

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ім. Ігоря Сікорського**

Навчально-науковий комплекс «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Кафедра Системного проектування
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ А.І.Петренко
(підпис) (ініціали, прізвище)

“ ___ ” _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки

6.050101 Комп'ютерні науки
(код і назва)

на тему: Реалізація віртуальної реальності у веб застосуваннях

Виконав (-ла): студент (-ка) 4 курсу, групи ДА-31
(шифр групи)

_____ Антонюк Софія Андріївна _____
(прізвище, ім'я, по батькові) (підпис)

Керівник _____ ас. Голубова І. А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант економічний _____ доцент, к. е. н. Рощина Н. В. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Нормоконтроль _____ старший викладач Бритов О.А. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2017 року

**Національний технічний університет України
«Київський політехнічний інститут»
ім. Ігоря Сікорського**

Інститут (факультет) ННК «Інститут прикладного системного аналізу»
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050101 «Комп'ютерні науки»
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

(ініціали, прізвище)

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Антонюк Софії Андріївни

(прізвище, ім'я, по батькові)

1. Тема роботи Реалізація віртуальної реальності у веб застосуваннях

керівник роботи Голубова Ірина Андріївна, ас.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «10» травня 2017 р. №1477-с

2. Термін подання студентом роботи 08.06.2017

3. Вихідні дані до роботи

Операційна система ОС Linux

Середовище розробки Sublime Text 3

Бібліотеки та фреймворки: WebVR, Three.js, Babylon.js, A-Frame

4. Зміст роботи

- 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.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

- 1) Підходи та інструменти для створення віртуальної реальності у веб застосуваннях
- 2) Результати порівняння інструментів за якісними характеристиками
- 3) Результати порівняння інструментів за кількісними характеристиками

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н. В., доцент		

7. Дата видачі завдання 01.02.2017

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.02.2017	
2	Збір інформації	15.02.2017	
3	Дослідження підходів та засобів для створення віртуальної реальності у веб-застосуваннях	28.02.2017	
4	Визначення критеріїв порівняння для різних технологій	12.03.2017	
5	Розробка архітектури застосувань	31.03.2017	
6	Написання тестових прикладів	16.04.2017	
7	Порівняння результатів дослідження	30.04.2017	
8	Оформлення дипломної роботи	31.05.2017	
9	Отримання допуску до захисту та подача роботи в ДЕК	08.06.2017	

Студент

(підпис)

С. А. Антонюк

(ініціали, прізвище)

Керівник роботи

(підпис)

І. А. Голубова

(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломної роботи.

АНОТАЦІЯ

бакалаврської дипломної роботи Антонюк Софії Андріївни
на тему: «Реалізація віртуальної реальності у веб застосуваннях»

Дана дипломна робота присвячена дослідженню підходів та інструментів для створення програмних застосувань для веб-середовища з підтримкою віртуальної реальності. Метою роботи є глибокий аналіз наявних засобів та їх порівняння, а також розробка веб додатку віртуальної реальності.

В роботі було розглянуто способи створення віртуальної реальності у веб застосуваннях, в результаті чого було визначено, що прикладним програмним інтерфейсом, який забезпечує можливість створення віртуальної реальності в мережі, є бібліотека WebVR. Також було детально описано та порівняно фреймворки, які використовуються разом з даною бібліотекою для генерації та обробки тривимірної графіки у застосунках. Було здійснено огляд можливостей рушіїв тривимірного моделювання для створення VR-застосувань для веб. Також було дано стислий огляд засобів для створення контенту віртуальної реальності. Для об'єктивного порівняння та дослідження ресурсного потенціалу технологій було розроблено додатки з підтримкою віртуальної реальності.

Дана робота може бути використана в якості методичного матеріалу.

Загальний обсяг роботи: 76 сторінок, 20 рисунків, 8 таблиць, 23 посилання.

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

АННОТАЦИЯ

бакалаврской дипломной работы Антонюк Софии Андреевны
на тему: «Реализация виртуальной реальности в веб приложениях»

Данная дипломная работа посвящена исследованию подходов и инструментов для создания программных приложений для веб-среды с поддержкой виртуальной реальности. Целью работы является глубокий анализ существующих средств и их сравнение, а также разработка веб приложения виртуальной реальности.

В работе были рассмотрены способы создания виртуальной реальности в веб приложениях, в результате чего было определено, что прикладным программным интерфейсом, который обеспечивает возможность создания виртуальной реальности в сети, является библиотека WebVR. Также были детально описаны та сравнены фреймворки, которые используются вместе с данной библиотекой для генерации и обработки трехмерной графики в приложениях. Был сделан обзор возможностей движков трехмерного моделирования для создания VR-приложений для веб. Также был дан сжатый обзор средств для создания контента виртуальной реальности. Для объективного сравнения и исследования ресурсного потенциала технологий были разработаны приложения с поддержкой виртуальной реальности.

Данная работа может быть использована в качестве методического материала.

Общий объем работы: 76 страниц, 20 рисунков, 8 таблиц, 23 ссылки.

Ключевые слова: виртуальная реальность, VR, WebVR, JavaScript, A-Frame, Three.js, Babylon.js, PlayCanvas, контент виртуальной реальности, трехмерная графика, 3D модель.

ANNOTATION

for the bachelor's work of Antoniuk Sofiia Andriivna
on "Virtual reality implementation in web applications"

This work is devoted to the research of methods and instruments for virtual reality web applications development. The goal is a deep analysis and comparison of available means as well as the development of web application with virtual reality support.

The ways of virtual reality creation in web applications were considered within this work. As a result it was defined, that the only application programming interface that enables virtual reality creation over the web is a WebVR library. The frameworks that are used in pair with this library for 3D graphics generation and manipulation were also described and compared thoroughly. There was also an overview of 3D modelling engines conducted in terms of web VR applications creating. There was also given a short inspection of means for virtual reality content mintage. To perform an objective comparison and analysis of technologies resource potential there were the applications with virtual reality support developed.

This work can be used as methodical material.

Total volume of work: 76 pages, 20 illustrations, 8 tables, 23 links.

Key words: virtual reality, VR, WebVR, JavaScript, A-Frame, Three.js, Babylon.js, PlayCanvas, virtual reality content, three-dimension graphics, 3D model.

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	13
1.1 Поняття віртуальної реальності	13
1.2 Інтерфейс взаємодії у VR.....	13
1.3 Віртуальна реальність у веб.....	14
1.4 Проблеми даної сфери та мета роботи	17
1.5 Висновки до розділу	19
2 АНАЛІЗ ЗАСОБІВ ДЛЯ СТВОРЕННЯ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ У ВЕБ ЗАСТОСУВАННЯХ	21
2.1 Підходи до створення віртуальної реальності у веб застосуваннях.....	21
2.2 WebVR – API для реалізації віртуальної реальності у веб	21
2.3 Бібліотеки та фреймворки JavaScript для генерації VR контенту в застосуванні	25
2.3.1 Бібліотека Three.js.....	25
2.3.2 Бібліотека Babylon.js.....	26
2.3.3 Фреймворк A-Frame.....	26
2.4 Рушії 3D-моделювання для розробки WebVR застосувань	31
2.4.1 PlayCanvas.....	31
2.4.2 Unity3D і Unreal.....	33
2.5 Способи створення VR-контенту	34
2.6 Висновки до розділу	38
3 ПОРІВНЯННЯ ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ У ВЕБ ЗАСТОСУВАННЯХ	41
3.1 Результати порівняння за якісними характеристиками	41
3.2 Результати порівняння за кількісними характеристиками	42
3.3 Висновки до розділу	43
4 ОПИС ПРОГРАМНОГО ПРОДУКТУ.....	44
4.1 Програмний продукт з використанням підготованого панорамного контенту	44

4.2 Програмний продукт з генерацією контенту віртуальної реальності	45
4.3 Особливості роботи з фреймворком A-Frame.....	47
4.4 Висновки до розділу	48
5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	49
5.1 Постановка задачі техніко-економічного аналізу	50
5.1.1 Обґрунтування функцій програмного продукту.....	51
5.1.2 Варіанти реалізації основних функцій.....	52
5.2 Обґрунтування системи параметрів ПП.....	54
5.2.1 Опис параметрів	54
5.2.2 Кількісна оцінка параметрів	55
5.2.3 Аналіз експертного оцінювання параметрів	57
5.3 Аналіз рівня якості варіантів реалізації функцій	62
5.4 Економічний аналіз варіантів розробки ПП	63
5.5 Вибір кращого варіанта ПП техніко-економічного рівня	69
5.6 Висновки до розділу	69
ВИСНОВКИ.....	71
ПЕРЕЛІК ПОСИЛАНЬ.....	74

ВСТУП

Сьогодні 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. Так, перша система віртуальної реальності з'явилася у 1960-х роках, а перша реалізація припадає на 1977 рік, але тогочасні обчислювальні ресурси та рівень комп'ютерної графіки не могли забезпечити належний рівень VR-контенту. Лише на 1970-ті роки припадає прихід персонального комп'ютера в життя населення, тому говорити про таку прогресивну форму взаємодії як віртуальна реальність було рано.

Необхідною передумовою набуття віртуальною реальністю масової популярності та завоювання своєї ніші на ринку є прохід через етапи розвитку та набуття статусу всюдисущості з боку інших технологій. Першою масовою платформою можна вважати РС, наступним етапом розвитку став Internet, після цього набувають масовості мобільні платформи. Таким чином, було пройдено три витки розвитку комп'ютерних технологій, і сьогодні є саме тим етапом, коли можна спостерігати прихід четвертої ери, пов'язаної з VR/AR/MR.

Чітко видно, що кожна наступна платформа використовувала надбання і ставала можливою завдяки попередній, і VR не є виключенням. Лише в результаті розвитку обчислювальних систем, комунікацій через мережу Інтернет та мобільних пристроїв, які можуть виступати платформами для роботи VR-застосувань, дана технологія має всі необхідні умови для переходу в масове користування.

Слід також відмітити психологічний фактор технологічного прогресу, коли сучасні користувачі, звиклі до наявних інтерфейсів взаємодії, жадають нових можливостей, які на даному етапі знаходять своє відбиття у сфері VR/AR/MR, а попит, відповідно, народжує пропозицію. Саме тому на галузь

віртуальної реальності покладаються великі надії з боку гігантів світової індустрії розробки апаратного та програмного забезпечення, в цю сферу інвестується велика кількість коштів. І це знаходить відгук у користувачів: так, обсяги продажів першої партії гарнітури PlayStation VR від Sony перевершили очікування самої компанії.

