

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки  
Кафедра програмування та математики

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр

спеціальність 121 „Інженерія програмного забезпечення”  
(шифр і назва спеціальності)

спеціалізація „Інженерія програмного забезпечення”

на тему „Реалізація програмних елементів віртуальної реальності у веб застосуваннях ”

Виконав: студент групи ІПЗ-16д

\_\_\_\_\_

( підпис )

В.С. Опришко

\_\_\_\_\_

(ініціали і прізвище)

Керівник

\_\_\_\_\_

( підпис )

Ю.Г. Ковальов

\_\_\_\_\_

(ініціали і прізвище)

Завідувач кафедри

\_\_\_\_\_

( підпис )

В.О. Лифар

\_\_\_\_\_

(ініціали і прізвище)

Рецензент \_\_\_\_\_

Севєродонецьк – 2020

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки  
Кафедра програмування та математики

Освітній ступінь бакалавр  
спеціальність 121 „Інженерія програмного забезпечення”  
(шифр і назва спеціальності)  
спеціалізація „Інженерія програмного забезпечення”  
(назва спеціалізації)

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри ПМ,**  
Д.Т.Н., доцент  
\_\_\_\_\_ Лифар В.О.  
“ \_\_\_\_ ” \_\_\_\_\_ 2020 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ**

\_\_\_\_\_ Опришко Владислав Сергійович \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Реалізація програмних елементів віртуальної реальності у веб застосуваннях

Керівник роботи \_\_\_\_\_ доцент, к.ф.-м..н.Ковальов Юрій Григорійович \_\_\_\_\_,  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)  
затверджений наказом університету від “ \_\_\_\_ ” \_\_\_\_\_ 20\_\_ року № \_\_\_\_\_

2. Строк подання студентом роботи 20 травня 2020 р.

3. Вихідні дані до роботи Середовище розробки Sublime Text 3, Бібліотеки та фреймворки: WebVR, Three.js, Babylon.js, A-Frame.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ, Аналіз предметної галузі, Основна частина, в якій висвітлити: JS бібліотеки Three та Babylon, Фреймворк A-Frame. Програмна реалізація, Висновки, Перелік використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)

## 6. Консультанти розділів роботи

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

7. Дата видачі завдання 30 березня 2020 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної випускної роботи	Строк виконання етапів	Примітка
1	Одержання завдання на виконання роботи	30.03.20	
2	Укладання і погодження з керівником плану і етапів виконання роботи	06.04.20	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	13.04.20	
4	Аналіз шляхів виконання завдання. Вибір і погодження керівником оптимального шляху	20.04.20	
5	Укладання та тестування програмного продукту	27.04.20	
6	Укладання, оформлення та погодження пояснювальної записки з керівником	04.05.20	
7	Здача готової пояснювальної записки на кафедрі	12.05.20	
8	Укладання доповіді і презентації	30.05.20	

**Студент**

\_\_\_\_\_

(підпис)

**В.С. Опришко**

\_\_\_\_\_

(ініціали і прізвище)

**Керівник роботи**

\_\_\_\_\_

(підпис)

**Ю.Г. Ковальов**

\_\_\_\_\_

(ініціали і прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ  
дипломної роботи студента гр. ІПЗ-16д Опришко В.С.

Науковий керівник

Доцент, к.ф.-м..н. \_\_\_\_\_ Ковальов Ю.Г.

Оцінка наукового керівника: \_\_\_\_\_

Рецензент ст..викл. каф. ПМ СНУ ім.В.Даля Ратов Д.В.  
місто роботи, посада, ПІБ

Оцінка рецензента: \_\_\_\_\_

Кінцева оцінка за результатами захисту:

\_\_\_\_\_

Голова ЕК

\_\_\_\_\_

підпис

Лифар В.О.

## РЕФЕРАТ

Текст – 52., рис. – 14, табл. – 3, літературних джерел – 23

Технології віртуальної реальності з'явилися відносно нещодавно, але за досить невеликий проміжок часу набули широкого визнання. Дана кваліфікаційна випускна робота присвячена вивченню підходів та інструментів створення програмних додатків для веб-середовища із підтримкою віртуальної реальності. Було розглянуто способи створення віртуальної реальності у веб застосуваннях та здійснено огляд можливостей рушіїв тривимірного моделювання для створення застосувань віртуальної реальності для веб.

Метою роботи є розробка веб-додатку з підтримкою віртуальної реальності.

Ключові слова: віртуальна реальність, контент віртуальної реальності, 3D модель, тривимірна графіка, WebVR, Three.js, Babylon.js, A-Frame.

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....	9
1.1. Загальна характеристика віртуальної реальності.....	9
1.2. Історія виникнення віртуальної реальності.....	11
1.3. Веб-технології з підтримкою віртуальної реальності.....	14
1.4. Проблеми впровадження віртуальної реальності у веб.....	17
1.5. Висновки до першого розділу.....	18
РОЗДІЛ 2. ГЕНЕРАЦІЯ КОНТЕНТУ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ.....	21
2.1. Аналіз засобів для створення віртуальної реальності у веб застосуваннях.....	21
2.2. Бібліотеки Three.js та Babylon.js.....	27
2.3. Фреймворк A-Frame.....	29
2.4. Порівняльний аналіз засобів для створення віртуальної реальності ..	32
2.5. Створення контенту VR.....	33
2.6. Висновки до другого розділу.....	36
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	40
3.1. Створення тривимірної моделі.....	40
3.2. Веб застосування з підготованим панорамним контентом.....	42
3.3. Веб застосування з генерацією контенту віртуальної реальності.....	44
3.4. Особливості реалізації.....	45
3.5. Висновки до третього розділу.....	46
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50

## ВСТУП

Актуальність. Віртуальна реальність являє собою ілюзію оточуючого нас світу, штучно створеного за допомогою технічних засобів і представленого в цифровій формі. Ефекти віртуальної реальності проєктуються на свідомість людини і дозволяють переживати почуття, максимально наближені до реальних. Сфера віртуальної реальності є однією з найбільш актуальних у світі інформаційних технологій. Причиною цього є те, що, по-перше, на даному етапі розвитку апаратно-програмних засобів реалізація якісних додатків віртуальної реальності, доступних для масового користування, уперше стає можливою, інвестиції та прибутки від області віртуальної реальності постійно зростають [1].

Дана технологія має успіх через можливість її застосування у різних сферах. Цей напрямок уже широко використовується для навчання та симуляційного тренування воєнно-промисловим комплексом, для інженерного проєктування, у галузі охорони здоров'я, а також у рекламі та торгівлі. Інвестиції у дану сферу також показують тенденцію до зростання, а основними вкладниками є такі яскраві представники ІТ-індустрії, як Google (платформи Cardboard, Daydream), Samsung (Gear VR), Sony (PlayStation VR), HTC (Vive), Facebook (Oculus Rift), а також багато інших [2].

Одним з попередніх етапів розвитку інформаційних технологій став прихід у повсякденне життя всесвітньої комп'ютерної мережі Internet. З подальшою глобалізацією та інтеграцією потреба в даній технології продовжуватиме зростати. Існує тенденція до перенесення усіх робочих даних та процесів у мережу, оскільки це швидко, зручно та вигідно. Саме тому сфера віртуальної реальності у веб є надзвичайно перспективною.

Об'єкт дослідження – технології віртуальної реальності.

Предмет дослідження – реалізоване веб застосування з підтримкою віртуальної реальності.

Мета дослідження: розробка веб застосування з використанням віртуальної реальності.

Задачі дослідження:

- 1) Проаналізувати особливості створення віртуальної реальності у веб застосуваннях.
- 2) Визначити існуючі підходи та технології для створення веб застосувань з підтримкою віртуальної реальності.
- 3) Проаналізувати інструменти розробника для створення віртуальної реальності у веб застосуваннях.
- 4) Порівняти інструменти за якісними та кількісними характеристиками.
- 5) Реалізувати веб застосування з підтримкою віртуальної реальності



## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1. Загальна характеристика віртуальної реальності

На сьогодні віртуальна реальність застосовується в тих галузях, де людині для ефективного сприйняття потрібно не просто тривимірне зображення об'єкта (це можна зробити і на плоскому екрані монітора - наприклад, 3D-ігри, дизайнерські та архітектурні пакети), а необхідно занурення спостерігача в світ досліджуваної моделі для більш ефективного сприйняття і взаємодії з нею: там, де об'єкти і дані дуже складні або дуже важлива реальність сприйняття об'єктів. Віртуальне оточення дозволяє людині відчувати себе частиною досліджуваного світу, виключаючи по можливості всі взаємодії зі справжнім, реальним світом (ефект занурення). Цей ефект є специфічною відмінністю від звичайних систем тривимірної графіки, широко доступних на персональних комп'ютерах. На рис. 1.1.1 продемонстровано приклад віртуальної реальності.



