналіз функціоналу даного API не був проведений. Можна лише зробити висновок, що OSVR є слабо підтримуваним проектом і його використання в сучасних VR-застосунках не є оптимальним рішенням.

2.2.3 SteamVR (OpenXR)

SteamVR – інтерфейс між VR-застосунками і VR-пристроями, що розробляється компанією Valve. З лютого 2021 року SteamVR реалізовує стандарт OpenXR в повній мірі, і тепер є найбільш протестованою і використовуваною його реалізацією.

SteamVR позиціонується як найлегший спосіб поєднати VR-розробників та VR-користувачів у межах онлайн-платформи Steam: VR-розробники можуть створювати застосунки, використовуючи лише цільовий API OpenXR, а користувачі зможуть працювати з цими застосунками, використовуючи будь-які VR-пристрої з підтримуваних SteamVR (такі як Oculus, HTC Vive, Valve Index, Samsung Gear VR, Windows MR Headsets тощо). Незважаючи на таку зв'язаність платформ Steam та SteamVR, будь-який застосунок, навіть не опублікований в Steam, може використовувати API SteamVR, якщо він є встановлений на комп'ютері користувача. Такий підхід дозволяє розробникам почати тестування і ранній продаж застосунків до публікації в Steam, і мотивує їх опублікуватися у Steam задля досягнення ширшої аудиторії.



Рисунок 17 - Офіційний веб-сайт SteamVR

SteamVR також надає додатковий функціонал, такий як:

- “дім” користувача у VR, з інтерфейсом для швидкого запуску VR-застосунків;
- розташування доступної ігровою зони (абстракція над Oculus Guardian та власними інструментами обмеження зони від HTC/Valve);
- підтримка мікротранзакцій через Steam Payments API;
- надання розробникам можливості встановлювати вікна VR-застосунків всередину основного інтерфейсу SteamVR за допомогою VR Overlay API.

2.3 Висновки до розділу

З трьох порівняних 3D-рушіїв лише Unreal Engine 4 відповідає поставленим критеріям: повна інтеграція з низькорівневим API VR-пристрою, підтримка декількох цільових платформ, зручний інтерфейс розробки для VR та велика кількість інформації для нових VR-розробників. Підтримка OpenXR та

Oculus Platform API у Unreal Engine 4 є першокласною, оскільки кожен API підтримується окремим плагіном Unreal Engine і не конфліктує один з одним, що дозволяє використовувати обидва API в межах одного проекту.

SteamVR (OpenXR) на даний момент є єдиним API VR, що дозволяє використовувати функції широкого ряду VR-пристроїв в уніфікованій манері без додаткового коду для конкретних платформ, а Unreal Engine є середовищем розробки, що найкраще підтримує даний API.

РОЗДІЛ 3: СТВОРЕННЯ ЗАСТОСУНКУ З ВИКОРИСТАННЯМ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

В даному розділі описується процес створення, тестування та публікації VR-застосунку з використанням 3D-рушія Unreal Engine 4, мови програмування C++ та графічному інтерфейсі Blueprints, API VR SteamVR (OpenXR) та бібліотек Oculus Platform. Ідея застосунку полягає у перегляді та редагуванні діаграм (pie chart, line chart тощо) у VR, використовуючи куби як найпростіший спосіб представлення даних. Для якісної розробки були проведені наступні роботи:

- створення прототипу з використанням всіх функцій VR-пристрою;
- додання функціоналу, пов'язаному з діаграмами, у прототип;
- тестування VR-застосунку;
- публікація застосунку в магазині VR-застосунків Oculus Store.

3.1 Створення проекту Unreal Engine та додання підтримки VR

Після завантаження Unreal Engine 4 створюється новий проект Blueprints. Налаштування платформи необхідно встановити як Desktop/Console, інакше такі функції, як рероекція втрачених кадрів, будуть відключені.

Blueprints – графічна мова програмування для Unreal Engine, що компілюється в нативний код (на рівні з C++) для цільових платформ і дозволяє розробнику візуалізувати потік дій (реакція на події тощо) та швидко створювати комплексні системи з великою кількістю взаємодій. Blueprints надає доступ до всього функціоналу Unreal Engine, тому, як і C++, не є обмеженою в можливостях. Blueprints надає основні будівельні блоки (керування шляхом

виконання, створення обробників подій, доступ до низькорівневих функцій OpenXR) та дозволяє створювати класи і функції на їх основі.

