

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота**  
**на здобуття освітнього рівня магістра**  
за спеціальністю 121 Інженерія програмного забезпечення

на тему:

**ВИКОРИСТАННЯ ТЕХНОЛОГІЙ 3D-ГРАФІКИ ТА ДОПОВНЕНОЇ  
РЕАЛЬНОСТІ ПРИ РОЗРОБЦІ ВЕБЗАСТОСУНКІВ**

Виконала студентка 2-го курсу магістратури  
Ольга КАРПИШИН

\_\_\_\_\_  
(підпис)

Науковий керівник:  
доцент, кандидат фіз.-мат. наук  
Лариса КАТЕРИНИЧ

\_\_\_\_\_  
(підпис)

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

Студент

\_\_\_\_\_  
(підпис)

Роботу розглянуто й допущено до захисту  
на засіданні кафедри інтелектуальних  
програмних систем

«\_\_\_\_» \_\_\_\_\_ 2023 р.,

протокол № \_\_\_\_

Завідувач кафедри

Олександр ПРОВОТАР

\_\_\_\_\_  
(підпис)

Київ – 2023

## РЕФЕРАТ

Обсяг роботи – 62 сторінки, 31 використане джерело, 31 рисунок, 4 таблиці, 1 додаток.

ДОПОВНЕНА РЕАЛЬНІСТЬ, 3D-ГРАФІКА, 3D-АНІМАЦІЯ, AR.JS, THREE.JS, REACT.JS, МОДЕЛЮВАННЯ, РЕНДЕРИНГ, ФРЕЙМВОРК, КОМЕРЦІЯ.

Об'єктом роботи є процес використання технологій 3D-графіки та доповненої реальності під час розробки вебзастосунків. Предметом роботи є програмний засіб для наочного представлення можливостей цих технологій.

Мета випускної кваліфікаційної роботи полягає у вивченні технологій 3D-графіки та доповненої реальності та створенні на базі отриманих знань вебзастосунку із функціями перегляду об'єктів у 3D- та AR-режимах, сфера використання застосунку – електронна комерція.

Інструменти розробки: мова програмування JavaScript, редактор коду Visual Studio Code 1.77, відкрита JavaScript бібліотека для створення інтерфейсів користувача React, візуалізатор 3D-моделей React Three Fiber, бібліотека для роботи з доповненою реальністю AR.js, бібліотека для стилізування TailwindCSS, бібліотека для анімації інтерфейсів Framer Motion, модель штучного інтелекту для генерації зображень з текстового опису DALL·E 2, засоби для розробки серверної частини вебзастосунків Node.js та npm, система контролю версій Git, вебсервіс GitHub.

Результати роботи: виконано загальний огляд технології 3D-графіки та доповненої реальності для створення вебзастосунків, досліджено сучасні методи реалізації технологій та тенденції їхнього розвитку, розроблено програмний продукт у вигляді інтернет-магазину із 3D- та AR-функціоналом.

Значимість роботи: практична цінність кваліфікаційного проекту полягає у можливості впровадження його результатів у сфері послуг та

розваг, освіти, туризму для покращення користувацького досвіду за рахунок візуального представлення об'єктів.

На базі реалізованої практичної частини можна створювати аналогічні інтернет-магазини, змінюючи при цьому контент для наповнення сайту. Серед напрямків удосконалення об'єкта розроблення варто виділити такі як:

- розробка інтерактивного мобільного додатку;
- розширення категорій товарів;
- використання штучного інтелекту для рекомендацій товарів;
- використання різних видів доповненої реальності;
- віртуальні тури приміщенням.