Рис. 1.2.1 – Приклад віртуальної реальності

Віртуальна реальність – створений технічними засобами світ, який передається людині через його відчуття: зір, слух, дотик і інші. Сучасні технології дозволяють імітувати справжню реальність за допомогою різних мультимедійних пристроїв, які впливають на слух, зір, нюх та інші відчуття. Занурення у віртуальну реальність може проводитися за допомогою абсолютно різних пристроїв, проте найбільш ефективним засобом є компоновка проекційних систем, що називається кімнатою віртуальної реальності. На стіни такої кімнати в реальному часі проектується 3D-стереозображення. Більш того, користувач подібної системи може взаємодіяти з віртуальним середовищем за допомогою різних пристроїв, які відстежують рухи. Таким чином, виникає досить сильне занурення у віртуальне оточення.

Метою даної технології є створення такого досвіду, який викликає у користувача відчуття присутності в штучному світі, реальності середовища та максимальної наближеності до реального світу, а також реалізація можливості штучної віртуальної реальності реагувати на дії користувача. В результаті виконання поставлених перед віртуальною реальністю задач буде створено такий інтерфейс взаємодії, який одночасно з властивою йому реалістичністю буде максимально зручним та інтуїтивно зрозумілим, що надаватиме йому перевагу при виборі користувачами технології взаємодії.

Таким чином, системи віртуальної реальності представляють новий щабель в технічному розвитку людства. Не всі технічні проблеми в цій галузі вирішені, але початок вже покладено, а далі вже справа часу. Іммерсійна система віртуальної реальності надає надзвичайно універсальні інструменти для втілення фантазії людини. Надалі вони стануть доступніші для масового споживача, а програмне забезпечення зробить спілкування користувача з віртуальним середовищем інтуїтивно зрозумілим.

## 1.2. Історія виникнення віртуальної реальності

Ближче до середини 1950-х років кінематографіст на ім'я Мортон Хейліг вирішив створити щось дивовижне, що вразило б публіку. У 1960 році він сконструював щось на зразок ігрової приставки і назвав її Сенсорама (рис. 1.2.1). У комплекті до приставки йшов стереоскопічний екран, кулера, емітери запаху, стерео колонки і крісло, що рухається.



Рис. 1.2.1 – Сенсорама Мортон Хейліга

Він також створив спеціальний телевізор, що дозволяє дивитися програми в тривимірному вимірюванні. У той час глядачі ще пасивно дивилися фільм, але це був перший крок до розвитку віртуальної реальності.

У 1961 році інженери корпорації Philco розробили перший шолом-дисплей, який отримав назву Headsight (рис. 1.2.2). Шолом складався з відео

екрану і системи стеження, яку інженери під'єднали до замкнутої системи відеостеження. Цей шолом був призначений для використання в небезпечних ситуаціях – користувач міг спостерігати реальну навколишнє середовище як би віддалено, регулюючи ракурс гортанням голови. Подібний пристрій використовували і пілоти. До шолома під'єднували інфрачервоні камери, прикріплені до основи літаків, які дозволяли пілотам орієнтуватися на місцевості навіть в нічний час доби.



Рис. 1.2.2 – Перший шолом віртуальної реальності

У 1965 учений на ім'я Іван Сатерланд створив новий пристрій "Ultimate Display." За допомогою цього дисплея людина могла заглянути в віртуальний світ, який йому здавався справжнім реальним світом. Його винахід дав новий поштовх для розвитку цілої науки про віртуальному просторі. Концепція Сатерланда: віртуальний світ, відтворений через шолом-дисплей, який повинен здаватися реальним будь-якому спостерігачеві.

Сатерланд сконструював нову модель шолома, яка вже під'єднується до комп'ютерної системи (раніше шолом підключався тільки до камер).

Комп'ютер формував зображення і відображав їх на екрані. Нова модель шолома могла одночасно із зображенням відтворювати стерео звук, що створювало просторову ілюзію, і коли людина повертав голову, миттєво змінювалося зображення.

З розвитком науки віртуальної реальності НАСА, Міністерство оборони і Національний науковий фонд почали фінансувати велику частину наукових досліджень в даній області. ЦРУ пожертвував 80 000 \$ в дослідження Сатерланда. Перші досягнення в цій галузі науки використовували головним чином з метою навчання пілотів.

Довгі роки розвиток цієї галузі уряд тримав у секреті. Тільки в 1984, коли вчений по імені Майкл Макгривай розробив «людино-машинний інтерфейс», ЗМІ швидко поширили цю новину майже по всьому світу. Людино-машинний інтерфейс до сих пір грає важливу роль в дослідженнях віртуальної реальності. Однак, сам термін «віртуальна реальність» запропонував Джерон Ланіер в 1987 році. У 1990-ті роки ЗМІ підхопили концепцію віртуального світу і переконали всіх людей в тому, що це нове досягнення науки може повністю змінити наше уявлення про життя. Поступово багато людей зрозуміли, що їх просто обдурили, оскільки пристрої віртуальної реальності були ще мало розроблені і не могли робити те, про що говорилося в ЗМІ, тому поступово інтерес народу до цієї галузі згас. Сьогодні розробники віртуальної реальності не роблять гучних заяв про нові можливості, як це було раніше, і навіть намагаються не вживати термін «віртуальна реальність». Але хто знає, можливо, вже в недалекому майбутньому віртуальна реальність в якійсь мірі замінить наш світ.

### 1.3. Веб-технології з підтримкою віртуальної реальності

Останнім часом з'являється все більше веб-браузерів з підтримкою віртуальної реальності (VR-браузери). Наприклад компанія Mozilla випустила браузер Firefox Reality 1.0, спеціально розроблений для віртуальної реальності (зображений на рисунку 1.3.1), компанія Google – браузер Chrome (оновлення 79). Підтримку VR-контенту також отримують інші браузери на базі Chromium, такі як Oculus Browser, Edge і Magic Leap's Helio. Таким чином, веб-розробники мають змогу створювати VR-контент (в тому числі ігри, відео 360 градусів і т.д.). Сайти можна буде переглядати за допомогою гарнітури віртуальної реальності для смартфона або за допомогою VR-шоломів.



Рис. 1.3.1 –VR-браузер Firefox Reality 1.0

Для веб-технологій, віртуальна реальність є одним зі способів взаємодії користувача з мультимедійною інформацією, а також іншими людьми в режимі реального часу з використанням мережевих технологій. Цей спосіб цілком незвичний для інтернету, оскільки зазвичай користувачу доводиться працювати з плоским двовимірним графічним інтерфейсом, який на даному етапі досягнув високого рівня зручності та задоволення потреб. Лише в деяких

випадках може використовуватися тривимірний контент, але лише як демонстраційний матеріал для чітко визначених та обмежених цілей, наприклад, об'ємна демонстрація товару в інтернет-магазині.

Аде на відміну від звичного способу керування, віртуальна реальність дає змогу повністю зануритися у тривимірний світ, який оточує користувача за допомогою 3D-моделей. Взаємодія у цьому випадку буде відрізнятися від стандартної для веб, оскільки тут немає таких пристроїв введення-виведення, як миша та клавіатура. На зміну їм приходять більш інтуїтивні засоби, такі як рух голови та керування об'єктами віртуальної реальності за допомогою спеціальних маніпуляторів. І шолом, і маніпулятори містять датчики руху, тому можуть ефективно реалізовувати інтуїтивно-зрозумілий інтерфейс та більш швидко та гнучку роботу.

Аде також слід розрізняти віртуальну реальність та звичайні плоскі об'єкти інтерфейсу користувача, які приведені до псевдо-тривимірного вигляду [5]. Досить часто такими є вікна VR-браузерів, які фактично є панелями, і майже нічим не відрізняють від вікон десктоп-браузерів, за виключенням того, що вони тепер змінили своє положення. На сьогодні, це єдиний варіант реалізації роботи з браузерами, і пов'язано це з тим, що, по-перше, він є звичним інтерфейсом для користувачів, які будуть переходити на нову платформу, а по-друге, ще немає розробок щодо нової моделі інтерфейсу, який зміг би задовольнити потреби оптимальнішим чином [6]. На даний момент здійснюється спільна робота консорціуму W3C, Mozilla, Google, а також багатьох інших розробників та дизайнерів, які працюють в напрямку VR у web, над розробкою стандарту для такого інтерфейсу взаємодії. Таким чином, досить скоро користувачі зможуть відкрити для себе новий спосіб роботи в Інтернет.

Інтерфейс сучасних застосувань віртуальної реальності базуються на графічному контенті, який змінюється у відповідь на зміну положення голови користувача. В деяких випадках даний інтерфейс доповнюється звуковою

інформацією. Google уже опублікували застосування для створення фотографій віртуальної реальності на платформі Android без необхідності використання спеціалізованого апаратного забезпечення [3]. На разі немає стандартних вбудованих широко розповсюджених технологій для створення тактильних відчуттів, запахів тощо, хоча в даному напрямку проводяться розробки. Але найголовнішим напрямком, в якому здійснюється робота над інтерфейсом віртуальної реальності, це створення так званого «room-scale VR». Дана технологія полягає в реакції середовища віртуальної реальності не лише на положення голови, а й на переміщення тіла користувача у просторі [4]. Цей підхід реалізовується встановленням у приміщення, де буде здійснюватися робота з додатком віртуальної реальності, камер та/або інфрачервоних датчиків, які фіксують переміщення користувача, передають зображення обчислювальному вузлу, який визначає зміну положення тіла та передає результуючі дані додатку віртуальної реальності, що в сукупності забезпечує реакцію середовища віртуальної реальності в режимі реального часу. Основними недоліками даної технології є перш за все те, що для її роботи необхідно придбання та встановлення додаткової апаратури, яка на даний момент ще досить дорога, а також виникає складність у використанні даної технології, оскільки вона вимагає певного вільного простору в декілька квадратних метрів, бо інакше буде користувач буде зіштовхуватися з об'єктами реального світу. Саме через вищезгадані недоліки більшість сучасних застосувань віртуальної реальності розроблено для стаціонарної парадигми віртуальної реальності, де фіксуються зміни лише положення голови.