Саме через вищезазначені причини інвестиції та прибутки від області віртуальної реальності постійно зростають [1]. Статистика та прогнози вражають: за 2016 рік дохід від продажу VR/AR контенту та пристроїв складає, за різними даними, від 3 до 6 млрд долларів США, і передбачається, що цей показник щороку зростатиме в 2 рази, і до 2020 року дохід галузі буде в межах від 140 до 160 млрд долларів. Інвестиції у дану сферу також показують тенденцію до зростання, а основними вкладниками є такі яскраві представники IT-індустрії, як Google (платформи Cardboard, Daydream), Samsung (Gear VR), Sony (PlayStation VR), HTC (Vive), Facebook (Oculus Rift), а також багато інших [2].

Успіх цієї технології також визначається можливістю її застосування у різних секторах. Очевидно, найбільш перспективною та масивною буде частка індустрії розваг [1]. Уже зараз можна переглянути фото та відео у VR, з'являються перші фільми у форматі віртуальної реальності. Але наймасштабнішою буде розробка VR-ігор. Так, уже можна отримати досвід гри такого формату з використанням контролерів для управління ігровим процесом як в спеціалізованих клубах, так і в себе вдома. Залежно від специфіки гри, ігровий процес може навіть не потребувати таких маніпуляторів, керування виконуватиметься лише за допомогою рухів голови.

Але сфера застосування VR-технологій не обмежується лише розвагами. Цей напрямок уже широко використовується для навчання та симуляційного тренування воєнно-промисловим комплексом, для інженерного проектування, у галузі охорони здоров'я, а також у рекламі та торгівлі. Як видно, можливості застосування віртуальної реальності є дуже різноманітними та перспективними.

Як було сказано вище, 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.

Поєднання цих двох технологій обіцяє дуже швидко знайти розповсюдження серед користувачів. Так, усі описані вище сфери застосування VR дуже влучно поєднуються з можливістю обміну інформацією за допомогою мережі. Яскравим прикладом можуть стати проведення трансляцій подій у прямому ефірі у форматі VR, що дасть ефект повної присутності, мережеві відеоігри, а також інтернет-магазини. У сфері продажу та реклами уже широко використовуються застосування для здійснення перед покупкою VR-туру нерухомістю, проведення тест-драйву авто, примірювання одягу тощо.

Важливим моментом є також те, що дедалі більше зростають можливості для створення VR-контенту самими користувачами: зробити таку фото- або відео-зйомку під силу будь-кому із застосуванням спеціалізованої камери 360°, а Google уже опублікували застосування для створення VR-фотографій на платформі Android без необхідності використання спеціалізованого апаратного забезпечення [3].

Отож, хоча віртуальна реальність стрімко набирає популярність, але ще досі знаходиться на початковому етапі свого розвитку. Більшість існуючих технологій для розробки програмного забезпечення з підтримкою VR відрізняються своєю новизною. Майже всі бібліотеки та напрацювання цієї сфери знаходяться на стадії розробки, нарощення функціоналу, подальшого розвитку та вдосконалення. Щодо вичерпуючих методичних та навчальних матеріалів, то їх, відповідно, досить мало, оскільки це незвідана область, яка, не дивлячись на те, що широко підтримується вищезазначеними гігантами IT-індустрії, ще не має достатньо широкорозповсюджених продуктів, і тому лише

невелика кількість розробників не лякається присвятити свій час ще не випробуваній технології і працює над проектами веб-застосувань з підтримкою VR. Але, як було сказано вище, дана технологія за всіма прогнозами та передбаченнями, найближчим часом стане провідним вектором розвитку інформаційних систем усіх галузей застосування. І тому сьогодні є саме тим моментом, під час якого досвід вивчення даної технології неодмінно стане конкурентною перевагою в майбутньому, а підтримка її подальшого розвитку забезпечить швидке й стале закріплення її продуктів у повсякденному використанні.

Описані вище причини яскраво ілюструють велику необхідність в дослідженні наявних інструментів, їх аналізі та порівнянні. Саме ця мета ставиться в рамках дипломного проекту. Це дасть змогу створити необхідний базис теоретико-практичного матеріалу, який одночасно з прикладним використанням може стати поштовхом для розробки нових концепцій, підходів, алгоритмів та технологій для реалізації віртуальної реальності у веб-застосуваннях.

Для більш глибокого аналізу з точки зору практичного застосування другою частиною роботи є набуття досвіду створення VR-застосувань для веб за допомогою наявного інструментарію. Це дасть змогу порівняти зручність та легкість використання досліджуваних інструментів, об'єм реалізованого функціоналу та доступні можливості, а також якість кінцевого продукту, і, відповідно, допоможе визначити, які частини архітектури і/або реалізації потребують покращення та в якому напрямку слід рухатися далі. Даний застосування зможе знайти практичне застосування у своїй галузі в подальшому, таким чином розповсюджуючи та розвиваючи VR-технологію.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття віртуальної реальності

Віртуальна реальність – сучасна сфера інформаційних технологій, яка передбачає інструменти й засоби для створення штучного комп'ютерного інформаційного середовища, інтерактивного по відношенню до користувача. Поняття віртуальної реальності, або VR, об'єднує в собі апаратне та програмне забезпечення, необхідне для створення поглинаючого інтерфейсу взаємодії користувача. Метою даної технології є створення такого досвіду, який викликає в юзера відчуття присутності в штучному світі, реальності середовища та максимальної наближеності до реального світу, а також реалізація можливості штучної віртуальної реальності реагувати на дії користувача (responsibility). В результаті виконання поставлених перед VR задач буде створено такий інтерфейс взаємодії, який одночасно з властивою йому реалістичністю буде максимально зручним та інтуїтивно зрозумілим, що надаватиме йому перевагу при виборі юзерами технології взаємодії.

1.2 Інтерфейс взаємодії у VR

Основою інтерфейсу сучасних застосувань віртуальної реальності є саме графічний контент, який змінюється у відповідь на зміну положення голови користувача. В деяких випадках даний інтерфейс доповнюється звуковою інформацією. На разі немає стандартних вбудованих широкорозповсюджених технологій для створення тактильних відчуттів, запахів тощо, хоча в даному напрямку проводяться розробки. Але найголовнішим напрямком, в якому здійснюється робота над VR-інтерфейсом, 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 VR додатком, камер та/або інфрачервоних датчиків, які фіксують

переміщення користувача, передають зображення обчислювальному вузлу, який визначає зміну положення тіла та передає результуючі дані додатку VR, що в сукупності забезпечує реакцію середовища VR 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 VR-застосувань розроблено для стаціонарної парадигми VR, де фіксуються зміни лише положення голови.

Взаємодія користувача з VR-середовищем відбувається за допомогою VR-контролерів, а для випадку, якщо їх немає, було де-факто 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 VR-середовища.

1.3 Віртуальна реальність у веб

З точки зору веб-технологій, віртуальна реальність – це один зі способів взаємодії користувача з мультимедійною інформацією, а також іншими людьми в режимі реального часу з використанням мережевих технологій. Цей спосіб цілком незвичний для Internet, 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 3D-контент, але лише як демонстраційний матеріал для чітко визначених та обмежених цілей, наприклад, об'ємна демонстрація товару в інтернет-магазині.

Віртуальна реальність же, на відміну від звичного способу керування, 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 VR за допомогою спеціальних маніпуляторів. І шолом, і маніпулятори містять датчики руху, тому можуть ефективно реалізовувати інтуїтивно-зрозумілий інтерфейс та більш швидку та гнучку роботу.

Слід також розрізняти віртуальну реальність та звичайні пласкі об'єкти інтерфейсу користувача, які приведені до псевдотривимірного 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 рис. 1.1 продемонстровано приклад віртуальної реальності, а на рис. 1.2 – зразок приведеного до тривимірного вигляду інтерфейсу VR-браузеру Samsung Internet for Gear VR.

На разі, 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.



Рис. 1.1 – приклад віртуальної реальності

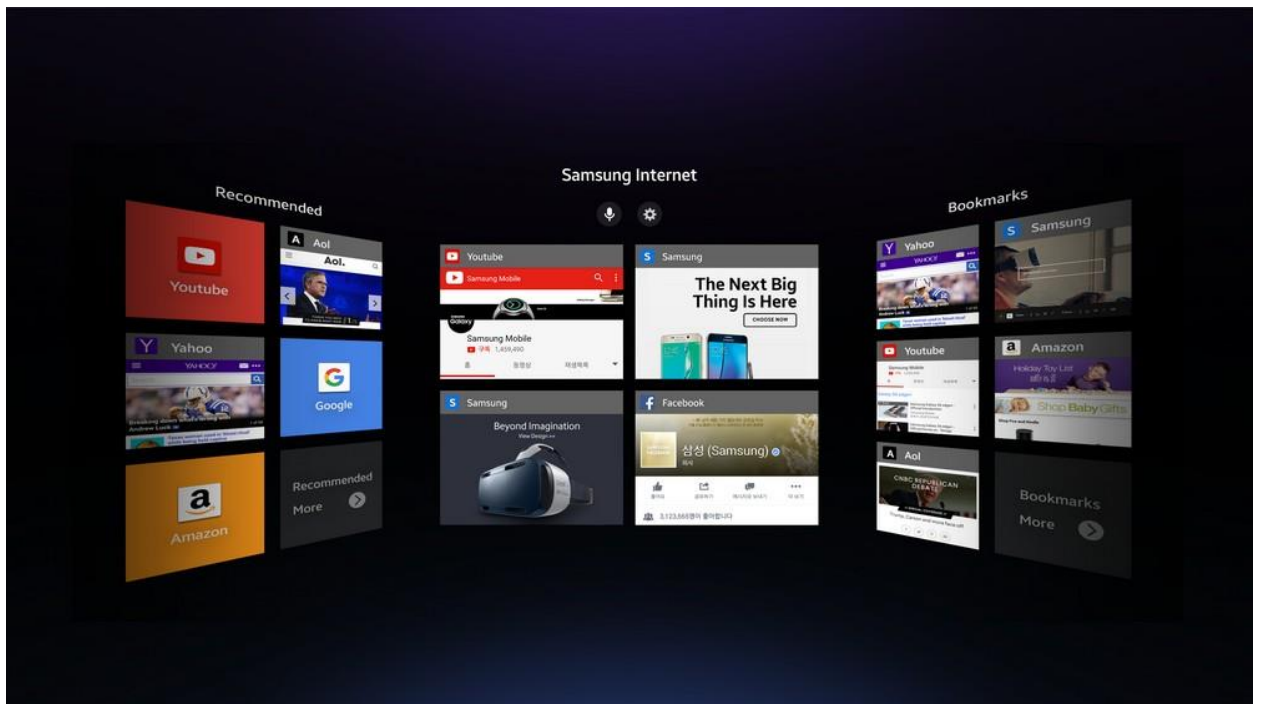


Рис. 1.2 – Інтерфейс VR-браузеру Samsung Internet for Gear VR

1.4 Проблеми даної сфери та мета роботи

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. Розглянемо їх.