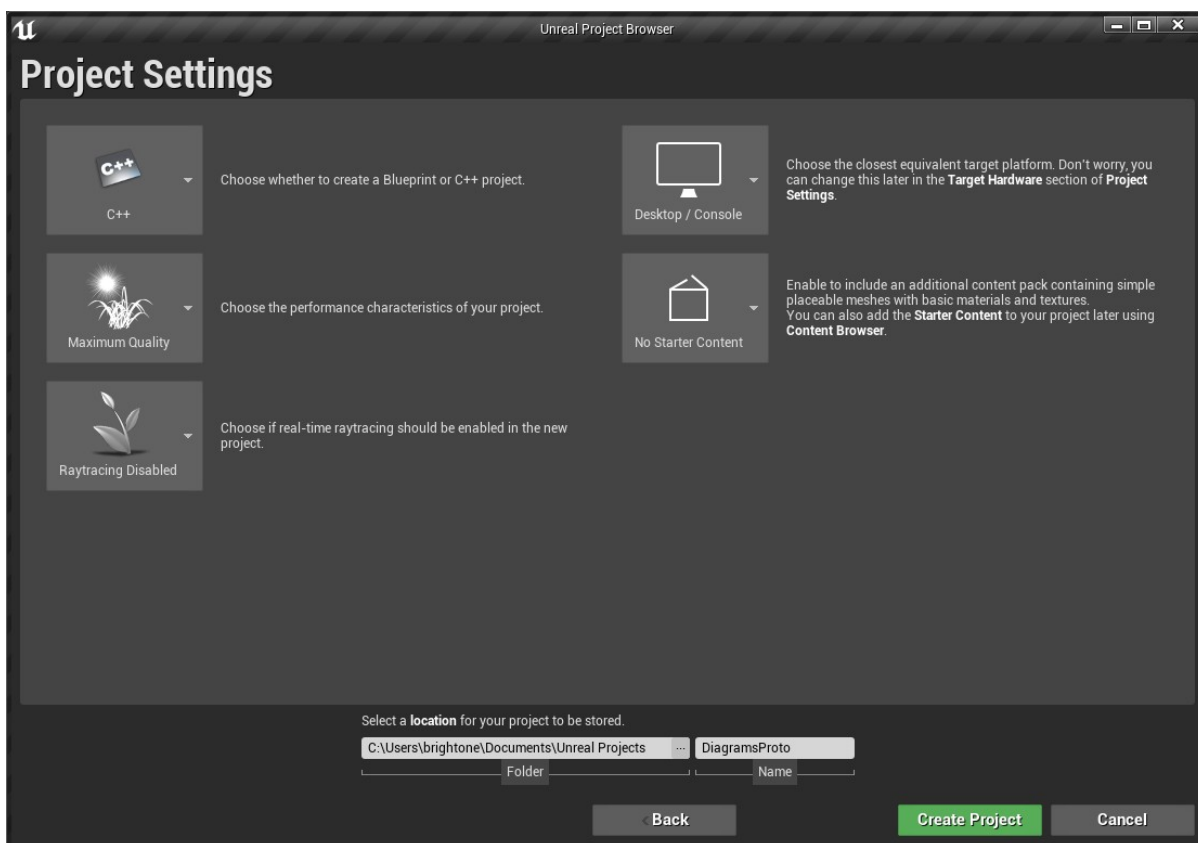


Рисунок 18 - Створення проекту Unreal Engine для VR

Після створення проекту необхідно підготувати два основних класи для VR-застосунку — VRGameMode (відповідає за початкову позицію гравця та умови виходу) та VRPawn (відповідає за персонажа користувача у віртуальному світі). Створюються вони шляхом виклику команди New Blueprints Class.