Користувач взаємодіє із середовищем віртуальної реальності за допомогою контролерів віртуальної реальності, а для випадку, якщо їх немає, було прийнято стандартний інструмент – курсор, який знаходиться в центрі зображення й виконує функцію вибору об'єкту (натиснення кнопки миші). Цей курсор спрацьовує, якщо буде протягом деякого часу затриманий на об'єкті середовища віртуальної реальності.



#### 1.4. Проблеми впровадження віртуальної реальності у веб

На сьогодні технологія віртуальної реальності для веб-застосувань, як і будь-які інші, має певні причини, які на початковому етапі її розвитку заважають увійти в повсякденний обіг та набути масової популярності. Першою проблемою є необхідність у використанні додаткової апаратури. Обов'язковим є шолом або окуляри віртуальної реальності. Шолом віртуальної реальності містить датчики, які реагують на рух голови, а сам він під'єднується до пристрою, який виступає обчислювальною платформою, що оброблює отримані дані й транслює зображення на дисплей.

Такими платформами зазвичай виступають персональний комп'ютер або ігрова консоль. Слід зазначити, що окуляри віртуальної реальності дають змогу використовувати смартфон у якості платформи, і з його екрану відбувається трансляція зображення. Єдиною вимогою до телефону є наявність датчику гіроскопу, а цим датчиком забезпечуються майже всі нові моделі телефонів. Звичайно, також бажано, щоб у смартфона був екран з високою роздільною здатністю, це забезпечить якість зображення. І саме винахід другого способу реалізації віртуальної реальності збільшив її популярність, оскільки значно здешевив вартість. Більше того, окуляри віртуальної реальності можна виготовити самому за допомогою підручних матеріалів [7].

Для обох даних підходів до створення апаратного забезпечення є характерний недолік, який полягає в тому, що зображення, спроектоване на екран, має затримку відображення відносно рухів голови, і тому досить часто у користувачів спостерігається погіршення самопочуття, рябіння в очах та морська хвороба при зануренні у віртуальну реальність. Для вирішення цієї проблеми розробники апаратної складової застосовують такі засоби, як сенсор з більшою частотою дискретизації, заміна LCD-панелей на OLED з низьким періодом відображення, передбачення руху “time warp” [8].

Хоча для досягнення потрібного результату слід впроваджувати покращення також з боку програмних модулів. Так, якщо буде здійснюватися глибоке вивчення технологій програмування даної сфери, то можна буде проаналізувати причини поточної якості та спроектувати системи, які на програмному рівні будуть оптимізовані для поставлених задач.

Більш того, іншою причиною є те, що зараз даний сегмент, у зв'язку з новизною, ще не набув поширення. Так, розробники ПЗ не поспішають витратити час та зусилля на вивчення нової технології, яка ще не дала стовідсоткової гарантії на майбутнє поширення та панування. У свою чергу, бізнес не створює запити на спеціалістів та послуги розробки контенту віртуальної реальності, оскільки далеко не всі користувачі мають необхідні засоби для його перегляду й немає фахівців зі створення такого продукту. А якщо немає попиту на даний вид послуг розробки, то не буде перекваліфікації. Таким чином, з'являється циклічність. На вирішення цієї проблеми стали вищезгадані корпорації Google, Facebook, Samsung, HTC та інші, які своєю підтримкою проектів віртуальної реальності «диктують моду» для всіх інших фірм, заставляючи їх звертати свою увагу на VR-розробку. Тому зараз саме час підтримувати розвиток даної технології, оскільки, по-перше, це стане майбутньою конкурентною перевагою, а по-друге, дасть можливість бути в авангарді серед тих, хто першими запровадить дану технологію в повсякденний обіг.

### **1.5. Висновки до першого розділу**

У сучасному світі Інтернет є засобом обміну будь-якої мультимедійної інформації – фото, аудіо чи відео, а також інформації, інтерактивної по відношенню до користувача. І в той же час, мережа є великим інструментом для ведення бізнесу: від демонстрації товарів та послуг до проведення транзакцій з купівлі/продажу. Необхідність у дослідженні та розвитку даної технології є дуже великою. Веб вміщує в собі всі сучасні наявні різновиди мультимедійних даних. Але він не передбачає способів взаємодії з ними як з

об'єктами реального світу, тобто так, як це звично для людини. Віртуальна реальність вирішує цю проблему. Крім того, що вона надає можливість переглядати VR-медіа, вона дозволяє взаємодіяти з об'єктами подібно до матеріальних. Таким чином, відбувається покращення як якості контенту, так і інтерфейсу користувача. Можна для прикладу розглянути ситуацію, коли людина хоче здійснити покупку одягу в інтернет-магазині та має на вибір два способи: звичайний, де можна побачити лише фото товару на манекені і, можливо, покрутити його тривимірну копію вздовж однієї заданої осі координат, та можливість поглянути, як ця річ буде виглядати на ній особисто, з можливістю вивчити товар з різних ракурсів. Очевидно, що буде обрано другий.

Більш того, на разі вищезазначеною командою розробників та дизайнерів здійснюється спільна робота над створенням моделі єдиного уніфікованого інтерфейсу для роботи з Інтернет у віртуальній реальності. Це передбачатиме не лише «точковий» досвід інтеракції з мультимедіа-контентом у віртуальній реальності, а й новий спосіб пошуку інформації, обміну повідомленнями, ведення бізнесу, і це все в режимі поглинаючої віртуальної реальності.

Отож, технологія створення віртуальної реальності в мережі вже зараз має якісні напрацювання, які обіцяють стрімкий та перспективний розвиток. Не зважаючи на те, що на даному етапі цій сфері присутня достатня кількість труднощів та задач, які потребують обов'язкового вирішення, технологія продовжує стрімко розвиватися та набирати популярність серед користувачів, замовників та розробників. Маючи необхідні передумови та потужності для реалізації розробок та рішень, віртуальна реальність у веб в найближчому майбутньому обіцяє завоювати великий сегмент ринку та стати однією з найбільших індустрій в ІТ в цілому та серед Інтернет-технологій зокрема.

Таким чином, з огляду на всі вищезазначені аргументи, можна зробити висновок, що дослідження технології створення віртуальної реальності у веб є дуже перспективним напрямком, який однозначно варто досліджувати

## РОЗДІЛ 2. ГЕНЕРАЦІЯ КОНТЕНТУ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

### 2.1. Аналіз засобів для створення віртуальної реальності у веб застосуваннях

Наразі технології віртуальної реальності для веб-застосувань можна умовно виділити два підходи до розробки додатків: поділ ґрунтується на контенті віртуальної реальності, який використовується в застосуванні. Відповідно до цього, можна умовно виокремити першу групу додатків, які використовують підготований VR-контент, та другу групу додатків, які генерують інтерактивний 360°-контент в режимі реального часу роботи за допомогою потужностей інструментів розробника.

Звичайно ж такий поділ є досить умовним, оскільки для першого випадку, коли веб застосування використовує готовий VR-контент і відображає його користувачу, все одно для реалізації даної задачі повинен використовуватися спеціальний інтерфейс, який взаємодітиме з апаратним забезпеченням та зробить можливим коректне відображення даного контенту на пристрої віртуальної реальності. Але в той же час, другий підхід, для якого характерне створення середовища віртуальної реальності під час роботи додатку, досить виразно відрізняється від описаного вище, оскільки передбачає використання додаткових спеціалізованих інструментів для реалізації даної задачі та, відповідно, а додатки передбачають абсолютно інакшу архітектуру.

Таким чином, як було зазначено вище, для реалізації віртуальної реальності в мережі при будь-якому підході необхідне використання API, яке забезпечить коректну взаємодію додатку з пристроєм.

Раніше інтернет-спільнота вже експериментувала з віртуальною реальністю, а саме з VRML (англ. Virtual Reality Modeling Language – мова моделювання віртуальної реальності), але потім з'явився WebVR, що

представляє зовсім інший підхід до віртуальної реальності, який більше підходить для сучасного інтернету.

У 2011 році з'явився WebGL – кросплатформовий API для 3D-графіки в браузері, що розробляється некомерційною організацією Khronos Group. WebGL використовує мову програмування шейдерів GLSL та являє собою елемент HTML5 і тому є повноцінною частиною об'єктної моделі документа (DOM API) браузера. Може використовуватися з будь-якими мовами програмування, які вміють працювати з DOM API, наприклад, JavaScript, Rust, Java, Kotlin та іншими. Всі провідні розробники браузерів Google (Chrome), Mozilla (Firefox), і Apple (Safari), є членами Khronos і реалізують WebGL в своїх браузерах.