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, ігрова консоль тощо.

На відміну від нього, 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.

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 – лінз та картону [7].

Але для обох даних підходів до створення апаратного забезпечення 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].

Але для досягнення потрібного результату слід імплементувати 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, і з'явиться можливість створювати програмні продукти з покращеними характеристиками. Саме аналіз та порівняння інструментів розробки є складовою частиною мети даної роботи.

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 VR-контенту, 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 розділу

Необхідність у дослідженні та розвитку даної технології є дуже великою. Так, в 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. 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.

Крім 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 VR, а й новий спосіб пошуку інформації, обміну повідомленнями, ведення бізнесу, і це все в режимі поглинаючого VR.

Таким чином, 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 присуща 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.

Отже, 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.

2 АНАЛІЗ ЗАСОБІВ ДЛЯ СТВОРЕННЯ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ У ВЕБ ЗАСТОСУВАННЯХ

2.1 Підходи до створення віртуальної реальності у веб застосуваннях

На сучасному етапі розвитку VR-технологій 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 додатку за допомогою потужностей інструментів розробника.

Цей 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 взаємодіятиме 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 VR-середовища під час роботи додатку, досить виразно відрізняється від описаного вище, оскільки передбачає використання додаткових спеціалізованих інструментів для реалізації даної задачі та, відповідно, передбачає абсолютно інакшу архітектуру застосунку.

2.2 WebVR – API для реалізації віртуальної реальності у веб

Отож, як було зазначено вище, для реалізації віртуальної реальності в мережі при будь-якому підході необхідне використання API, яке забезпечить коректну взаємодію додатку з пристроєм. На разі єдиним реалізованим робочим

продуктом є експериментальний API WebVR, який на разі залишається в процесі подальшої розробки, відлагодження та розширення функцій [9, 10].

Даний API забезпечує підтримку таких пристроїв віртуальної реальності, як HTC Vive, Oculus Rift та Google Cardboard у веб-браузері. Цей API 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.

Проект WebVR був анонсований навесні 2014 року Володимиром Вукічевіч з Mozilla. Над створенням цього API також працювали Брендон Джонс, Борис Смас й інші розробники з команди Mozilla. 1 березня 2016 року, команда Mozilla VR і команда Google Chrome оголосила про випуск WebVR API версії 1.0. В результаті рефакторингу було здійснено багато удосконалень WebVR.

Робота над даною бібліотекою продовжується, найновіший реліз версії 1.1 припадає на 5 квітня 2017. 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].

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 широквикорозповсюджених браузерах 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].

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 Kronos 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'язі з певною JavaScript бібліотекою, яка призначена для роботи з тривимірною графікою у веб застосуваннях. Такими бібліотеками є Three.js та Babylon.js [1, 15].

2.3 Бібліотеки та фреймворки JavaScript для генерації VR контенту в застосуванні

2.3.1 Бібліотека Three.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.

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.

2.3.2 Бібліотека Babylon.js

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].

Як і Three.js, Babylon.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 VR.

2.3.3 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. 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.

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 (рис. 2.1).

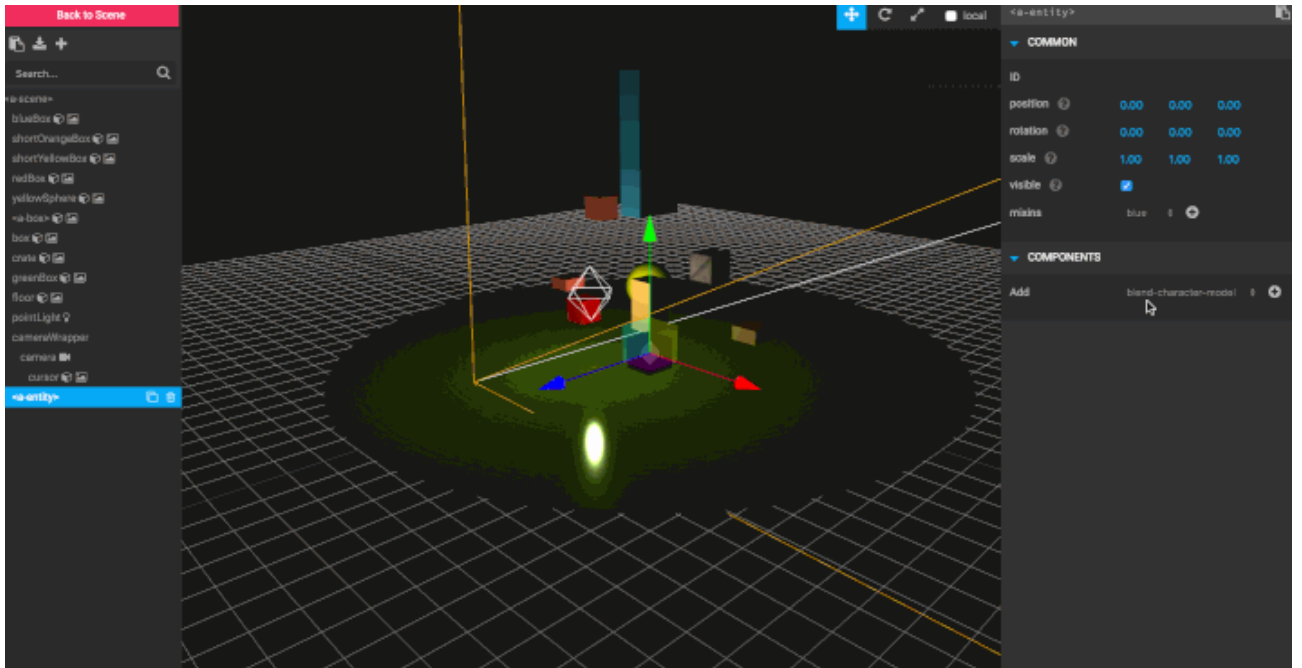


Рис. 2.1. Інспектор A-Frame.

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

Так, за допомогою лічених рядків коду можна створити просту VR-сцену. На рисунку 2.2 наведено приклад коду, а на рис. 2.3 – зображення роботи відповідного VR-додатку.

```

1 <html>
2   <head>
3     <title>Hello, World!</title>
4     <script src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
5   </head>
6   <body>
7     <a-scene>
8       <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
9       <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
10      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
11      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
12      <a-sky color="#ECECEC"></a-sky>
13    </a-scene>
14  </body>
15 </html>

```

Рис. 2.2 – Код простого VR-додатку на A-Frame

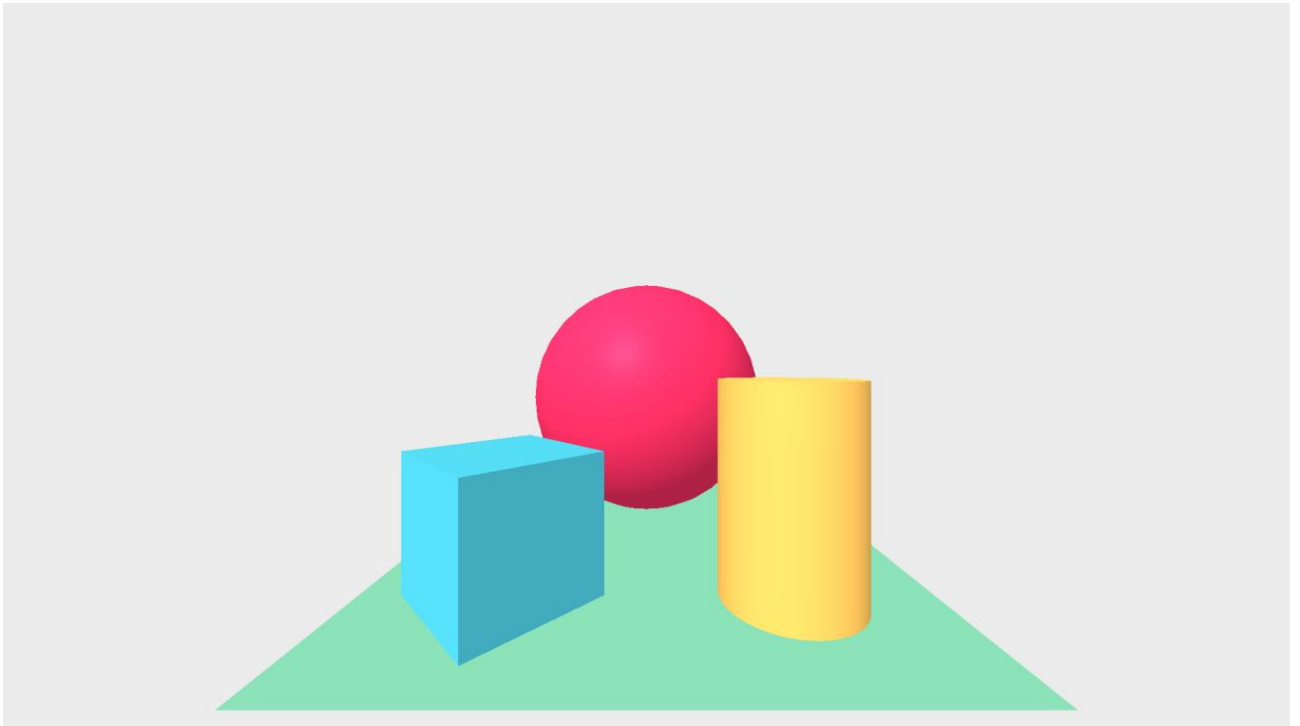


Рис. 2.3 – Скріншот роботи додатку.

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 scope, сервіси й управління для класів компонентів.

ECS дозволяє нам створювати складні об'єкти з багатою поведінкою, підключаючи різні компоненти багаторазового використання всюкети на сутностях. Це відрізняється від традиційного спадкування, де, якщо необхідно розширити об'єкт, потрібно вручну створити новий клас, щоб зробити це.

ECS 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].

У A-Frame для реалізації даної архітектури є спеціально призначений для цього HTML-тег `<a-entity>`. 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. Прикладом може служити код на рис. 2.4.

```
<a-entity geometry="primitive: sphere; radius: 1.5"  
          material="color: tomato; metalness: 0.7"></a-entity>
```

Рис. 2.4 – Entity-component system у A-Frame

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. Так, приклад на рис. 10 цілком відповідає архітектурі ECS, але міг би бути замінений на код на рис. 2.5 без зміни у продуктивності та функціональності.

```
<a-sphere radius="1.5"  
          color="tomato"  
          metalness="0.7"></a-sphere>
```

Рис. 2.5 – Альтернативний лістинг

2.4 Рушії 3D-моделювання для розробки WebVR застосувань

2.4.1 PlayCanvas

PlayCanvas – це рушії тривимірного моделювання з відкритим вихідним кодом, який працює в онлайн режимі та надає інтерфейс для розробки додатків у межах вікна браузера [17, 23]. Створені додатки можна тестувати відразу у вікні браузера, що є доступним за допомогою одного натиснення кнопки миші, а також зберігати створені проекти в хмарних сховищах.