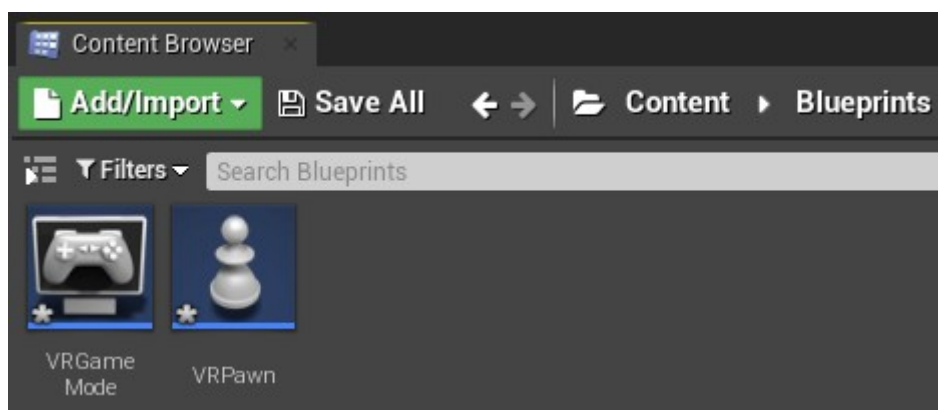


Рисунок 19 - Основні класи VR-застосунку

3.2 Інтеграція основних функцій API VR-пристрою у SteamVR/OpenXR

Перед тим, як описувати бізнес-логіку VR-застосунку, необхідно надати майбутнім програмним модулям застосунку зручний доступ до функціоналу VR-пристрою шляхом абстракції над діями користувача. Unreal Engine надає основні абстракції низького рівня для роботи з сенсорами та основною інформацією від VR-пристрою, але для комфортної розробки необхідно створити власні абстракції (класи, функції) для основних функцій VR-пристрою в контексті розповсюджених методів налаштування взаємодій у VR.

3.2.1 Налаштування відстеження сенсорів і відповідних вхідних подій

В UE4 елементи застосунку, які можуть існувати окремо в світі, називаються Акторами (Actors). Підтип Actor, яким можна керувати інтерактивно (за допомогою геймпаду, клавіатури, VR-пристрою чи машинного інтелекту) називається Pawn (Пішак).

Для VR-застосунку створюється спеціальний клас VRPawn, що наслідує Pawn. Він відповідатиме за створення і розташування контролерів рук та обробку подій від VR-шолому і контролерів, що відповідають переміщенню та діям користувача. Налаштування відстеження подій контролерів рук відбувається у окремому класі VRMotionController.

3.2.1.1 Налаштування початкової висоти VR-пристрою

Кожен VR-пристрій при налаштуванні робочої зони запам'ятовує початкову точку (VR Origin) і посилається на неї при рухах. Розташування цієї точки відносно доступної зони гравця відрізняється для різних пристроїв: наприклад, для Oculus Rift та HTC Vive це рівень підлоги, а для Windows MR та PlayStation VR – рівень очей.

Використовуючи функцію OpenXR GetHMDDeviceName, можна дізнатися назву VR-пристрою і встановити відповідний відступ у випадку відстеження від рівня очей за допомогою функцій UE4 SetTrackingOrigin та AddLocalOffset.

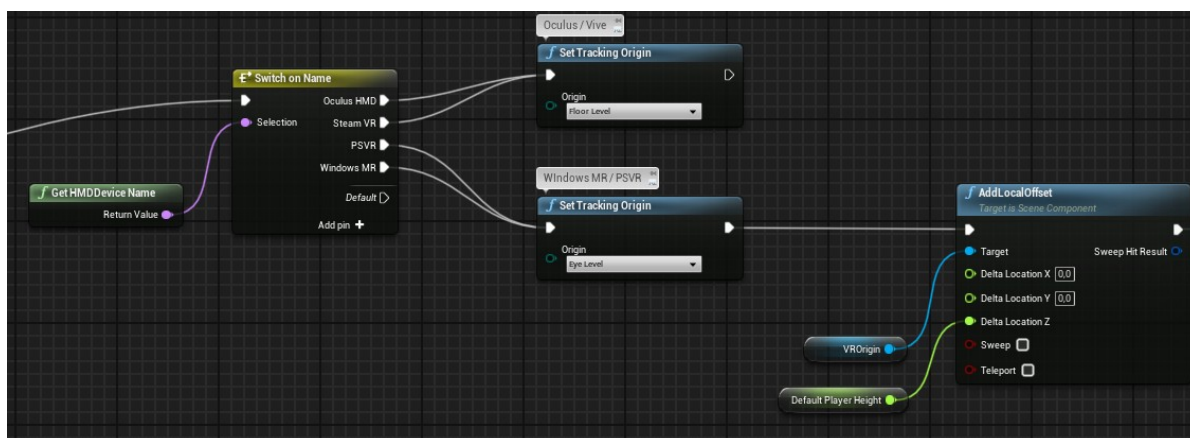


Рисунок 20 - Опис налаштування рівня відстеження

3.2.1.2 Створення об'єктів контролерів рук

Наступною дією є створення контролерів для обох рук. Модель руки, що використовується для візуалізації контролерів у VR, створена для правої руки, тому для лівої вона має бути інвертована функцією SetWorldScale3D. Руки переміщуються разом з гравцем завдяки прикріплення до точки VROrigin функцією AttachActorToComponent.