Саме завдяки випуску даного API вдалося прискорити роботу з тривимірною графікою в мережі. Тепер віртуальна реальність можлива і в інтернеті завдяки новим API веб-інтерфейсів, що використовують переваги VR-пристроїв.

Завдяки цим API стало можливим відображення WebGL контенту в 3D за допомогою VR-гарнітури. Вони також надають дані відстеження гарнітури і контролера для створення ефекту присутності у віртуальному світі.

WebVR, який на разі залишається в процесі подальшої розробки, відлагодження та розширення функцій [9, 10], був розроблений в 2014 році компанією Mozilla. У 2016 році більш рання версія стандарту була доступна для десктопних браузерів Chrome, Firefox і браузера Samsung з підтримкою віртуальної реальності (Samsung Internet for Gear VR). WebVR все ще покладається на спеціальну версію браузера з підтримкою API в спеціальних налаштуваннях екрану, що більшість користувачів не зможуть легко знайти. На даний момент, WebVR API ще знаходиться в ранній стадії розробки. Бібліотека відноситься до класу API, які не можна вважати стабільним. Хоча API допрацьовується, версія 1.1 вже наявна в більшості сучасних широковикорозповсюджених браузерів для VR-платформ: Chrome для

Android, Firefox Nightly, Samsung Internet для серії смартфонів Galaxy, Microsoft Edge, Chromium, Servo і Oculus Carmel [10].

Для випадку, коли в браузер не вбудована підтримка WebVR, існує рішення в підключенні окремого файлу WebVR-поліфілу. Поліфіл (англ. polyfill) – це маленька бібліотека, зазвичай на мові JavaScript, яка реалізує підтримку певної функції, що не забезпечується стандартними засобами браузера або певної (зазвичай, старішої) версії браузера. Поліфіл для підтримки WebVR може бути знайдений за посиланням на офіційному сайті, присвяченому проекту WebVR, у розділі для розробників [9].

Слід зазначити, що підтримка WebVR не означає, що певний браузер підтримує певну гарнітуру. Але він забезпечує необхідний API, залишаючи браузерам можливість вибрати гарнітуру для підтримки [10]. Проте все ж таки сьогодні стандарт добре підтримується на телефонах і десктопних комп'ютерах більшістю основних гарнітур. Ці дані наведені у таблиці 2.1.1.

Таблиця 2.1.1. – Підтримка WebVR браузерами

	<b>Mozilla Firefox</b>	<b>Google Chrome</b>	<b>Microsoft Edge</b>	<b>Oculus Browser</b>	<b>Samsung Internet</b>	<b>Safari (iOS)</b>
<b>HTC Vive</b>	Developer edition (для розробників)	експериментальна розробка	-	-	-	-
<b>Oculus Rift</b>	Developer edition (для розробників)	експериментальна розробка	-	-	-	-
<b>Windows Mixed Reality</b>	-	-	Win 10 з Creators Update та	-	-	-

	<b>Mozilla Firefox</b>	<b>Google Chrome</b>	<b>Microsoft Edge</b>	<b>Oculus Browser</b>	<b>Samsung Internet</b>	<b>Safari (iOS)</b>
			режимом розробника			
<b>Samsung Gear VR</b>	-	-	-	підтри- мується	підтри- мується	-
<b>Google Daydream</b>	-	Chrome для Android	-	-	-	-
<b>Cardboard</b>	-	Chrome для Android	-	-	-	через полі- філ

WebVR розроблений з урахуванням наступних цілей [10]:

- Виявлення доступних пристроїв віртуальної реальності.
- Запит можливостей пристрою.
- Опитування позиції пристрою та орієнтації.
- Відображення зображення на пристрої з заданою частотою кадрів.

Крім співробітників Mozilla і Google, до роботи над версією WebVR 2.0 приєдналися також члени Microsoft.

WebVR API надає кілька нових інтерфейсів (наприклад, VR Display, VR Pose), які дозволяють веб-додаткам представити вміст у віртуальній реальності за допомогою WebGL з необхідними налаштуваннями камери і взаємодією пристроїв (такими як контролери або точка зору) [10, 11]. API був розроблений, щоб дотримуватися певної послідовності дій при розробці. Необхідні кроки [11, 12]:

- Запросити список доступних пристроїв VR.



- Перевірити, чи потрібний пристрій підтримує режими представлення, які потрібні додатку.
- Якщо це так, додаток повідомляє про можливість використання функціональності VR для користувача.
- Користувач виконує дію, яка вказує, що він хоче увійти в режим VR.
- Запит VR сесії представити VR контент.
- Почати цикл візуалізації, який виробляє графічні кадри, які повинні відображатися на пристрої VR.
- Продовжити робити кадри до тих пір, поки користувач не вкаже на те, що він хоче вийти з режиму VR.
- Завершення сесії VR.

Також API WebVR визначає багато спеціалізованих відповідно до нового інтерфейсу подій та відслідковує їх появу впродовж всього часу виконання додатку [11, 12].

В цілому, WebVR API забезпечує наступні можливості:

- Гарнітури з технологіями відстеження дозволяють користувачеві, що знаходиться у віртуальному середовищі, озиратися по сторонах. Завдяки вбудованій кадрової інтерполяції (frame interpolation) відслідковуються всі рухи голови.
- Підтримка «шести ступенів свободи» і «трьох ступенів свободи» дозволяє контролерам, як у HTC Vive і Gear VR, працювати у віртуальній реальності. Це дозволяє людині використовувати руки для взаємодії з віртуальним середовищем.
- Надається інформація про те, як повинна бути відображена 3D-інформація для гарнітури. Наприклад: видима область і як забезпечити рендеринг для кожного ока в canvas.
- Новий параметр `requestAnimationFrame` синхронізується з частотою оновлення екрану гарнітури.

- Існує метод відправки підготовлених кадрів на гарнітуру в формі елемента canvas підтримуваного WebGL.

Також важливим пунктом щодо роботи WebVR застосувань є те, що розроблений інтерфейс є «адаптивним». Так, створений додаток буде однаково коректно відпрацьовувати як на браузері настільного ПК, так і на підключених до ПК шоломах віртуальної реальності чи браузерів мобільних пристроїв всередині корпусу-окулярів VR [13].

Як було зазначено, відображення графіки у WebVR ґрунтується на використанні API-стандарту для роботи з тривимірною графікою у веб – WebGL, який створений на основі OpenGL ES (OpenGL for Embedded Systems або GLES) і дозволяє уникнути використання плагінів. WebGL, як і OpenGL, розробляється і підтримується групою Khronos Group, до якої входять фірми-розробники провідних браузерів на ринку, таких як Google Chrome, Mozilla Firefox, Apple Safari та Opera WebGL, а також світові гіганти IT-індустрії – AMD, Intel, nVidia, Samsung Electronics, Sony та багато інших. WebGL інтегровано в усі веб-стандарти для браузерів, що дозволяє прискорене використання фізичних ресурсів на GPU та обробку графіки та ефектів як частини полотна веб-сторінки [14].

Але WebVR використовує можливості WebGL лише для уніфікації доступу до графічного контенту задля його відображення, використовуючи полотно WebGL як джерело VR-контенту [11]. Власне WebVR не має інструментів для створення та обробки графіки у форматі для VR, а лише засоби для коректної трансляції VR-контенту користувачу. Саме тому WebVR зазвичай використовується у зв'язці з певною JavaScript бібліотекою, яка призначена для роботи з тривимірною графікою у веб застосуваннях. Такими бібліотеками є Three.js та Babylon.js [1, 15].

## 2.2. Бібліотеки Three.js та Babylon.js

Three.js – це фреймворк, який ґрунтується на вищеописаних стандартах WebGL. Це кросбраузерна бібліотека JavaScript, що використовується для створення та відображення анімованої комп'ютерної 3D графіки при розробці веб-додатків. Вихідний код бібліотеки розташований в репозиторії GitHub [16]. Технологія відображення графіки за допомогою Three.js базується на вищеописаному стандарті WebGL. Будучи найбільш популярною JavaScript бібліотекою для роботи з графікою у веб на базі цього стандарту, вона була також першою серед бібліотек цієї мови, яка була розширена до підтримки функцій VR.

Фактично, розробниками з Mozilla було створено два додаткових плагіни, які й забезпечують інструментом для створення VR-застосування [17].

Таким чином, для роботи Three.js з використанням можливостей WebVR необхідно, крім власне файлу бібліотеки Three.js, підключити до проекту два файли розширення:

- VRControls.js – плагін для three.js для інтерфейсу користувача в режимі VR посередництвом WebVR;
- VREffect.js – three.js плагін для самого VR ефекту.

Однак слід зауважити, що дані файли, як і більшість розширень для Three.js, не були включені в основну збірку бібліотеки, в зв'язку з чим їх можна знайти в репозиторії команди розробників з Mozilla, де розміщені шаблонні демонстраційні проекти роботи з WebVR і Three.js [17]. Також не слід забувати, що у випадку використання браузера, який не має у своїй комплектації підтримки API WebVR, слід підключити також файл поліфілу для WebVR, що було описано в попередньому пункті.