Будучи призначеним для створення проектів тривимірної графіки у web на базі WebGL, можливості PlayCanvas було розширено до підтримки інструментів для створення застосунків у VR на базі WebVR. На разі, є декілька способів для створення додатків з підтримкою віртуальної реальності в PlayCanvas [23].

Перший спосіб призначений для створення VR-додатків під мобільні пристрої. Для цього розробники PlayCanvas виклали у github-репозиторій два бібліотечних файли, які дозволяють VR розробку: `input_hmd.js` та `vr_camera.js`. Перший файл слід додати до списку «Script Priorities» через однойменне меню, оскільки це бібліотечний файл, який повинен бути виконаний перед початком роботи всього додатку [pc1]. Цей файл відповідає за підтримку обробки вхідної інформації від гарнітури VR. Файл `vr_camera.js` повинен бути доданий до сутності (entity), яка виконуватиме роль камери. При входженні у режим VR даний скрипт забезпечить створення пари камер для двох очей.

Інший спосіб полягає в наступній послідовності дій: спочатку в середовищі розробки в меню налаштувань рендерингу слід поставити прапорець «Enable VR», оскільки за замовчанням PlayCanvas не ініціалізує VrManager та не завантажує поліфіл WebVR з причин оптимізації. Після цього розробник може додавати скрипти для визначення роботи складової VR частини додатку. Обов'язково слід прописати скрипти входу та виходу з VR-режиму для камери. Для більш детальної інформації можна скористатися документацією до середовища розробки PlayCanvas або скористатися офіційними уроками на сайті.

PlayCanvas також надає можливість ознайомлення з шаблонним VR проектом для більш швидкого інтуїтивного навчання. Ця опція доступна при створенні нового проекту – слід обрати «VR Starter Kit».

Даний пакет розробки відрізняється своєю зручністю та легкістю освоєння, оскільки має інтуїтивно зрозумілий інтерфейс користувача, а також спеціальний ввідний туторіал, який дуже швидко ознайомлює розробника з інтерфейсом даного пакету за допомогою поступових підказок на базі спеціально підготованого тестового прикладу.

Даний рушій, як і більшість ігрових рушіїв, базується на вищеописаній архітектурі Entity-Component system.

На рисунку 2.6 зображено інтерфейс рушію PlayCanvas.

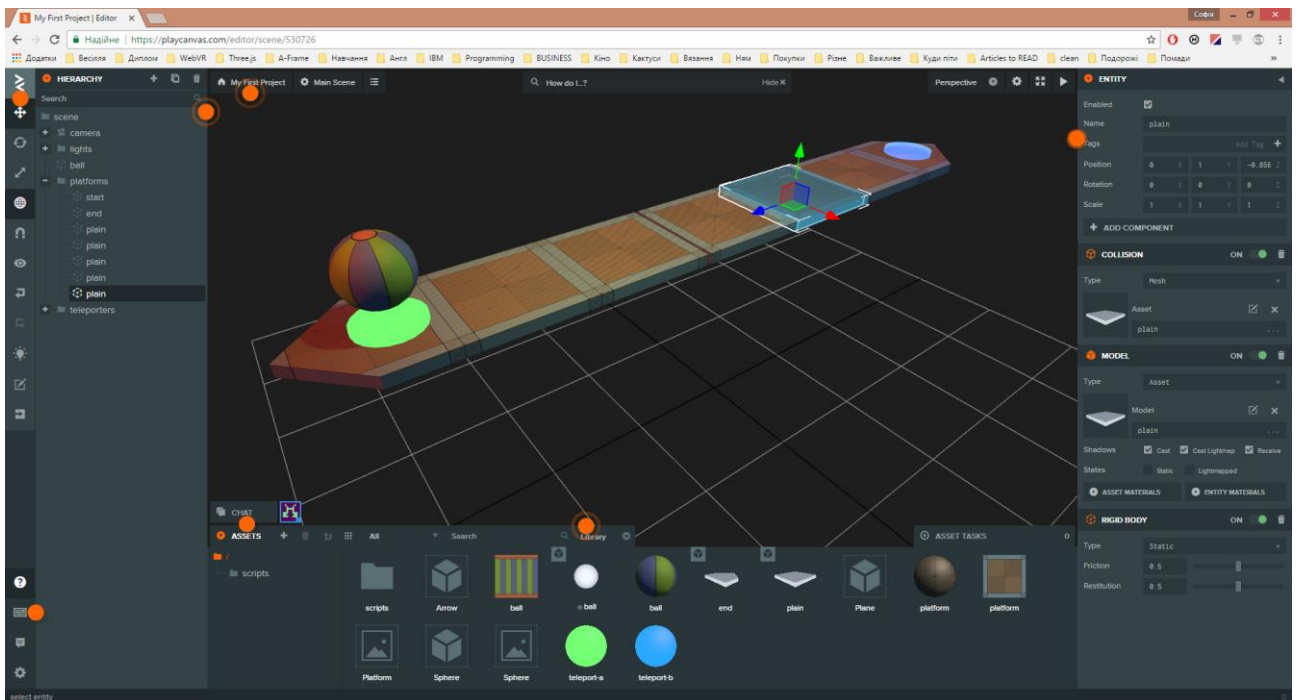


Рис. 2.6 – Інтерфейс середовища створення додатків PlayCanvas

2.4.2 Unity3D і Unreal

Два розповсюджених рушії для розробки тривимірних проектів та ігор для настільних систем Unity3D і Unreal також додали підтримку віртуальної реальності, але лише для розробки «нативних» для настільної або мобільної платформи застосунків. Але у зв'язку з тим, що до функцій обох зазначених рушіїв було додано експорт у формат WebGL (до речі, безкоштовний, доки застосування, що розробляється, не буде використовуватись для прибутку), створення VR-проектів для веб також стає можливим [17].

Покриття нового користувацького сегменту є перевагою для розробників, які вже давно працюють з даними рушіями. Але тут є одна особливість: екпорти ґрунтуються на використанні source-to-source компілятора Emscripten, яке полягає у крос-компіляції результуючого вихідного коду, отриманого з рушію, до низькорівневого коду підмножини JavaScript, який зветься asm.js. Кінцевий код має переваги, оскільки, являючись валідним JavaScript кодом, використовує статичну типізацію та ручне управління пам'яттю (як у мові C). В результаті буде отримано застосування з дуже високою частотою відображення кадрів (frame rate), що забезпечить покращення якості зображення. Але недоліком може стати те, що при source-to-source компіляції результуючий код застосування може мати об'єм в декілька мегабайтів, і цей код повинен бути завантажений в браузер, після чого почне запуск власне застосування. Відповідно, може виникнути достатньо великий час очікування для користувача. Але дана проблема може досить сильно залежати від конкретного екземпляру застосування та не мати загальної критичності.

На рисунку 2.7 зображено інтерфейс роботи над VR проектом в Unity.

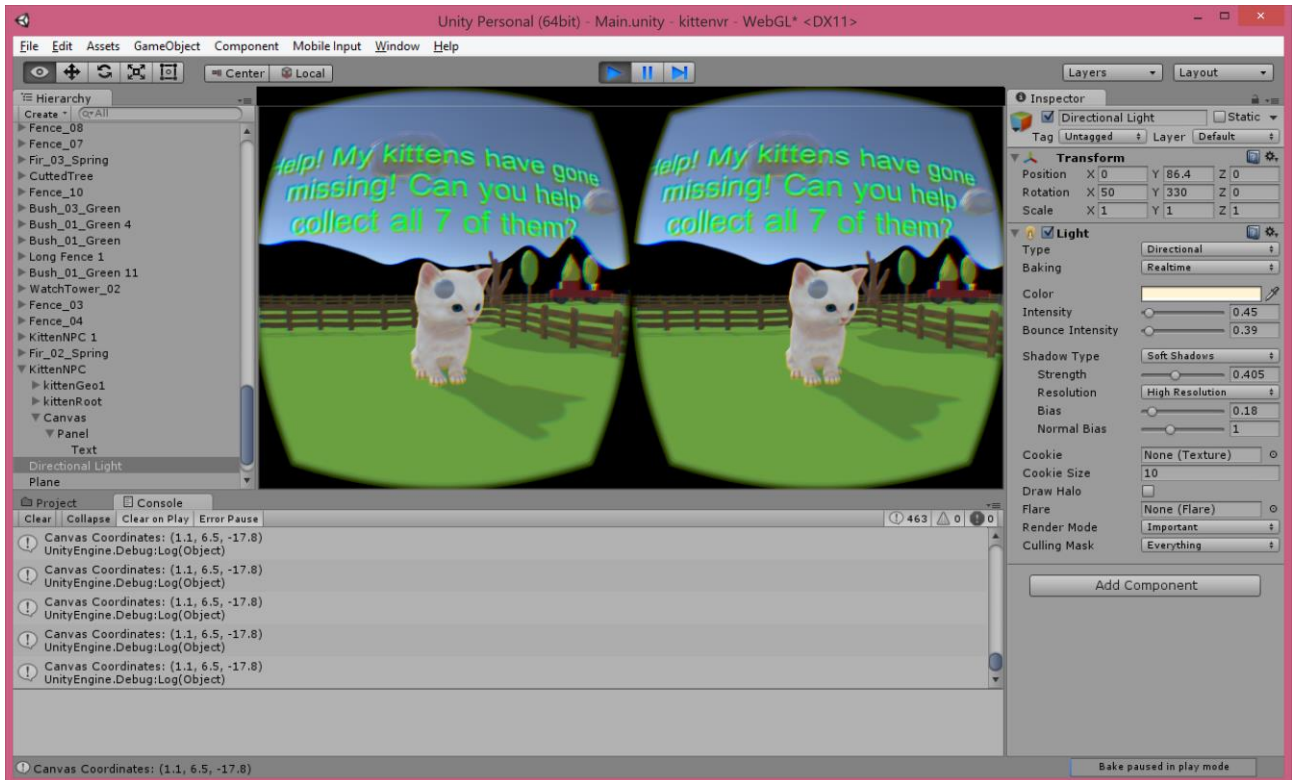


Рис. 2.7 – Приклад роботи над VR-проектом в Unity

2.5 Способи створення VR-контенту

Незважаючи на те, що створення VR-контенту не є темою дослідження даної роботи, цей процес пов'язаний з реалізацією віртуальної реальності у веб застосуваннях, тому слід виокремити наступні моменти.