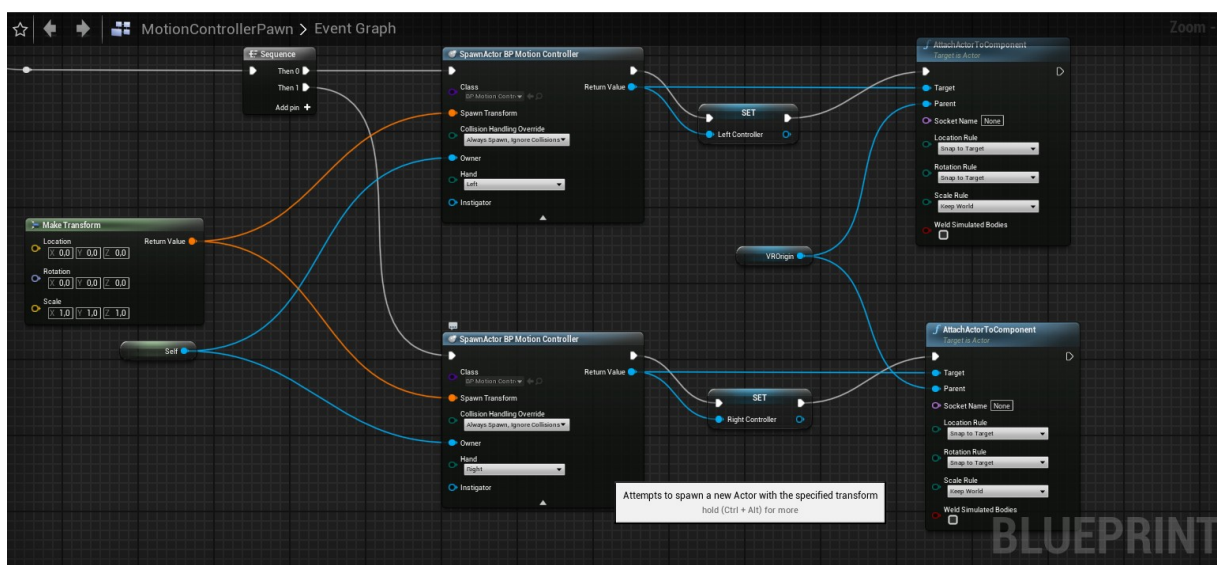


Рисунок 21 - Код створення і прикріплення контролерів до персонажу

3.2.2 Взаємодія з об'єктами за допомогою контролерів для рук

Для виклику подій взаємодії необхідно спочатку описати відповідність вхідних подій VR-пристрою та контролерів рук до динамічних подій в UE4. Це налаштовується в меню Project Settings – Input – Mapping. В даному випадку дії GrabLeft і GrabRight відповідають натисненню кнопки Grab на Oculus Touch, а TeleportLeft і TeleportRight – відхиленню джойстика, що викликає подію телепортації. Код взаємодій описаний в додатку А.

3.2.3 Переміщення користувача різними методами

Для того, аби користувач міг обирати зручний для себе спосіб переміщення у VR, застосунок реалізовує два способи переміщення: телепортація (за замовчуванням) та неперервне переміщення.

3.2.3.1 Телепортація

Телепортація є найлегшим способом переміщення користувача у просторі. Телепортація реалізована шляхом використання джойстиків на контролерах — користувачу необхідно відхилити джойстик, навестися рукою на потрібне місце, за бажанням відредагувати цільовий оберт за допомогою джойстика і відпустити його для завершення телепортації.