На базі даної бібліотеки було створено потужний фреймворк для розробки спеціалізованих застосунків віртуальної реальності в мережі – A-Frame, який описано далі в цій роботі.

Babylon.js – інша розповсюджена бібліотека для роботи з тривимірною графікою у веб, яка недавно отримала розширення для підтримки WebVR.

Фактично, було додано два нових типи камери [17, 18, 19]:

- VRDeviceOrientationFreeCamera – камера для реакції на зміну орієнтації пристрою;
- WebVRFreeCamera – камера для власне роботи з WebVR бібліотекою, яка реалізовує стерео рендеринг зображення та контролер камери, що є аналогом комбінації двох плагінів Three.js для WebVR.

Більш детальну інформацію можна знайти в офіційній документації до бібліотеки [18].

Babylon.js як і Three.js, також для відображення графіки використовує WebGL стандарт і, відповідно, не потребує додаткових плагінів [20]. Але вона менш розповсюджена, ніж Three.js, і в неї було пізніше додано підтримку віртуальної реальності. Але в той же час, слід сказати, що на момент написання роботи Babylon.js надає VR-засоби для розробки в межах стандартного пакету бібліотеки, тобто нові функції постачаються «вшитими», на відміну від Three.js, для якої необхідно додатково підключати файли для роботи з віртуальною реальністю.

### 2.3. Фреймворк A-Frame

A-Frame являє собою веб-фреймворк з відкритим вихідним кодом для створення веб-додатків з підтримкою віртуальної реальності [21]. Розробка в першу чергу підтримується Mozilla і спільнотою WebVR. Це entity component system фреймворк на базі Three.js, де розробники можуть створювати 3D і WebVR сцени з використанням HTML. На рисунку 2.3.1 наведено приклад створення боксу за допомогою фреймворка.

Мова розмітки HTML надає знайомий інструмент створення додатків для веб-розробників і дизайнерів, при цьому стає можливою реалізація популярного шаблону розробки ігор, який використовується ігровими движками, такими як Unity.

```
1 <a-scene>  
2 |   <a-box color="#6173F4" width="4" height="10" depth="2"></a-box>  
3 </a-scene>
```

Рис. 2.3.1 – Створення примітивного боксу за допомогою фреймворку A-Frame

Також характерною особливістю є те, що при створенні додатку на базі даного фреймворку відбувається автоматичне підвантаження бібліотеки Three.js та поліфілу WebVR для забезпечення підтримки браузером даного API [22].

Переваги роботи з фреймворком A-Frame [22]:

- 3D і VR створення контенту з HTML без кроків збірки.
- Налаштування сцени з одного рядка HTML (<a-scene>) для обробки полотна візуалізації, циклу візуалізації, освітлення, блоку управління, налаштування WebVR.

- Сумісність з більшістю існуючих веб-бібліотек і фреймворків (React.js, Angular.js, D3.js, Vue.js).
- Архітектура entity component system, яка передбачає використання композиції замість наслідування для визначення складних 3D об'єктів за допомогою багаторазово використовуваних компонентів.
- Розширювана плагінова сутність-компонент екосистема.
- Візуальний інструмент інспектора, який може бути викликаний в браузері з будь-якої сцени A-Frame.

Вся сцена-AFrame може міститися в HTML-файлі, в тому числі з одним файлом JavaScript [22].

Також слід додати декілька слів про entity component system (ECS) архітектуру по відношенню до A-Frame. Як було сказано вище, модель ECS представляє собою модель, яка широко використовується при розробці ігор, оскільки надає перевагу композиції над традиційним успадкуванням та ієрархією. Сутність (entity) є об'єктом загального призначення, який за своєю суттю нічого не робить. Компонент (component) являє собою багаторазово використовуваний модуль, який підключається до сутностей, щоб забезпечити зовнішній вигляд, поведінку і / або функціональні можливості. Вони відносяться до класу plug-and-play для об'єктів. Система забезпечує глобальний score, сервіси й управління для класів компонентів.

ECS дозволяє нам створювати складні об'єкти з багатою поведінкою, підключаючи різні компоненти багаторазового використання всюди на сутностях. Це відрізняється від традиційного спадкування, де, якщо необхідно розширити об'єкт, потрібно вручну створити новий клас, щоб зробити це. Дана модель надає розробникам ключ до інновацій, які не потребують дозволів авторів фреймворку. Розробники можуть писати, ділитися, і підключити

компоненти, які створюють нові функції або розширюються на основі існуючих функцій [22].

Для реалізації даної архітектури у A-Frame є спеціально призначений для цього HTML-тег `<a-entity>` (рис. 2.3.2). Компонентами в даному випадку виступають атрибути тегів з конкретними значеннями. Цікаво, що, оскільки ці атрибути оброблюються засобами JavaScript, то вони досить часто бувають представлені не як одиночні значення, а як об'єкти у JSON-форматі.

Не дивлячись на те, що до сцени може бути включено елементи, які визначаються різними HTML-тегами (наприклад геометричні примітиви `<a-box>`, `<a-sphere>` та інші), для розробки згідно з ECS слід притримуватися правила, коли використовується «порожня» оболонка, на яку «нарощується» функціонал.

```

1  <script src="https://rawgit.com/ngokevin/aframe-layout-component/master/dist/
    aframe-layout-component.min.js"></script>
2
3  <a-scene>
4    <a-assets>
5      
6    </a-assets>
7
8    <a-entity layout="type: circle; radius: 5" position="0 0.5 0">
9      <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
10     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
11     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
12     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
13     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
14     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
15     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
16     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
17     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
18     <a-image look-at="#player" src="#enemy-sprite" transparent="true"></a-image>
19   </a-entity>
20
21   <a-camera id="player" position="0 1.8 0"></a-camera>
22
23   <a-sky color="#252243"></a-sky>
24 </a-scene>

```

Рис. 2.3.2 – створення власного компоненту за допомогою фреймворку

## 2.4. Порівняльний аналіз засобів для створення віртуальної реальності

Порівняння інструментів здійснювалось в розрізі якісних та кількісних характеристик: результати порівняння інструментів розробника за якісними характеристиками наведено в таблиці 2.4.1.

Таблиця 2.4.1 – Результати порівняння технологій

	<b>Three.js</b>	<b>Babylon.js</b>	<b>A-Frame</b>
<b>Мова програмування</b>	JavaScript	JavaScript	HTML, JavaScript
<b>Залежності</b>	-	-	Побудований на основі Three.js
<b>API для 3D графіки</b>	WebGL 1.0	WebGL 1.0	WebGL 1.0
<b>Підтримка WebVR</b>	Плагіни до бібліотеки	Вбудована, поліфіл для WebVR	Вбудована
<b>Можливості для моделювання</b>	Ні	Ні	Так

Слід зазначити, що кожна із технологій, що порівнювались, використовувалася для розробки додатків віртуальної реальності у зв'язці з API WebVR. Також у всіх трьох інструментів наявна вбудована підтримка звуку та анімації.

У свою чергу результати порівняння технологій за кількісними характеристиками було здійснено на основі випробувань на тестових додатках, які були розроблені окремо на базі кожної технології та реалізують однакову функціональність (анімована 3D-графіка), наведено в таблиці 2.4.2.



Таблиця 2.4.2 – Результати випробування технологій

	<b>Three.js</b>	<b>Babylon.js</b>	<b>A-Frame</b>
Час відгуку, мс	648	524	1029
Об'єм ОЗП, Мб	8.6 – 11.8	11.3 – 14.7	11.6 – 14.0
Кількість рядків коду	1600	1600	600

На основі отриманих результатів можна зробити наступні висновки:

1. Фреймворк A-Frame потребує порівняно більше часу для завантаження та відпрацювання, оскільки базується на Three.js
2. Найбільш економне використання ресурсів здійснюється завдяки використанню Three.js, а Babylon.js та A-Frame потребують ОЗП приблизно в однаковій ємності
3. A-Frame є найшвидшим інструментом розробки

## **2.5. Створення контенту VR**

Для «наповнення» нашої віртуальної реальності потрібен VR-контент. Як було зазначено вище, зараз є два способи розробки VR-контенту: створення 360° відео або 3D анімації. Відображення контенту в режимі віртуальної реальності у веб забезпечується засобами API WebVR. Наприклад, якщо відкрити панорамне 360°-зображення звичайним способом в браузері з підтримкою VR, воно буде відображено пласким (рис. 2.5.1).



Рис. 2.5.1 – Відображення панорамного зображення стандартними засобами

Слід зазначити, що для відображення панорамних зображень у веб вже існують спеціалізовані інструменти. Вони надають можливість за допомогою клавіатури та миші змінювати кут огляду. Але дані застосунки пристосовані для перегляду панорамних зображеннях на звичайних пласких екранах і не відносяться до застосунків VR. Таким чином, слід мати на увазі, що відображення в браузері VR контенту у VR режимі на даний момент не можливе без API WebVR. Відповідно, будь-який контент віртуальної реальності у веб при відображенні в такому режимі виступає складовою частиною застосування віртуальної реальності.