Як було зазначено вище, відображення контенту в режимі VR у веб забезпечується засобами API WebVR. Наприклад, якщо відкрити панорамне 360°-зображення звичайним способом в браузері з підтримкою VR, воно буде відображено пласким (рис. 2.8).



Рис. 2.8 – Приклад відображення панорамного зображення стандартними засобами

Водночас 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 плоских 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 VR у веб при відображенні в такому режимі виступає складовою частиною застосування віртуальної реальності.

Перш за все, 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 юзер 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.

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 16 фото для отримання непомітного переходу.

Очевидно, 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.

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 положення камери, тобто місця спостереження користувача.

Іншим способом використання пакетів 3D моделювання для роботи з VR є створення тривимірних моделей, які в подальшому можуть бути експортовані з пакету та вбудовані в додаток VR за допомогою стандартних інструментів JavaScript бібліотеки або фреймворку. Приклад роботи VR-застосування з імпортованою моделлю зображено на рис. 2.9. 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. 2.9 – Результат імпорту 3D-моделі у VR застосування

2.6 Висновки до розділу

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

Було зазначено, що 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 відображати 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-застосунків 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.

Також 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.

3 ПОРІВНЯННЯ ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ У ВЕБ ЗАСТОСУВАННЯХ

В попередній частині даної роботи було описано головні підходи та засоби, за допомогою яких можна створити додатки VR для веб середовища, які наразі існують. Оскільки способи створення саме контенту віртуальної реальності не є предметом дослідження даної роботи, аналіз здійснювався саме по відношенню до інструментів розробника, а саме бібліотек та фреймворків.

Порівняння інструментів здійснювалось в розрізі якісних та кількісних характеристик

3.1 Результати порівняння за якісними характеристиками

Результати порівняння інструментів розробника за якісними характеристиками наведено в таблиці 3.1.

Кожна із технологій, що порівнювались, використовувалася для розробки додатків віртуальної реальності у зв'язці з API WebVR.

Таблиця 3.1 – Результати порівняння технологій

Критерії\Технології	A-Frame	Three.js	Babylon.js
Мова програмування	HTML, JavaScript	JavaScript	JavaScript
Залежність від інших технологій	Побудований на основі Three.js	-	-
API для рендерингу 3D графіки	Вбудована, WebGL 1.0	Вбудована, WebGL 1.0	Вбудована, WebGL 1.0
Підтримка WebVR	Вбудована	Плагіни до бібліотеки	Вбудована, в розробці; робота з поліфілом для WebVR
Можливості для моделювання	Так (інспектор)	Ні	Ні

Також у всіх трьох інструментів наявна вбудована підтримка звуку та анімації.

3.2 Результати порівняння за кількісними характеристиками

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 3.2.

Таблиця 4.2 – Результати випробування технологій

Критерії	A-Frame	Three.js	Babylon.js
Час відгуку, мс	1170	731	637
Об'єм ОЗП, Мб	12.3 – 13.2	9.1 – 10.6	12.1 – 13.5
Кількість рядків коду, сотні	0.3	0.8	0.8

На основі отриманих результатів можна зробити наступні висновки:

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

3.3 Висновки до розділу

В даному розділі було представлено зведений результат порівняння технологій та інструментів для розробки додатків віртуальної реальності, які надають можливості для створення VR-контенту.

Технологіями, що порівнювалися, є JavaScript фреймворки: A-Frame, Three.js та Babylon.js.

Було зазначено, що для всіх інструментів спільним є використання мови JavaScript для розробки додатків, використання вбудованого API для рендерингу 3D графіки WebGL версії 1.0, підтримка WebVR (з певними відмінностями, описаними детально в розділі 3), а також вбудована підтримка анімації та звуку.

A-Frame є єдиною технологією, що залежить від іншої (Three.js), але також лише даний фреймворк надає зручний інструмент інспектора з можливістю моделювання через графічний інтерфейс.

В результаті практичних тестів було визначено, що найшвидше відпрацьовують додатки на Babylon.js, з незначним відставанням – Three.js, а додатки на A-Frame мають суттєвішу затримку. Але така затримка не є некомфортною при роботі з готовим продуктом навіть при більшій кількості ресурсозатратних елементів у додатку. Об'єм ОЗП приблизно однаковий у Babylon.js та A-Frame, а Three.js дає незначно, але кращі результати. Щодо кількості коду в результуючому додатку, то слід сказати, що A-Frame має суттєву перевагу. Таким чином, фреймворк A-Frame є технологією, яка потребує найменше людських і/або часових ресурсів для розробки еквівалентного додатку.

4 ОПИС ПРОГРАМНОГО ПРОДУКТУ

Протягом роботи над даним дипломним проектом було створено два додатки на базі фреймворку A-Frame – з використанням підготованого панорамного контенту та генерацією контенту VR.

Дані застосунки були створені для визначення можливостей фреймворку, ознайомлення з особливостями та отримання навичок роботи з даним інструментом.

4.1 Програмний продукт з використанням підготованого панорамного контенту

Розроблений застосунок дає можливість переглядати панорамний контент у режимі віртуальної реальності на пристроях VR та з максимальним зануренням в повноекранному режимі на десктоп-системах.

Скріншоти роботи додатку наведені на рисунках 4.1 та 4.2.



Рис. 4.1 - Скріншот роботи застосування в десктоп-браузері



Рис. 4.2 – Скріншот роботи застосування в браузері мобільного пристрою

4.2 Програмний продукт з генерацією контенту віртуальної реальності

Даний додаток надає можливість прогулянки серед дерев у віртуальній реальності, створеній засобами лише фреймворку.

Так, усі фігури створено за допомогою стандартних примітивів. Управління відбувається залежно від середовища. Якщо додаток виконується у десктоп-браузері, він схожий на звичну 3D гру, управління відбувається стрілками (так званий WASD-контроль). 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 VR-платформ. Тому для переміщення було створено спеціалізовані «телепорти», які активується при наведенні на них курсору впродовж 1 секунди в режимі ВР та наведенні курсору і натиснення лівої кнопки миші в режимі ПК.

Як було зазначено, WebVR підтримує технологію адаптивного інтерфейсу, тобто даний додаток може бути запущений як на пристрої ВР, так і на стаціонарному інтерфейсі ПК.

Скріншоти роботи програми наведено на рисунках 4.3 та 4.4.



Рис. 4.3 – Скріншот роботи застосування в десктоп-браузері

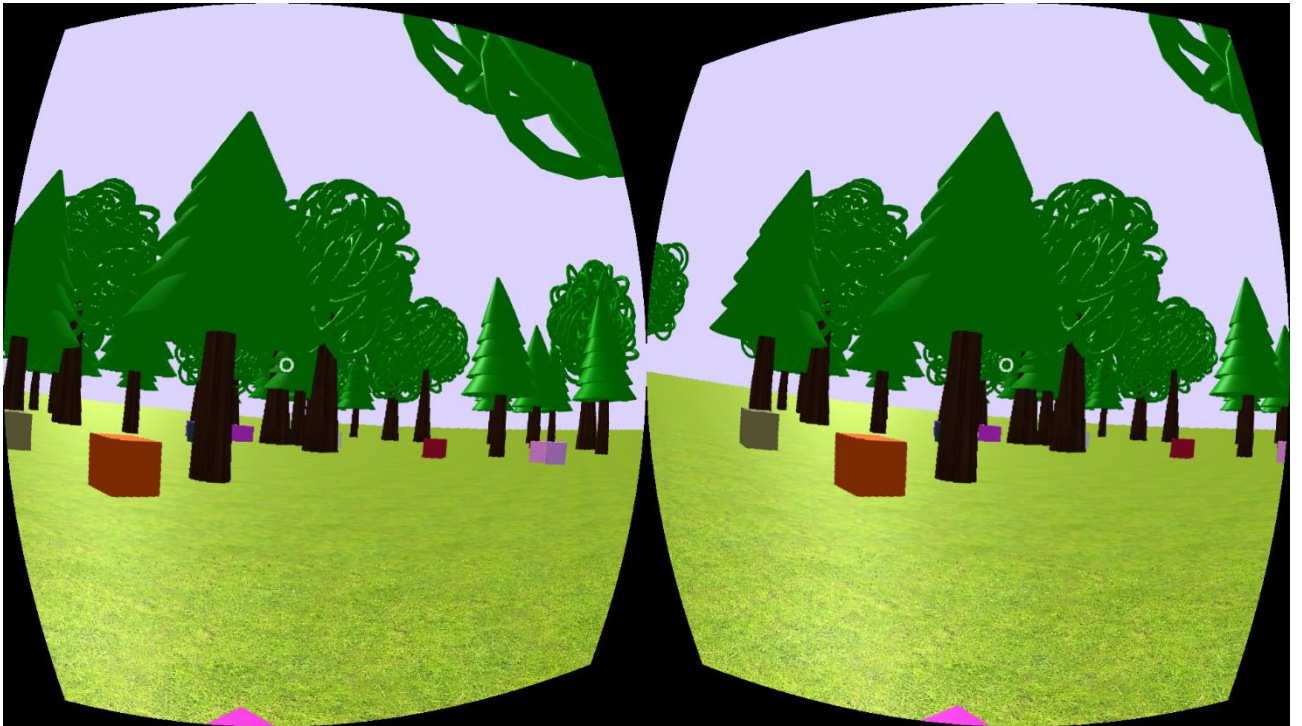


Рис. 4.4 – Скріншот роботи застосування в браузері мобільного пристрою

4.3 Особливості роботи з фреймворком A-Frame

Говорячи про особливості роботи з фреймворком A-Frame, слід відмітити наступні фактори. Перш за все, це стосується анімації можливостями фреймворку. Так, початок та кінець анімації певної сутності може реагувати на події – це можуть бути як стандартні події (натиснення кнопки миші, наведення курсору тощо), так і кастомні 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.

Також при роботі над додатком було відмічено, що 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.

Також слід зазначити, що 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.

4.4 Висновки до розділу

В даному розділі було описано розроблені на базу фреймворку A-Frame додатки з підтримкою віртуальної реальності для веб-середовища. Також було надано скріншоти роботи додатків як на пристрої з підтримкою VR інтерфейсу, так в браузері настільного ПК, де додаток працює в режимі тривимірного представлення віртуального світу.

Також було надано опис особливостей роботи з фреймворком A-Frame. Дані відмітки можуть стати методичним матеріалом для майбутніх веб-розробників, які працюватимуть з цим фреймворком.

5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, який реалізовує віртуальну реальність у веб-середовищі. Даний програмний продукт буде створено з використанням потужностей бібліотеки WebVR.