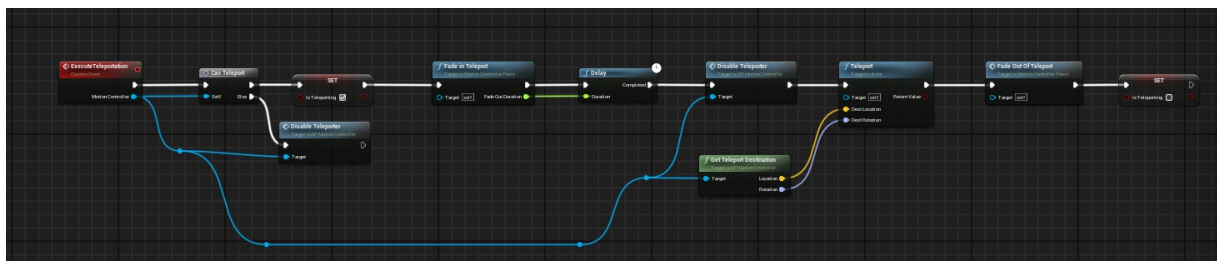


Рисунок 22 - Код VRPawn, що керує процесом телепортації

3.2.3.2 Неперервне переміщення

Під час неперервного переміщення користувач відхиляє джойстик без попереднього натискання на нього (як при телепортації) і рухається в вказаному напрямку відносно напрямку зору — тобто поняття “вперед”, “вліво” тощо визначаються відносно напрямку, в який дивиться користувач. Ці напрямки отримуються функціями `GetForwardVector` і `GetRightVector` на `VRPawn`, а потім проєкуються на підлогу функцією `ProjectVectorOntoPlane`. Проекція необхідна, аби користувач залишався на підлозі, навіть якщо дивиться з нахилом угору чи вниз.

3.3 Опис MVP проекту

Оскільки основний технічний функціонал та складність представленого застосунку знаходиться у роботі з апаратною складовою, програмна логіка була створена виключно достатньою для демонстрації функцій VR-пристрою у заданому контексті. Застосунок дає користувачу можливість використовувати контролери рук для більшості операцій, таких як переміщення і зміна розміру об'єктів.

Користувач може піднімати фізичні об'єкти (які мають компоненти `Static Mesh Collider` для колізії та симулюються фізичним рушієм) за допомогою контролерів для рук, використовуючи кнопку `Grab` на контролері. Також користувач може змінити розмір куба, використовуючи кнопку `Grab` на іншому контролері одночасно і змінюючи відстань від руками. Розмір кубу відображається біля моделі руки, яка його тримає.

Також за допомогою кнопки `Trigger` (під вказівним пальцем на `Oculus Touch`) користувач може відобразити лазерний промінь в напрямі від контролера, використовуючи який можна обрати куб для взаємодії на відстані.

За допомогою лазерного вказівника можна як піднімати, так і змінювати розмір об'єктів.

Дана система є зручною для розширення в майбутньому, наприклад, для розробки взаємодій з 2D-інтерфейсами як безпосередньо натисканнями, так і на відстані.

Як прототип реагування на події від VR-контролерів створений актор CubeContainer, що відображає сумарний параметр Weight кубів всередині та візуалізує його за допомогою елемента-колони. Розмір кубів співвідноситься з параметром ваги за допомогою експоненціальної функції. Контейнер реагує на зміну ваги об'єктів всередині, включаючи ті, що зараз змінюються користувачем.

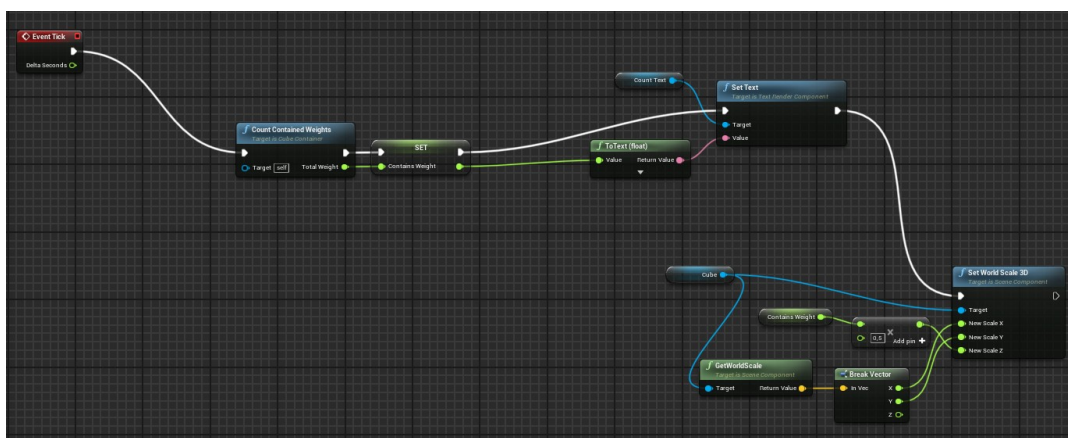


Рисунок 23 - Код візуалізації параметрів кубів CubeContainer

3.4 Тестування проекту на декількох платформах

Тестування було виконане з використанням того ж пристрою, як і для розробки застосунку – Oculus Rift S. API SteamVR (OpenXR) надає можливість використовувати застосунок на декількох підтримуваних платформах без додаткової адаптації коду, а для перевірки даного твердження застосунок було протестовано на двох основних платформах SteamVR – Windows 10 та Linux (Ubuntu 21.04). Цільова платформа для браузерів – WebVR – на даний момент

не підтримується Unreal Engine 4, але підтримка має бути додана у версії 5 даного рушія.