Більш того, слід відзначити, що на даний момент існує багато інструментів для створення панорамних зображень, придатних для перегляду у форматі VR. Зокрема, існує декілька підходів до створення 360°-фотографій. По-перше, можна скористатися спеціальними застосуваннями для мобільних платформ, які дають змогу створити панорамне фото за один знімок. Прикладом такого є Panorama 360 від Google [3], для використання якого

користувач повинен натиснути кнопку початку зйомки та плавно обернутися навколо своєї осі. В результаті буде отримано фото у форматі 360°. Але слід відзначити, що даний спосіб зйомки та застосунок зокрема дуже чутливі до малих зміщень по вертикалі під час зйомки, внаслідок чого погіршується якість фотографії, оскільки видно вертикальні місця стику плоских зображень, з яких складалося панорамне.

У свою чергу, більш якісне фото можна отримати, якщо відзняти декілька звичайних зображень, кожен раз обертаючись на фіксований кут навколо себе і здійснивши таким чином повний оберт, а потім за допомогою спеціалізованих застосунків сформувати кінцеве панорамне фото. Але даний підхід також вимагає зйомки фотографій на однаковій висоті, а також близько 12 фото для отримання непомітного переходу.

Можна побачити, що описані вище підходи мають певні недоліки. І у зв'язку з поширенням VR-технологій, виробниками камер було нещодавно представлено спеціалізовані продукти для зйомки як фото, так і відео у форматі, придатному для перегляду в режимі стаціонарної віртуальної реальності. На разі, такі камери представляють собою поєднання декількох синхронізованих між собою камер, тобто, фактично, лінз. Під час зйомки кожна камера записує зображення зі свого кута огляду, а після завершення зображення/відео асемблюється в єдиний панорамний знімок. Перегляд таких відео в режимі VR вже досить довгий час можливий на Youtube. На разі, даний напрямок дуже стрімко розвивається.

Але, якщо розглядати VR-контент, створений за допомогою засобів моделювання, слід зазначити, що така функція вже наявна в багатьох сучасних пакетах 3D-моделювання. Яскравими прикладами можуть послужити такі продукти, як Blender та 3DMax. Вони надають можливість після створення певної тривимірної моделі інтер'єру перетворити її у VR-зображення за

допомогою стандартних інструментів пакету, зафіксувавши при цьому місця спостереження користувача (положення камери). Приклад 3D моделі наведений на рисунку 2.5.2.



Рис. 2.5.2 – Приклад 3D моделі дерева

## 2.6. Висновки до другого розділу

У даному розділі було розглянуто існуючі підходи до створення віртуальної реальності у веб застосуваннях, описано інструменти та способи їх використання.

На даний момент єдиним інтерфейсом прикладного програмування, який забезпечує можливість створення додатків віртуальної реальності в мережі, є бібліотека WebVR, над релізом якої працювали розробники з Mozilla, до яких пізніше приєдналися Google та Microsoft. Даний API є

експериментальним і не відноситься до стабільних, але робота над проектом триває, і найближчим часом планується випуск нової версії. WebVR підтримує власне гарнітури віртуальної реальності, а також мобільні платформи. Для відпрацювання функцій цієї бібліотеки необхідно, щоб її підтримка була «вшита» в браузер, що забезпечується лише деякими, або підключення поліфілу WebVR – бібліотеки, яка реалізує підтримку функцій WebVR. Поліфіл для підтримки WebVR може бути знайдений за посиланням на офіційному сайті, присвяченому проекту WebVR, у розділі для розробників.

Слід зазначити, що WebVR має на меті створення адаптивних застосувань, які будуть відображатися у VR-режимі на VR-пристроях, а в desktop браузерах надаватимуть можливість перегляду контенту в панорамному режимі з максимальним ефектом занурення та можливістю керування стандартними пристроями введення – клавіатурою і мишею.

WebVR виступає інтерфейсом, через який взаємодіють браузер та VR-пристрій. Він надає інструменти для веб-браузера для отримання даних від пристрою щодо позиції та орієнтації голови користувача та команд, щоб згідно з цими даними відобразити правильний контент. Для відображення графіки використовуються потужності бібліотеки WebGL, яка є де-факто стандартом для роботи з графікою у веб.

Але WebVR вирішує лише задачі коректного відображення VR-контенту користувачу та надає інтерфейс для входу-виходу з віртуальної реальності, але не має інструментів для створення та обробки графічної інформації. Тому зазвичай дана бібліотека використовується в парі зі спеціалізованою графічною JavaScript бібліотекою. Такими є Three.js та Babylon.js. Інтеграція з WebVR стає можливою, оскільки робота обох зазначених бібліотек ґрунтується на тому ж стандарті WebGL. Three.js є найбільш популярним рішенням для роботи з графікою в браузері, тому вона частіше

використовується і для VR-застосунків. Але розширення для підтримки функцій WebVR не входить в стандартну збірку бібліотеки, і потрібні файли слід підключати окремо. На противагу, розробники Babylon.js включили підтримку інструментів для розробки VR-застосувань у розповсюджуваний файл бібліотеки.

На базі Three.js розробники з Mozilla створили JavaScript фреймворк A-Frame. Цей інструмент надає можливість створювати додатки саме у форматі VR і лише за допомогою HTML-тегів. Даний фреймворк має в основі архітектуру Entity-Component System, яка часто використовується в ігрових рушіях. Для розробника створено спеціальний інспектор елементів, який надає зручний інтерфейс для маніпуляцій з об'єктами сцени. Оскільки даний фреймворк побудовано на базі Three.js, то в межах застосунку є можливість доступу до всіх елементів засобами цієї бібліотеки. Більш того, використання A-Frame можливе одночасно з більшістю сучасних JavaScript бібліотек (Angular, React.js тощо).

Також у зв'язці з WebVR для генерації власне графічного контенту можуть використовуватися рушії тривимірного моделювання. Досить популярними з даного сегменту є Unity та Unreal, але з допомогою їх засобів створюються нативні для платформи застосунки. Для того, щоб експортувати їх у веб, необхідно вихідний код, отриманий після роботи в середовищі рушію, за допомогою source-to-source компіляторів перетворити у підмножину коду JavaScript – asm.js. Застосунки на даній мові відрізняються порівняно кращою швидкістю виконання, оскільки для цього коду властива статична типізація та ручне керування пам'яттю. Але водночас даний код матиме досить великий розмір у зв'язку з ретрансляцією з однієї мови в іншу, які приблизно однакові за рівнем абстракції, а це збільшить час завантаження коду застосунку мережею. Тому чудовим рішенням може стати рушій з онлайн-інтерфейсом PlayCanvas. Будучи зручним та потужним інструментом, даний сервіс

дозволяє створювати додатки для веб і миттєво тестувати їхню роботу у браузері. Підтримка WebVR забезпечується стандартними опціями та бібліотеками і може бути реалізована будь-яким з цих двох способів.

Також було описано можливості для створення власне панорамного контенту, оскільки зображення та відео у форматі 360° можуть бути представлені в браузері лише з використанням функцій бібліотеки WebVR і, відповідно, є складовою частиною веб-застосування віртуальної реальності. Для вирішення цієї задачі можуть використовуватися як програмні засоби, коли фото вручну компонуються, так і апаратно-програмні, коли використовується спеціальна панорамна камера, яка складається з декількох об'єктивів, синхронізованих між собою, які після зйомки фото/відео здійснюють рендеринг і утворюють панорамне зображення.

Зазначено можливе використання пакетів тривимірного моделювання для створення як панорамного зображення для подальшого перегляду у VR, так і моделей віртуального світу, які в подальшому можуть бути імпортовані в додаток. Таким чином, було визначено та проаналізовано основні підходи, інструменти та можливості для створення віртуальної реальності у веб застосуваннях.

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

У процесі виконання даної роботи було створено два додатки на базі фреймворку A-Frame: один додаток з використанням підготованого панорамного контенту, другий – з генерацією контенту віртуальної реальності. Додатки були створені для визначення можливостей фреймворку, ознайомлення з особливостями та отримання навичок роботи з інструментами для створення віртуальної реальності.

Реалізований програмний продукт призначено для використання на мобільних пристроях, які мають гіроскоп та один з браузерів, що забезпечують підтримку віртуальної реальності для мобільних пристроїв, на шоломах віртуальної реальності у зв'язці з веб-браузерами. Також даний програмний продукт може використовуватися на персональних комп'ютерах в режимі сумісності за умови, що використовується браузер, який підтримує відображення віртуальної реальності на desktop-системах.

### 3.1. Створення тривимірної моделі

Для створення 3D моделі було використано пакет моделювання 3DMax. Одну із створених 3D моделей зображено на рисунку 3.1.1. Для подальшої роботи з віртуальною реальністю тривимірні моделі потрібно було експортувати з пакету та вбудувати в додаток віртуальної реальності за допомогою стандартних інструментів: JavaScript бібліотеки або фреймворку. Через зазначені вище причини для створення веб застосування було обрано фреймворк A-Frame.