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.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

- визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі оптимізується сама послідовність скороченням кроків, що не впливають на цінність, і, відповідно, витрат.
- для кожної функції визначаються повні річні витрати й кількість робочих годин.
- для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.
- після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

5.1 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи віртуальної реальності у веб-середовищі. Оскільки основні проектні рішення стосуються всієї системи, кожна окрема підсистема має їм задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу гетероскедастичних процесів в економіці та фінансах.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на мобільних пристроях з вбудованим гіроскопом та встановленими веб-браузерами, що підтримують віртуальну реальність, на гарнітурах

віртуальної реальності зі встановленими веб-браузерами, в режимі сумісності – на персональних комп'ютерах зі встановленими веб-браузерами, які підтримують віртуальну реальність для desktop-систем;

- забезпечувати високу швидкість обробки графічних даних у реальному часі;
- забезпечувати зручність і простоту взаємодії з користувачем або з розробником програмного забезпечення у випадку використання його як модуля;
- передбачати мінімальні витрати на впровадження програмного продукту.

5.1.1 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка веб-застосування, яке підтримує віртуальну реальність. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір бібліотек програмування;

F_2 – вибір середовища розробки;

F_3 – вибір способу збереження та тестування ПП.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

- а) фреймворк A-Frame;
- б) зв'язка бібліотек Three.js + WebVR;

Функція F_2 :

- а) IDE SublimeText;
- б) IDE WebStrom.

Функція F_3 :

- а) збереження виконуваних файлів ПП на виділеному сервері з режимом доступу;
- б) збереження виконуваних файлів ПП локально на кожному пристрої для тестування.

5.1.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 5.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 5.1).

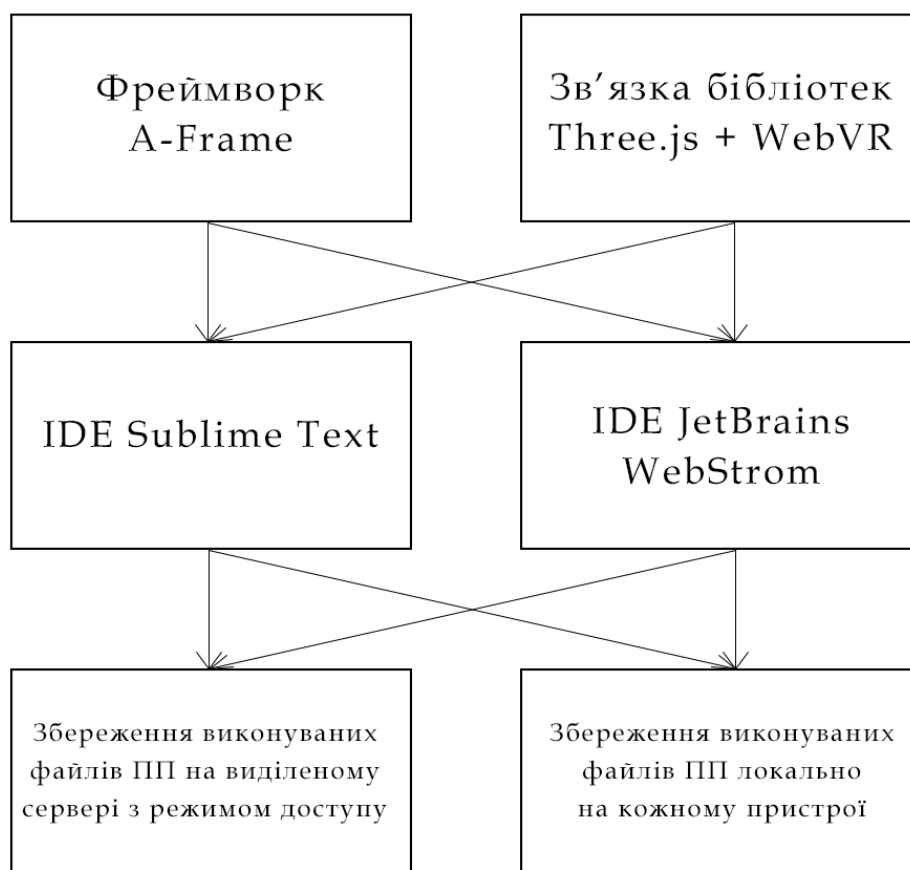


Рисунок 5.1 – Морфологічна карта

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

Таблиця 5.1 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	A	Методична забезпеченість, відносна простота, легкість підтримки, швидкість написання коду, зручність відлагодження	Строго визначена функціональність
	B	Більший набір інструментів	Займає більше часу при написанні коду, обмеженість методичного матеріалу, порівняно важче відлагодження
F2	A	Безкоштовність	Менша функціональність та зручність
	B	Великі функціональні можливості	Платний продукт
F3	A	Швидкість внесення змін та тестування, мала ймовірність виникнення помилки та невідповідності	Матеріальні затрати на оренду сервера
	B	Безкоштовність	Необхідність перенесення нової версії ПП на пристрої вручну, ймовірність виникнення помилок та невідповідностей

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто

відкинути, тому що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F1:

Оскільки функціональності, яку забезпечує варіант а), цілком вистачає для реалізації ПП, а легкість підтримки, що забезпечується наявністю методичних матеріалів та порівняною простотою інструменту, є однією з основних вимог до ПП, то варіант б) слід відкинути.

Функція F2:

Оскільки варіанти а) і б) мають свої відмінності, але обидва забезпечують реалізацію ПП, вважатимемо їх гідними розгляду.

Функція F3:

Оскільки необхідно, щоб ПП був у постійному доступі та легко тестувався на різних пристроях, то варіант б) має бути відкинтий.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

F1a – F2a – F3a

F1a – F2б – F3a

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

5.2 Обґрунтування системи параметрів ПП

5.2.1 Опис параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

X1 – швидкодія фреймворку;

X2 – об'єм оперативної пам'яті на машині користувача;

X3 – час обробки даних;

X4 – потенційний об'єм програмного коду.

X1: Відображає швидкодію операцій залежно від обраної мови програмування.

X2: Відображає об'єм пам'яті в оперативній пам'яті персонального комп'ютера, необхідний для збереження та обробки даних під час виконання програми.

X3: Відображає час, який витрачається на дії.

X4: Показує розмір програмного коду який необхідно створити безпосередньо розробнику.

5.2.2 Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у табл. 4.2.

Таблиця 5.2 – Основні параметри ПП

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			кращі	середні	гірші
Швидкодія фреймворку	X1	Оп/мс	20000	11000	2000
Об'єм оперативної пам'яті на машині користувача	X2	Мб	32	64	96
Час обробки даних	X3	мс	20	160	300
Потенційний об'єм програмного коду	X4	кількість строк коду	1000	1400	1800

За даними таблиці 5.2 будуються графічні характеристики параметрів – рис. 5.2 – рис. 5.5.