За результатами тестування застосунок проявив себе коректно на обох платформах, зберігаючи всі доступні взаємодії. Розширений функціонал Oculus Online Platform (інтеграція в бібліотеку застосунків Oculus, меню паузи) працює на Windows і коректно не використовується на Linux.

3.5 Публікація у Oculus Store

Щоб перевірити, наскільки легко початківцю в VR-розробці опублікувати власний застосунок, було обрано платформу Oculus Store, зокрема за швидкий процес розгляду заявок та додаткові функції для розробників застосунків в стадії раннього доступу.

3.5.1 Прив'язка Oculus Online Platform до VR-застосунку

Використовуючи функцію `GetEntitlementCheck` від Oculus Platform, застосунок після запуску перевіряє, чи є у користувача доступ (наприклад, застосунок зараз у розробці чи був отриманий через Oculus Store). Якщо перевірка неуспішна, застосунок надає повідомлення про помилку і завершується. Варто наголосити, що додання такої перевірки є обов'язковим критерієм до включення застосунку в Oculus Store.

Після успішного проходження Oculus Entitlement Check застосунок може використовувати розширені функції платформи Oculus, які можуть бути додані після публікації в Oculus Store.

3.5.2 Перевірка відповідності критеріям VRC (Virtual Reality Checks)

Oculus надає розробникам список критеріїв Oculus VRC, яким має відповідати застосунок перед публікацією у Oculus Store. Приклади VRC, що були прийняті до уваги під час розробки застосунку:

- VRC.PC.SDK.4: використання версії Unreal Engine 42+;
- VRC.PC.Functional.3: призупинення застосунку при знятті VR-пристрою;
- VRC.PC.Functional.7: відстеження як позиційних, так і поворотних сенсорів на VR-пристрої;
- VRC.PC.Input.8: для підняття об'єктів контролерами використовувати кнопку Grab, а не Trigger;
- VRC.PC.Security.1: застосунок має викликати Oculus Entitlement Check впродовж 10 секунд після запуску.

На даний момент застосунок відповідає 90% обов'язкових VRC, що є дозволеним для застосунків, які розміщені в каналах розповсюдження Alpha/Beta/RC.

3.5.3 Відправлення збірки на перевірку до Oculus Store

Використовуючи інструмент Oculus Platform Tool, вбудований в Unreal Engine, розробник може завантажити версію VR-застосунку в Oculus Developer Console безпосередньо у редакторі. Для цього необхідно надати:

- Oculus Application ID та Oculus Application Token – своєрідні “логін та пароль” для вашого застосунку, надаються при створенні застосунку в Oculus Developer Console;

- Release Channel – відповідає каналу розповсюдження Oculus Store. Канали Alpha, Beta і Release Candidate не публікуються у Oculus Store (вони існують для тестування застосунку перед публікацією), а канал Store потребує модерації збірки перед публікацією;
- Build Version – номер збірки, що відповідатиме версії застосунку в Oculus Store.