Рис. 3.1.1 – Створена 3D модель

Скріншот застосування віртуальної реальності з імпортованими 3D моделями зображено на рис. 3.1.2. Ключовий момент в даному підході полягає у форматі файлу моделі, оскільки різні бібліотеки підтримують фіксований набір розширень. Але зазначена вище проблема може бути легко вирішена, оскільки зазвичай доступна достатня кількість розповсюджених розширень для збереження моделі в пакеті моделювання. А якщо потрібне розширення відсутнє, то можна скористатися проміжним пакетом моделювання, який зможе імпортувати модель у наявному розширенні та експортувати у тому розширенні, яке необхідно для застосування.



Рис. 3.1.2– Результат імпорту 3D-моделей у застосування віртуальної реальності

### **3.2. Веб застосування з підготованим панорамним контентом**

Для створення даного додатку було використано вже готовий панорамний контент. Застосування дає можливість переглядати панорамний контент у режимі віртуальної реальності на пристроях віртуальної реальності та з максимальним зануренням в повноекранному режимі на десктоп-системах. Скріншоти роботи додатку наведені на рисунках 3.2.1 та 3.2.2.

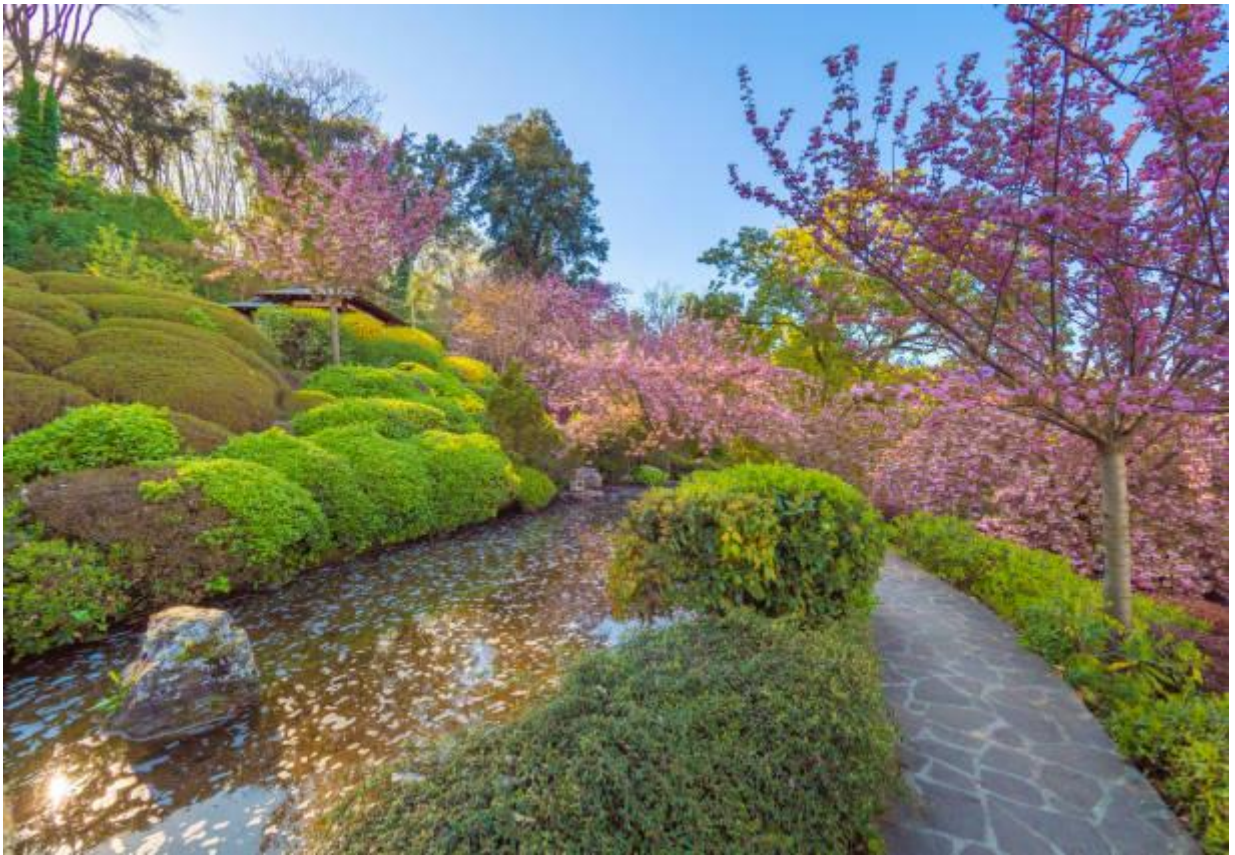


Рис. 3.2.1 - Скріншот роботи застосування у десктопному браузері



Рис. 3.2.2 – Скріншот роботи застосування у браузері мобільного пристрою

### **3.3. Веб застосування з генерацією контенту віртуальної реальності**

Для створення даного додатку було використано лише засоби фреймворку A-Frame. Усі фігури створено за допомогою стандартних примітивів. Управління відбувається залежно від середовища. Якщо додаток запускається у десктопному браузері, тоді він нагадує звичну 3D гру, управління відбувається стрілками, або клавішами W, A, S, D. Якщо ж цей додаток запущено на пристрої, який підтримує режим віртуальної реальності, то його можна відтворити у даному режимі. У цьому випадку засобами WebVR-менеджера, який автоматично викликається ресурсами Three.js, на якому побудовано фреймворк, здійснюється поділ екрану на 2 частини, а залежно від рухів голови відбувається зміна контенту, який відображається.

Як було зазначено, на даному етапі розвитку, віртуальна реальність з можливістю зчитування не лише поворотів голови, а й зміни положення тіла в просторі реалізована лише на обмеженій кількості платформ віртуальної реальності. Тому для переміщення було створено спеціалізовані «телепорти», які активується при наведенні на них курсору впродовж 1 секунди в режимі віртуальної реальності та наведенні курсору і натиснення лівої кнопки миші у десктопному режимі.

Вже відзначалося, що WebVR підтримує технологію адаптивного інтерфейсу, тобто даний додаток може бути запущений як на пристрої віртуальної реальності, так і на стаціонарному інтерфейсі комп'ютера.

Скріншоти роботи програми наведено на рисунках 3.3.1 та 3.3.2





Рис. 3.3.1 – Скріншот роботи застосування у десктопному браузері

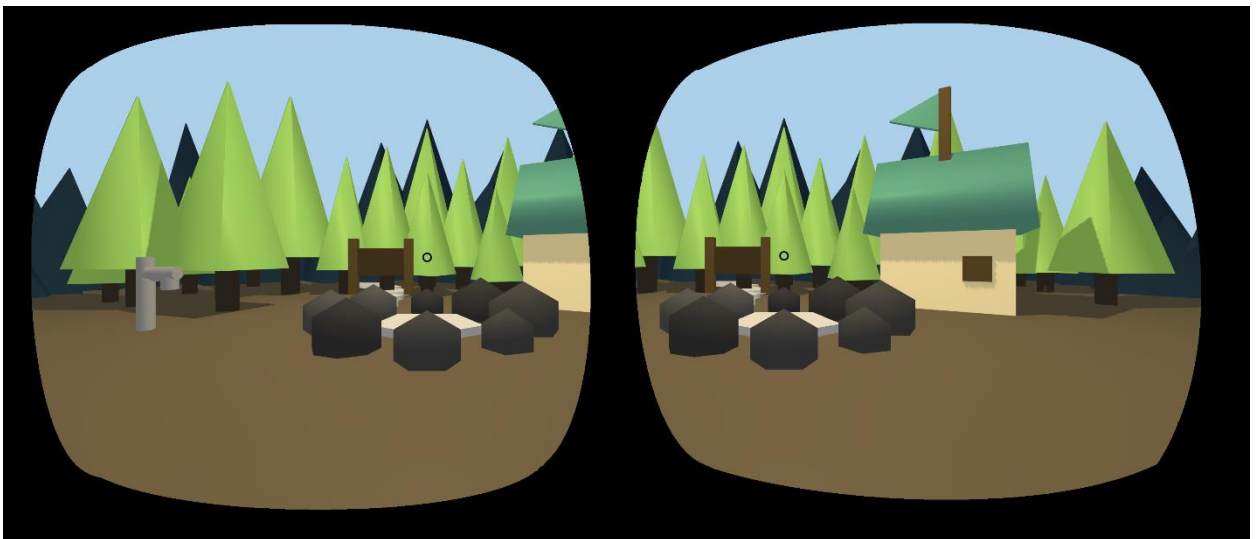


Рис. 3.3.2 – Скріншот роботи застосування у браузері мобільного пристрою

### 3.4. Особливості реалізації

Під час роботи над веб застосуваннями з підтримкою віртуальної реальності були виявлені наступні особливості:

1. Анімаційні можливості фреймворку A-Frame: початок та кінець анімації певної сутності може реагувати на події – це можуть бути

як стандартні події (натиснення кнопки миші, наведення курсору тощо), так і рамдомні події, які можуть бути випущені сутностями. Але якщо для сценаріїв «чистого» JavaScript будь-який об'єкт DOM-структури може реагувати на події від будь-якого іншого об'єкту, то в межах даного фреймворку для того, щоб анімований об'єкт помітив випущену подію, необхідно, щоб об'єкт, який її випускає, був або батьківським, або того ж рівня в межах батьківського.