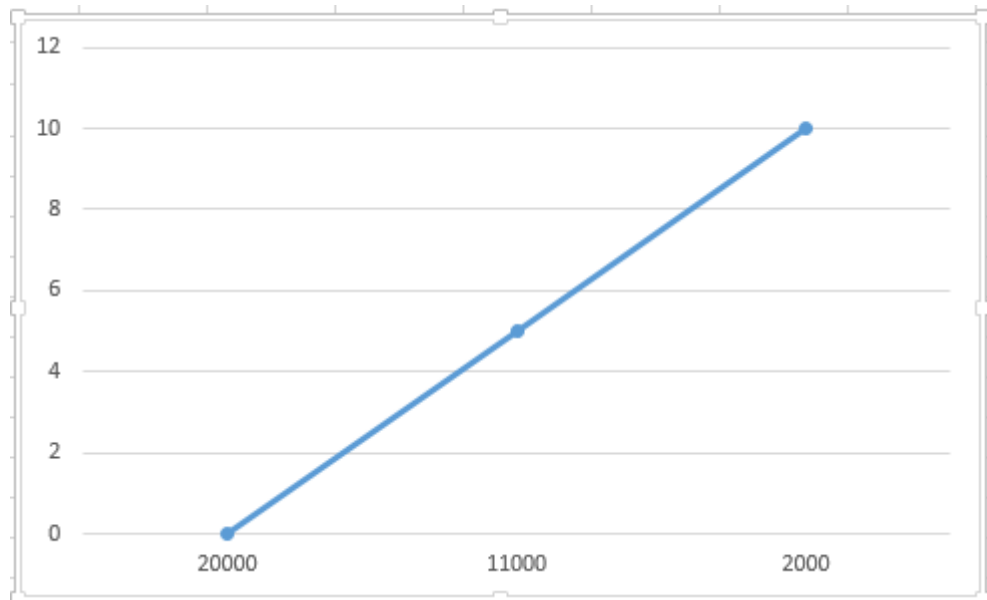


Рисунок 5.2 – X1, швидкодія фреймворку

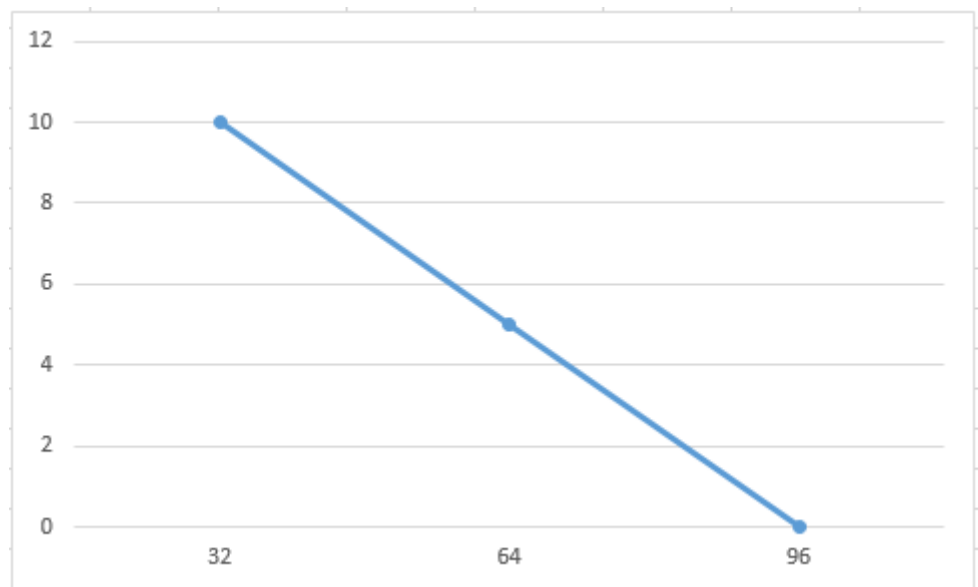


Рисунок 5.3 – X2, об'єм оперативної пам'яті на машині користувача

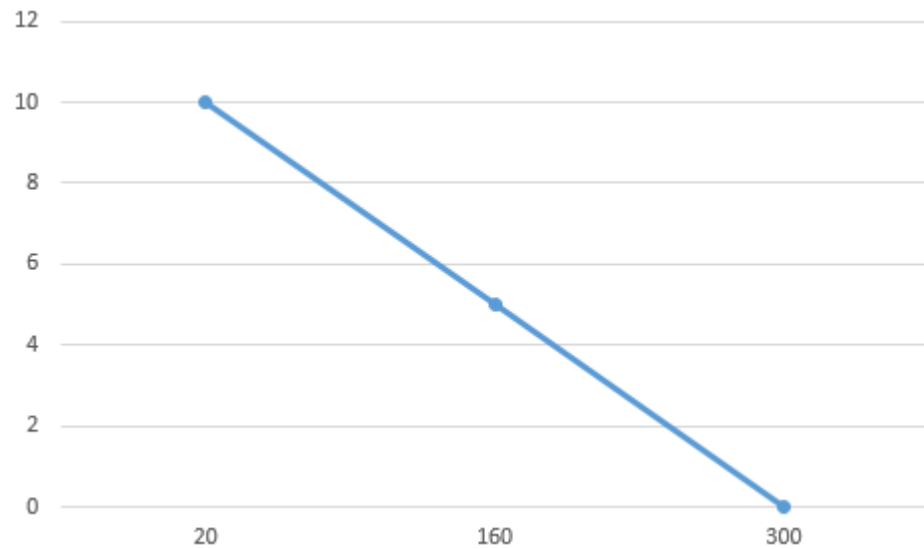


Рисунок 5.4 – X3, час обробки даних

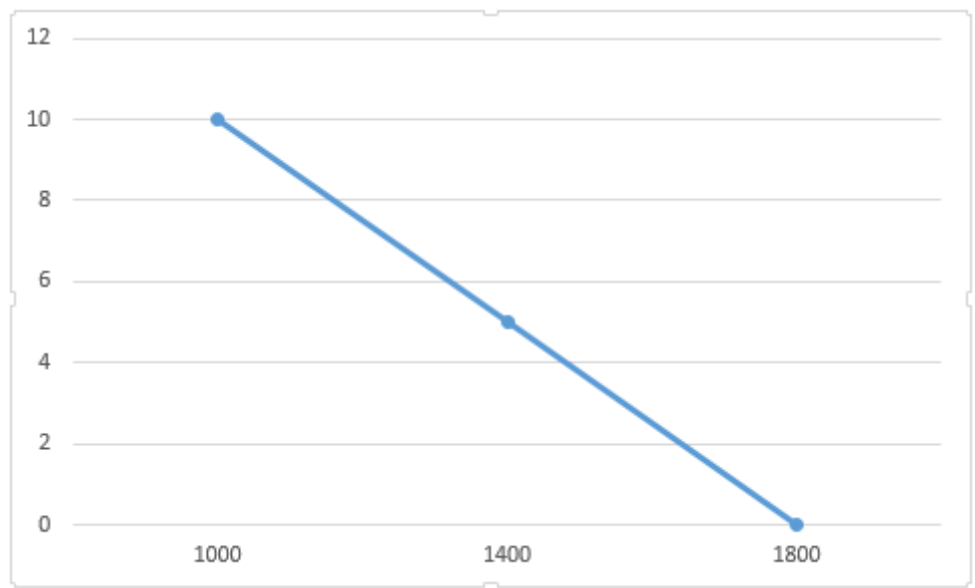


Рисунок 5.5 – X4, потенційний об'єм програмного коду

5.2.3 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні

параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 5.3.

Таблиця 5.3 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів в R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X1	Швидкодія фреймворку	Оп/мс	3	2	3	2	3	3	3	19	1,5	2,25
X2	Об'єм оперативної пам'яті на машині користувача	Мб	1	1	2	3	1	2	1	11	-6,5	42,2
X3	Час обробки даних	Мс	2	3	1	1	2	1	2	12	-5,5	30,2
X4	Потенційний об'єм програмного коду	кількість строк коду	4	4	4	4	4	4	4	28	10,5	110,
	Разом		10	10	10	10	10	10	10	70	--	185

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_{ij} = \sum_{j=1}^N r_{ij}, \quad (5.1)$$

$$R_i = \frac{n(n+1)}{2} = 10, \quad (5.2)$$

де N – число експертів, n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_i = 17,5. \quad (5.3)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad (5.4)$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 185. \quad (5.5)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 185}{7^2(4^3-4)} = 0,755 > W_k = 0,67 \quad (5.6)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 5.4.

Таблиця 5.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	<	>	<	<	<	<	0,5
X1 і X3	<	>	<	<	<	<	<	<	0,5
X1 і X4	>	>	>	>	>	>	>	>	1,5
X2 і X3	>	>	<	<	>	<	>	>	1,5
X2 і X4	>	>	>	>	>	>	>	>	1,5
X3 і X4	>	>	>	>	>	>	>	>	1,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1,5 \text{ при } X_i > X_j \\ 1,0 \text{ при } X_i = X_j \\ 0,5 \text{ при } X_i < X_j \end{cases} \quad (5.7)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості $K_{\delta i}$ за наступними формулами:

$$K_{\text{вi}} = \frac{b_i}{\sum_{i=1}^n b_i}, \quad (5.8)$$

де:

$$b_i = \sum_{j=1}^N a_{ij}. \quad (5.9)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 5%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{вi}} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (5.10)$$

де:

$$b'_i = \sum_{j=1}^N a_{ij} b_j. \quad (5.11)$$

Як видно з таблиці 5.5, починаючи з 3 ітерації, різниця значень коефіцієнтів вагомості не перевищує 5%, тому більшої кількості ітерацій не потрібно.

Таблиця 5.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2
X1	1,0	0,5	0,5	1,5	3,5	0,2188	12,25	0,20762712	44,875	0,20775463
X2	1,5	1,0	1,5	1,5	5,5	0,3438	21,25	0,36016949	77,875	0,36053241
X3	1,5	0,5	1,0	1,5	4,5	0,2813	16,25	0,27542373	59,125	0,27372685
X4	0,5	0,5	0,5	1,0	2,5	0,1563	9,25	0,15677966	34,125	0,15798611
Всього:					16	1	59	1	216	1

5.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X1 (швидкодія фреймворку), X2 (об'єм оперативної пам'яті на машині користувача) та X3 (час обробки даних) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X4 (потенційний об'єм програмного коду) обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту а) 1200 LoC або варіанту б) 1600 LoC.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 5.6):

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j}, \quad (5.12)$$

де n – кількість параметрів; K_{ei} – коефіцієнт вагомості i -го параметра; B_i – оцінка i -го параметра в балах.

Таблиця 5.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметр	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1(X1, X2)	А	X1	16000	7,78	0,2077546	1,61633102
		X2	48	7,5	0,3605324	2,70399308
F2(X4)	А	X4	1600	2,5	0,1579861	0,39496528
	Б	X4	1200	7,5	0,1579861	1,18489583
F3(X3)	А	X3	50	8,93	0,2737269	2,44438077

За даними з таблиці 5.6 за формулою

$$K_K = K_{TY}[F_{1k}] + K_{TY}[F_{2k}] + \dots + K_{TY}[F_{zk}], \quad (5.13)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = F1a + F2a + F3a = 1,616 + 2,704 + 0,395 + 2,444 = 7,16$$

$$K_{K2} = F1a + F2б + F3a = 1,616 + 2,704 + 1,185 + 2,444 = 7,95$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

5.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Обидва варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;

2. Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в обох завданнях, належать до групи 1.

Завдання 1 використовує контроль вхідної інформації, яка характеризується групою 5,1 (вхідні дані та документи різноманітного розміру і структури), та контроль вихідної інформації, яка характеризується групою 5,3 (друкування документів складної багаторівневої структури, різноманітної форми і змісту). Завдання 2 використовує контроль вхідної інформації, яка характеризується групою 5,2 (вхідні дані та документи одноманітної форми і змісту), та контроль вихідної інформації, яка характеризується групою 5,4 (друкування документів одноманітної форми і змісту, видів масивів даних на машинні носії).

Для реалізації завдання 1 використовується нормативно-довідкова інформація (НДІ). Завдання 2 використовує перемінну інформацію (ПІ).

Використовуються мови програмування високого рівня – JavaScript та HTML.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (5.14)$$

де T_P – трудомісткість розробки ПП; K_{Π} – поправочний коефіцієнт; $K_{СК}$ – коефіцієнт на складність вхідної інформації; K_M – коефіцієнт рівня мови програмування; $K_{СТ}$ – коефіцієнт використання стандартних модулів і

прикладних програм; $K_{ст.м}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_P = 90$ людино-днів. Поправковий коефіцієнт, який враховує вид використаної інформації для першого завдання (нормативно-довідкова інформація): $K_{п} = 1,7$. Поправковий коефіцієнт, який враховує складність контролю вхідної та вихідної інформації, для першого завдання становить $K_{ск} = 1,16$. Під час розробки першого завдання не використовуються стандартні модулі, тому коефіцієнт $K_{ст}$ не впливає на розрахунок трудомісткості завдання і становить 1. Коефіцієнти K_M і $K_{ст.п}$, які враховують відповідно програмування на мові низького рівня та розробку стандартного програмного забезпечення, завдань становлять 1: $K_M = K_{ст.п} = 1$. Отже, загальна трудомісткість розробки першого завдання дорівнює ($K_{ст}$, K_M та $K_{ст.п}$ можемо не враховувати):

$$T_1 = 90 \cdot 1.7 \cdot 1.16 = 177,48 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для другого завдання.