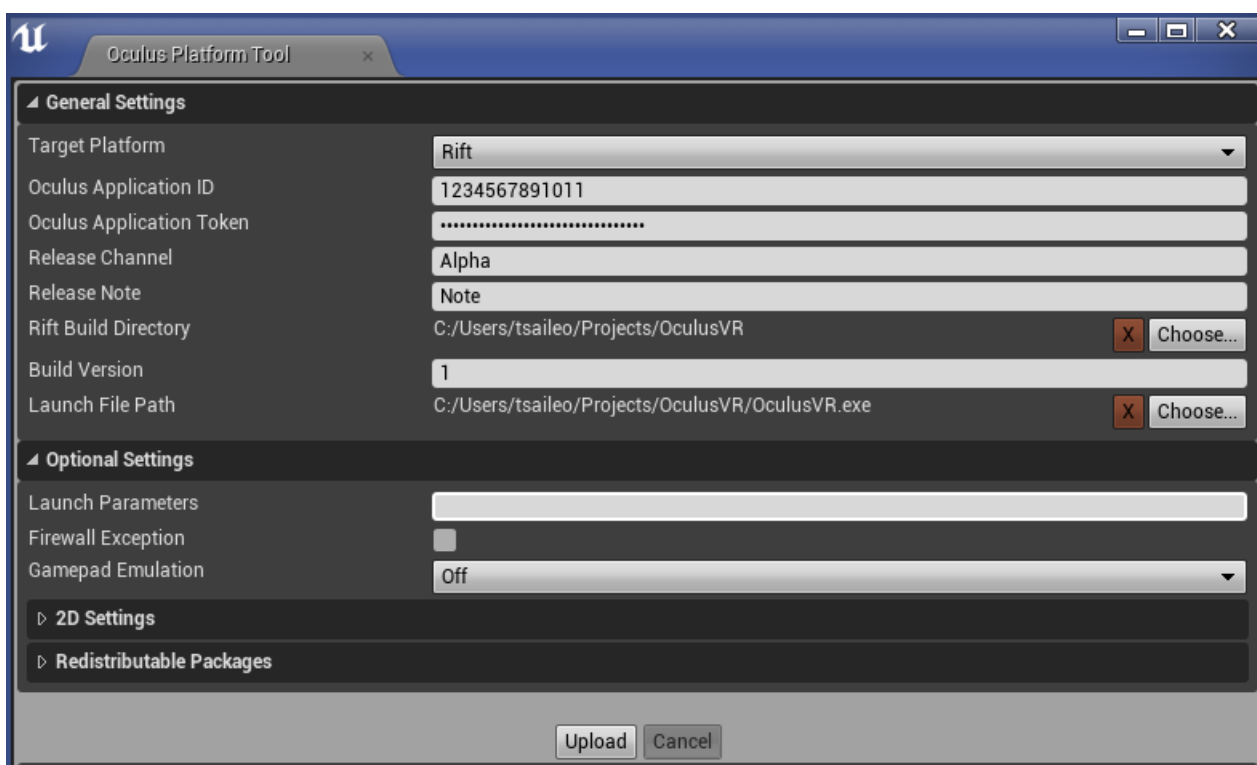


Рисунок 24 - Інструмент Oculus Platform Tool

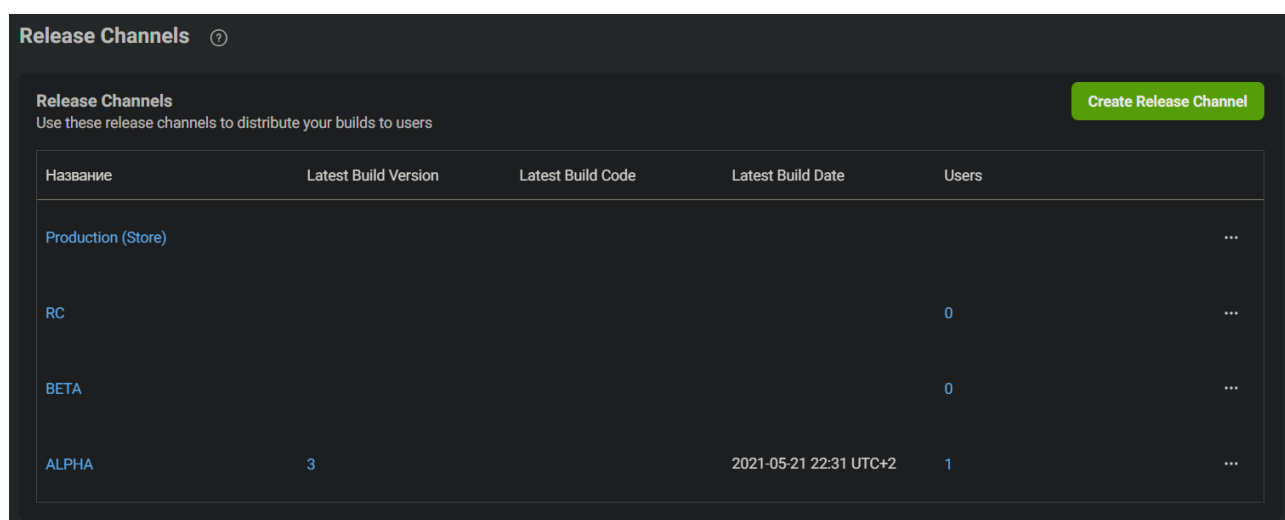


Рисунок 25 - Екран Release Channels в Oculus Developer Dashboard

ВИСНОВКИ

В даній дипломній роботі було проаналізовано та порівняно доступні способи та методи створення застосунку з використанням віртуальної реальності. Були описані поточний стан, переваги та недоліки доступних VR-пристроїв, API віртуальної реальності для них та 3D-рушіїв, що їх використовують: Unity, Godot та Unreal Engine.