2. Для даного фреймворку зображення, які використовуються в додатку (наприклад, в якості власне вставлених зображень або текстур), повинні або бути збережені локально на тому ж сервері, або, якщо вони доступні за іншою адресою, підтримувати технологію Cross-origin resource sharing (CORS), яка дозволяє сучасним браузерам надавати доступ до ресурсів домену, який відрізняється від домену сторінки.
3. У випадку використання локальних зображень, даний фреймворк коректно взаємодіятиме з ресурсами не у всіх браузерах. Так, у Mozilla Firefox не виникає помилок при роботі з локальними ресурсами, а Google Chrome не може на даному етапі працювати з такими зображеннями, і для можливої коректної роботи потребує відключення функцій браузера, які можуть призвести до порушення безпеки і некоректної роботи в цілому. Але дана ситуація є цілком зрозумілою, оскільки проект на стадії розробки, і в найближчому майбутньому це буде виправлено.

### **3.5. Висновки до третього розділу**

Даний розділ відображає розроблені на базі фреймворку A-Frame додатки з підтримкою віртуальної реальності для веб-середовища. Було надано скріншоти роботи додатків як на пристрої з підтримкою інтерфейсу віртуальної реальності, так в браузері настільного комп'ютера, де додаток

працює в режимі тривимірного представлення віртуального світу. Також наведено особливості реалізації веб застосунків з підтримкою віртуальної реальності, зроблених за допомогою фреймворка A-Frame.

## ВИСНОВКИ

Дана випускна кваліфікаційна робота присвячена розробці веб застосувань з підтримкою віртуальної реальності. У ході виконання цієї роботи було здійснено аналіз предметної галузі. Було виділено особливості, характерні для даної сфери, а також проблеми та задачі, які стоять перед фахівцями. Відповідно, було обґрунтовано актуальність та важливість поставленої мети. Наведена порівняльна характеристика інструментів для створення віртуальної реальності у веб.

Висвітлено основні підходи до розробки додатків для веб з підтримкою віртуальної реальності. Таким чином, було досліджено, що такі додатки або використовують готовий панорамний контент, або генерують контент за допомогою власних ресурсів. Але в при будь-якому з зазначених підходів для надання інтерфейсу віртуальної реальності використовується API WebVR, який на сьогодні підтримується більшістю браузерів, що працюють з віртуальною реальністю. У випадку, якщо браузер не має вбудованої підтримки, можна використати JavaScript поліфіл – невелику за обсягом бібліотеку, яка надає потрібний інтерфейс шляхом використання функцій, які підтримуються браузером. Крім цього, розробка WebVR ведеться в напрямку підтримки «VR-адаптивності», яка полягає в тому, що функціональність розробленого застосування віртуальної реальності буде коректно відпрацьовувати як на пристроях віртуальної реальності, так і на настільних системах у панорамному режимі з максимальним ефектом занурення та можливістю керування стандартними пристроями введення.

WebVR не має інструментів для створення та обробки графіки у форматі віртуальної реальності, він лише надає інтерфейс для коректної трансляції контенту віртуальної реальності користувачу. Саме тому WebVR зазвичай використовується у зв'язці з певною JavaScript бібліотекою, яка призначена для роботи з тривимірною графікою у веб застосуваннях. Такими бібліотеками є Three.js та Babylon.js, а також побудований на базі Three.js



фреймворк A-Frame. Впродовж подальшої роботи було досліджено та проаналізовано можливості цих інструментів розробника, охарактеризовано їхні особливості.

Було описано способи розробки додатків віртуальної реальності на основі JavaScript фреймворків. Оскільки відображення панорамного контенту в режимі віртуальної реальності у веб здійснюється за допомогою функцій WebVR і є програмним застосуванням, було зроблено невеликий огляд сучасних програмних та апаратних можливостей для створення фото та відео такого формату.

Було здійснено огляд можливостей для створення контенту віртуальної реальності за допомогою 3D редакторів та пакетів моделювання. По-перше, дані продукти мають опцію збереження зображення в придатному для перегляду у віртуальній реальності панорамному вигляді, а також, завдяки підтримці стандарту WebGL, надають експорт моделей з подальшою можливістю портування їх у віртуальну реальність.

Було також розроблено 2 веб додатки з підтримкою віртуальної реальності на базі фреймворку A-Frame. Впродовж цієї роботи було здійснено глибоке дослідження ресурсів технологій, визначено особливості роботи з фреймворком, оглянуто наявні напрацювання та отримано цінні практичні навички створення додатків віртуальної реальності.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 3dspace: VR for Web Developers. – Режим доступу: [3dspace.com/2016/04/vr-for-web-developers/](http://3dspace.com/2016/04/vr-for-web-developers/). – Дата доступу: 23.05.2020.
2. Hi-News.ru: Крупнейшие vr-компании сформировали глобальную ассоциацию виртуальной реальности. – Режим доступу: <https://hi-news.ru/technology/krupnejshie-vr-kompanii-sformirovali-globalnuyu-associaciyu-virtualnoj-realnosti.html>. – Дата доступу: 23.05.2020.
3. Holographica: Google опубликовала приложение для создания VR-фотографий. – Режим доступу: <http://holographica.space/news/google-cardboard-camera-android-2511>. – Дата доступу: 23.05.2020.
4. WhatIs.com: room-scale VR. – Режим доступу: <http://whatis.techtarget.com/definition/room-scale-VR-room-scale-virtual-reality>. – Дата доступу: 23.05.2020.
5. Designmodo: Virtual Reality and Web Design: Everything You Need to Know. – Режим доступу: [designmodo.com/virtual-reality-design/](http://designmodo.com/virtual-reality-design/). – Дата доступу: 23.05.2020.
6. Co.Design: Why A Virtual Reality Web May Never Happen. – Режим доступу: <https://www.fastcodesign.com/3058591/why-a-virtual-reality-web-may-never-happen>. – Дата доступу: 23.05.2020.
7. Instructables: DIY You Virtual Reality Headset - Google Cardboard. – Режим доступу: <http://www.instructables.com/id/DIY-You-Virtual-Reality-Headset-Google-Cardboard/>. – Дата доступу: 23.05.2020.
8. XinReality: Timewarp. – Режим доступу: <https://xinreality.com/wiki/Timewarp>. – Дата доступу: 23.05.2020.
9. Офіційний сайт проекту WebVR. – Режим доступу: <https://webvr.info/developers/>. – Дата доступу: 23.05.2020.

10. Офіційний опис проекту WebVR. – Режим доступу: <https://github.com/w3c/webvr/blob/master/explainer.md>. – Дата доступу: 25.05.2020.
11. Документація WebVR – Режим доступу: <https://w3c.github.io/webvr/spec/1.1/>. – Дата доступу: 25.05.2020.
12. Офіційний блог Google для розробників. – Режим доступу: <https://developers.google.com/web/fundamentals/vr/getting-started-with-webvr/>. – Дата доступу: 25.05.2020.
13. Офіційний блог Mozilla для розробників. – Режим доступу: [https://developer.mozilla.org/en-US/docs/Games/Techniques/3D\\_on\\_the\\_web/WebVR](https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/WebVR). 01.06.2017. – Дата доступу: 25.05.2020.
14. Документація до бібліотеки WebGL. – Режим доступу: [https://developer.mozilla.org/uk/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/uk/docs/Web/API/WebGL_API). – Дата доступу: 25.05.2020.
15. Sitepoint, Patrick Catanzariti: How to Build VR on the Web Today. – Режим доступу: <https://www.sitepoint.com/how-to-build-vr-on-the-web-today/>. – Дата доступу: 26.05.2020.
16. Документація до Three.js. – Режим доступу: <https://threejs.org/docs/index.html#manual/introduction/Creating-a-scene>. – Дата доступу: 26.05.2020.
17. Tony Parisi. Learning Virtual Reality / Tony Parisi // O'Reilly Media, Inc. – 2015. – First Edition. – С. 75-97.
18. Документація до Babylon.js. – Режим доступу: [http://doc.babylonjs.com/overviews/webvr\\_camera](http://doc.babylonjs.com/overviews/webvr_camera). – Дата доступу: 26.05.2020.

19. Документація до Babylon.js від Microsoft. – Режим доступу: <https://docs.microsoft.com/en-us/windows/uwp/get-started/adding-webvr-to-a-babylonjs-game>. – Дата доступу: 26.05.2020.
20. Офіційний сайт Babylon.js. – Режим доступу: <http://www.babylonjs.com/>. – Дата доступу: 26.05.2020.
21. Офіційний сайт A-Frame. – Режим доступу: <https://aframe.io/>. – Дата доступу: 28.05.2020.
22. Документація до A-Frame. – Режим доступу: <https://aframe.io/docs/0.5.0/introduction/>. – Дата доступу: 28.05.2020.
23. Документація до рушію PlayCanvas. – Режим доступу: <https://developer.playcanvas.com/en/tutorials/legacy/beginner/cardboard-vr/>. – Дата доступу: 28.05.2020.