Для другого завдання (використовується алгоритм першої групи складності, степінь новизни Б), тобто $T_P = 64$ людино-днів. У другому завданні використовується перемінна інформація, тому $K_{п} = 2.02$. Поправковий коефіцієнт, який враховує складність контролю вхідної та вихідної інформації, для другого завдання становить $K_{ск} = 1$ і не впливає на розрахунок. Аналогічно коефіцієнти $K_{ст}$, K_M і $K_{ст.п}$ становлять 1. Отже, загальна трудомісткість програмування другого завдання дорівнює:

$$T_2 = 64 \cdot 2.02 = 129.28 \text{ людино-днів.}$$

Складаємо трудомісткість 1 і 2 завдань, щоб отримати загальну трудомісткість, яка є рівною для обох варіантів:

$$T = T_I = T_{II} = (177,48 + 129,28) \cdot 8 = 2454,08 \text{ людино-годин.}$$

В розробці беруть участь два програмісти з окладом 10000 грн., один архітектор з окладом 17000грн. Визначимо зарплату за годину за формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.}, \quad (5.15)$$

де M – місячний оклад працівників; T_m – кількість робочих днів тиждень; t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{10000+10000+17000}{3 \cdot 21,1 \cdot 8} = 73,06 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{\text{ЗП}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}}, \quad (5.16)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста; T_i – трудомісткість відповідного завдання; $K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників для обох варіантів становить:

$$C_{\text{ЗП}} = 73,06 \cdot 2454,08 \cdot 1,2 = 215\,154,10 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0,22 = 215\,154,10 \cdot 0,22 = 47333,90 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. ($C_{\text{М}}$)

Так як одна ЕОМ обслуговує одного програміста з окладом 10000 грн., з коефіцієнтом зайнятості 0.2, то для однієї машини отримаємо:

$$C_{\text{Г}} = 12 \cdot M \cdot K_3 = 12 \cdot 10000 \cdot 0,2 = 24000 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{\text{ЗП}} = C_{\text{Г}} \cdot (1 + K_3) = 24000 \cdot (1 + 0,2) = 28800 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0.22 = 28800 \cdot 0.22 = 6336 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 8000 грн.

$$C_A = K_{\text{ТМ}} \cdot K_A \cdot C_{\text{ПР}} = 1.15 \cdot 0.25 \cdot 8000 = 2300 \text{ грн.},$$

де $K_{\text{ТМ}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; K_A – річна норма амортизації; $C_{\text{ПР}}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{\text{ТМ}} \cdot C_{\text{ПР}} \cdot K_P = 1.15 \cdot 8000 \cdot 0.05 = 460 \text{ грн.},$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4 \text{ годин},$$

де D_K – календарна кількість днів у році; D_B , D_C – відповідно кількість вихідних та святкових днів; D_P – кількість днів планових ремонтів устаткування; t_3 – кількість робочих годин в день; K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot C_{\text{ЕН}} = 1706,4 \cdot 0,156 \cdot 1,938 = 515,89 \text{ грн.},$$

де N_C – середньо-споживча потужність приладу, кВт; $C_{\text{ЕН}}$ – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ЗП}} \cdot 0.67 = 28800 \cdot 0,67 = 19296 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}} \quad (5.17)$$

$$C_{\text{ЕКС}} = 28800 + 6336 + 2300 + 460 + 515,89 + 19296 = 57707,89 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 57707,89 / 1706,4 = 33,82 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту, ведуться на ЕОМ, витрати на оплату машинного часу для обох варіантів реалізації складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T$$

$$C_{\text{М}} = 33,82 \cdot 2454,08 = 82996,99 \text{ грн.};$$

Накладні витрати складають 67% від заробітної плати:

$$C_{\text{Н}} = C_{\text{ЗП}} \cdot 0,67$$

$$C_{\text{Н}} = 215\,154,10 \cdot 0,67 = 144153,25 \text{ грн.};$$

Отже, вартість розробки ПП для першого варіанту становить:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}} \quad (5.18)$$

$$C_{\text{ПП}} = 215\,154,10 + 47333,90 + 82996,99 + 144153,25 = 489\,638,24 \text{ грн.};$$

Для варіанту II слід додати матеріальні витрати на середовище розробки ПП JetBrains WebStorm – ліцензія на 1 рік на 2 ЕОМ. Ліцензія на 1 ПК коштує 129\$, на 2 – 258\$. Станом на 1 червня 2017 року, курс доллара НБУ становить 26,3103. Вартість двох ліцензій – 6788,06 грн.

Таким чином, для 2-го варіанту маємо:

$$C_{\text{ПП}} = 215\,154,10 + 47333,90 + 82996,99 + 144153,25 + 6788,06 = 496\,426,30 \text{ грн.}$$

5.5 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{TEP}j} = K_{Kj} / C_{\Phi j},$$

$$K_{\text{TEP}1} = 7,16 / 489\,638.24 = 0,1462 \cdot 10^{-4};$$

$$K_{\text{TEP}2} = 7,95 / 496\,426.30 = 0,1601 \cdot 10^{-4};$$

Як бачимо, більш ефективним є другий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{TEP}1} = 0,1601 \cdot 10^{-4}$.

5.6 Висновки до розділу

В даному розділі проведено повний функціонально-вартісний аналіз ПП, який було розроблено в рамках дипломного проекту. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження ПП з технічної точки зору: було визначено основні функції ПП та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

Другу частину ФВА присвячено вибору із альтернативних варіантів реалізації найбільш економічно обґрунтованого. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу, оптимальним є другий варіант реалізації програмного продукту. У

нього виявився кращий показник техніко-економічного рівня якості $K_{\text{TEP}} = 0,1601 \cdot 10^{-4}$.

Цей варіант реалізації програмного продукту має такі параметри:

фреймворк A-Frame;

IDE JetBrains WebStorm;

збереження виконуваних файлів ПП на виділеному сервері з режимом доступу.

Даний варіант виконання програмного комплексу дає користувачу високу швидкодію. Також даний варіант забезпечує зручність розробки ПП, оскільки використовується фреймворк, який має серед своїх переваг простоту використання, порівняно більшу кількість методичного матеріалу, і тому забезпечує легкість супроводу. Також даний варіант забезпечує розробників зручним способом оновлення виконуваних файлів, відстеження змін та відлагодження ПП, оскільки передбачається використання виділеного сервера.

ВИСНОВКИ

У даній дипломній роботі було здійснено системний аналіз предметної області. Було виділено особливості, характерні для даної сфери, а також проблеми та задачі, які стоять перед фахівцями. Відповідно, було обґрунтовано актуальність та важливість поставленої мети.

В результаті аналізу було виділено основні підходи до розробки додатків для веб з підтримкою віртуальної реальності. Таким чином, додатки VR умовно діляться на ті, що використовують готовий панорамний контент, і ті, що генерують контент за допомогою власних ресурсів. Але в при будь-якому з зазначених підходів для надання VR-інтерфейсу використовується API WebVR. Даний проект є експериментальним і знаходиться на стадії розробки.

Слід зазначити, що даний інтерфейс на разі підтримується лише деякими браузерами. Але для 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 VR 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 – клавіатурою і мишею.

WebVR використовує можливості WebGL для уніфікації доступу до графічного контенту задля його відображення, використовуючи полотно WebGL як джерело VR-контенту. Але WebVR не має інструментів для створення та обробки графіки у форматі для VR, а лише інтерфейс 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, а також побудований на базі Three.js фреймворк A-Frame. Впродовж подальшої роботи було досліджено та проаналізовано можливості цих інструментів розробника, охарактеризовано їхні особливості.

Також було описано способи розробки VR додатків як на основі JavaScript фреймворків, так і на базі ігрових рушіїв тривимірного моделювання. Було описано можливості розробки на базі PlayCanvas з онлайн-інтерфейсом, а також зазначено особливості при використанні рушіїв на кшталт Unity та Unreal, які призначені для створення нативних додатків, але завдяки source-to-source компіляції можуть використовуватись у веб-розробці для VR.

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.

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

Також було здійснено порівняння інструментів розробника для реалізації VR у веб. Для цього було створено додатки з підтримкою VR на базі кожного з фреймворків. В результаті було визначено, що кожен інструмент має як сильні, так і слабкі сторони. Приймаючи до уваги всі характеристики, можна стверджувати, що найбільш зручним у розробці є інструмент A-Frame, при тому, що він допустимо програє лише у швидкості запуску додатку, оскільки базується на технології Three.js.

Було також розроблено 2 веб додатки з підтримкою віртуальної реальності на базі A-Frame. Впродовж цієї роботи було здійснено глибоке дослідження

ресурсів технологій, визначено особливості роботи з фреймворком, оглянуто наявні open-source напрацювання та отримано цінні практичні навички створення додатків VR.

Було здійснено функціонально-вартісний аналіз програмного продукту, що розроблявся. Було проаналізовано варіанти реалізації функції, здійснено їхній економічний аналіз та за результатами порівняння обрано кращий.

Даний дипломний проект стане практично застосовним матеріалом для подальшого вивчення та поглиблення знань у сфері розробки додатків віртуальної реальності в цілому та веб-додатків з підтримкою VR зокрема.

ПЕРЕЛІК ПОСИЛАНЬ

1. 3dspace: VR for Web Developers. – Режим доступу: 3dspace.com/2016/04/vr-for-web-developers/. – Дата доступу: 01.06.2017.
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: 01.06.2017.
3. Holographica: Google опубликовала приложение для создания VR-фотографий. – Режим доступу: <http://holographica.space/news/google-cardboard-camera-android-2511>. – Дата доступу: 01.06.2017.
4. WhatIs.com: room-scale VR. – Режим доступу: <http://whatis.techtarget.com/definition/room-scale-VR-room-scale-virtual-reality>. – Дата доступу: 01.06.2017.
5. Designmodo: Virtual Reality and Web Design: Everything You Need to Know. – Режим доступу: designmodo.com/virtual-reality-design/. – Дата доступу: 01.06.2017.
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: 01.06.2017.
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: 01.06.2017.
8. XinReality: Timewarp. – Режим доступу: <https://xinreality.com/wiki/Timewarp>. – Дата доступу: 01.06.2017.
9. Офіційний сайт проекту WebVR. – Режим доступу: <https://webvr.info/developers/>. – Дата доступу: 01.06.2017.

- 10.Офіційний опис проекту WebVR. – Режим доступу: <https://github.com/w3c/webvr/blob/master/explainer.md>. – Дата доступу: 01.06.2017.
- 11.Документація WebVR. – Режим доступу: <https://w3c.github.io/webvr/spec/1.1/>. – Дата доступу: 01.06.2017.
- 12.Офіційний блог Google для розробників. – Режим доступу: <https://developers.google.com/web/fundamentals/vr/getting-started-with-webvr/>. – Дата доступу: 01.06.2017.
- 13.Офіційний блог Mozilla для розробників.. – Режим доступу: https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/WebVR. – Дата доступу: 01.06.2017.
- 14.Документація до бібліотеки WebGL. – Режим доступу: https://developer.mozilla.org/uk/docs/Web/API/WebGL_API. – Дата доступу: 01.06.2017.
- 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: 01.06.2017.
- 16.Документація до Three.js. – Режим доступу: <https://threejs.org/docs/index.html#manual/introduction/Creating-a-scene>. – Дата доступу: 01.06.2017.
- 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. – Дата доступу: 01.06.2017.

- 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: 01.06.2017.
- 20.Офіційний сайт Babylon.js. – Режим доступу: <http://www.babylonjs.com/>. – Дата доступу: 01.06.2017.
- 21.Офіційний сайт A-Frame. – Режим доступу: <https://aframe.io/>. – Дата доступу: 01.06.2017.
- 22.Документація до A-Frame. – Режим доступу: <https://aframe.io/docs/0.5.0/introduction/>. – Дата доступу: 01.06.2017.
- 23.Документація до рушію PlayCanvas. – Режим доступу: <https://developer.playcanvas.com/en/tutorials/legacy/beginner/cardboard-vr/>. – Дата доступу: 01.06.2017.