Досліджено, які з даних технологій розробки для віртуальної реальності є більш зручними для створення багатоплатформових VR-застосунків. Також було наведено наочні та актуальні приклади використання 3D-рушіїв та API VR у існуючих проектах віртуальної реальності.

Здійснено аналіз та описано процес інтеграції основних функцій VR-пристроїв в межах даних API VR у застосунок.

Був створений застосунок за допомогою 3D-рушія Unreal Engine 4, що використовує API VR SteamVR (OpenXR) та протестовано його на VR-пристрої Oculus Rift S.

Після проведення тестування застосунків було оптимізовано для використання з пристроями Oculus шляхом використання API Oculus Platform та було проведене порівняльне тестування на двох різних цільових платформах — Windows 10 та Linux. Взаємодія з 3D-об'єктами та обробка подій від контролерів працювали коректно на обох платформах.

За результатами роботи можна зробити висновок, що віртуальна реальність, хоча і є відносно новою технологією, вже дає розробникам можливість створювати унікальний досвід для мільйонів користувачів.

СПИСОК ЛІТЕРАТУРИ

1. Jerald J. The VR Book: Human-Centered Design for Virtual Reality – ACM, 2015. 635 с.
2. Novak D., Miheli M., Beguš S. Virtual Reality Technology and Applications – Springer Science & Business Media, 2013. 231 с.
3. Daniela L. New Perspectives on Virtual and Augmented Reality — Routledge, 2020. 322 с.
4. Guazzaroni G., Pillai A.S. Virtual and Augmented Reality in Education, Art, and Museums — IGI Global, 2019. 385 с.
5. Parisi T. Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile — O’Reilly Media, 2015. 151 с.
6. Barfield W, Blitz M. Research Handbook on the Law of Virtual and Augmented Reality – Edward Elgar Publishing, 2018. 689 с.
7. Murray J. Building Virtual Reality with Unity and Steam VR – CRC Press, 2017. 326 с.
8. Linowes J. Unity Virtual Reality Projects — Packt, 2015. 286 с.
9. Davis B.A., Bryla K., Benton P.A. Oculus Rift in Action — Manning, 2015. 440 с.
10. McCaffrey M. Unreal Engine VR Cookbook — Addison-Wesley Professional, 2017. 288 с.
11. Mack K. Unreal Engine 4 Virtual Reality Projects — Packt, 2019. 632 с.
12. Pv S. Unreal Engine 4 Game Development Essentials — Packt, 2016. 266 с.
13. Tavakkoli A. Game Development and Simulation with Unreal Technology — CRC Press, 2017. 672 с.

14. Sherry D., Emperore K. Unreal Engine Physics Essentials — Packt, 2015. 216 с.
15. 3 Dimensional Space and Virtual Reality, Understanding X, Y, Z Coordinates – Режим доступу: <https://learnvr.org/3-dimensional-space-and-virtual-reality-understanding-x-y-z-coordinates/>. Дата доступу: 16.02.2021
16. OSVR Developer Portal | Compatibility – Режим доступу: <https://osvr.github.io/compatibility/>. Дата доступу: 18.02.2021
17. Oculus Documentation | Oculus Developers – Режим доступу: <https://developer.oculus.com/documentation/>. Дата доступу: 05.03.2021
18. Oculus Platform SDK – Режим доступу: <https://developer.oculus.com/downloads/package/oculus-platform-sdk/>. Дата доступу: 11.03.2021
19. Unity XR Interaction Toolkit documentation – Режим доступу: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.9/manual/index.html>. Дата доступу: 16.03.2021
20. SteamVR 1.16 з повною підтримкою OpenXR – Режим доступу: <https://store.steampowered.com/news/app/250820/view/3044967019267211914>. Дата доступу: 20.03.2021
21. Developing for VR in Unreal Engine – Режим доступу: <https://docs.unrealengine.com/en-US/SharingAndReleasing/XRDevelopment/VR/DevelopVR/index.html>. Дата доступу: 26.03.2021

ДОДАТКИ

Додаток А. Код керування контролерами VR_MotionController