Технології, пов'язані із 3D-моделюванням та доповненою реальністю, розвиваються стрімкими темпами, попит на використання цих технологій на ринку зростає з кожним роком, тому доцільність продовження розробки виправдана.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ .....	6
ВСТУП .....	7
РОЗДІЛ 1. ЗАГАЛЬНИЙ ОГЛЯД 3D-ГРАФІКИ ТА АНІМАЦІЇ .....	10
1.1 Історія виникнення та розвитку комп'ютерної графіки .....	10
1.2 Створення об'ємного зображення .....	14
1.2.1 Тривимірна графіка та її переваги порівняно із двовимірною.....	14
1.2.2 Аналіз сучасних методів 3D-модельовання.....	16
РОЗДІЛ 2. ІНСТРУМЕНТИ ДЛЯ ІНТЕГРАЦІЇ 3D-ГРАФІКИ У ВЕБ- СЕРЕДОВИЩЕ .....	22
2.1 Загальний огляд WebGL.....	22
2.2 Найпопулярніші JavaScript бібліотеки та фреймворки для використання тривимірної графіки.....	23
2.2.1 Бібліотека Three.js.....	24
2.2.2 Фреймворк A-Frame.....	32
2.2.3 Фреймворк Babylon.js.....	33
2.2.4 Рушій PlayCanvas .....	34
РОЗДІЛ 3 ЗАГАЛЬНИЙ ОГЛЯД ПОНЯТТЯ ДОПОВНЕНОЇ РЕАЛЬНОСТІ	36
3.1 Історія розвитку доповненої реальності .....	36
3.2 Як працює доповнена реальність .....	38
3.3 Ключові технології доповненої реальності .....	40
3.3.1 Технологія інтелектуального дисплея .....	40
3.3.2 Технологія 3D-реєстрації .....	41
3.3.3 Технологія інтелектуальної взаємодії.....	42
3.4 Види доповненої реальності .....	42

	5
3.4.1 Доповнена реальність на основі маркерів.....	43
3.4.2 Безмаркерна доповнена реальність.....	44
3.4.3 Доповнена реальність на основі розташування.....	45
3.4.4 Накладання доповненої реальності.....	45
3.4.5 Доповнена реальність на основі проєкції.....	46
3.4.6 Окреслення доповненої реальності.....	46
3.5 Різниця між доповненою та віртуальною реальністю.....	47
3.6 Головні тренди доповненої реальності.....	49
<b>РОЗДІЛ 4 ОГЛЯД ІНСТРУМЕНТІВ РОЗРОБКИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.....</b>	<b>51</b>
4.1 Платформа Vuforia.....	51
4.2 Бібліотека AR.js.....	52
4.3 Платформа 8th Wall.....	53
4.4 Платформа ZapWorks.....	54
<b>РОЗДІЛ 5 ПРАКТИЧНА ЧАСТИНА.....</b>	<b>55</b>
5.1 Вибір моделі розробки.....	55
5.2 Огляд використаних технологій.....	55
5.3 Специфікація користувацьких вимог.....	58
5.4 Архітектура системи.....	59
5.5 Тестування.....	60
5.6 Опис кінцевої системи.....	60
5.7 Способи покращення функціонування системи.....	60
<b>ВИСНОВКИ.....</b>	<b>62</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....</b>	<b>63</b>
<b>ДОДАТОК А.....</b>	<b>67</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

AR – Augmented Reality, доповнена реальність;

VR – Virtual Reality, віртуальна реальність;

3D – тривимірний;

API – Application Programming Interface, прикладний програмний інтерфейс;

WebGL – Web Graphics Library, прикладний програмний інтерфейс JavaScript для створення та відтворення тривимірних графічних зображень у браузері;

UI – User Interface, інтерфейс користувача;

JSX – JavaScript XML, синтаксис розширення JavaScript, який використовується в бібліотеці React для опису компонентів і створення динамічного вмісту на стороні клієнта;

MR – Mixed Reality, змішана реальність;

DOM – Document Object Model, об'єктна модель документа.

## ВСТУП

**Оцінка сучасного стану об'єкта розробки.** Із розвитком високих технологій 3D-графіка та доповнена реальність набули величезної популярності та перебувають у стані активного розвитку. Саме вони дали змогу створювати об'ємні та анімовані моделі, починаючи від макетів окремих предметів і продовжуючи проєкціями цілих віртуальних світів. Для підтримки належного рівня якості в сучасному технологічному та конкурентному середовищі, відбувається постійне удосконалення цих сфер.

У галузі 3D-моделювання та графіки набули популярності технології рендерингу в реальному часі, які дозволяють створювати більш деталізовані та реалістичні об'єкти з швидкістю, достатньою для використання в іграх та інших реалізаціях віртуальної реальності.

Доповнена реальність досягається за допомогою різноманітних технологічних інновацій, які можуть бути реалізовані окремо або в поєднанні один з одним. Ці технології включають такі компоненти як: процесор, дисплеї, датчики та пристрої введення, програмне забезпечення. Наразі існують додатки, доступні або досліджувані для AR майже в кожному промисловому секторі: мистецтві, архітектурі, комерції, сфері розваг тощо.

**Актуальність роботи та підстави для її виконання.** За останні роки технології 3D-графіки та доповненої реальності розширили свої можливості. Розробники знайшли способи їхнього впровадження у різних сферах життя, включаючи інтерактивні рекламні кампанії, вебдизайн, віртуальні тури, візуалізацію даних, створення ігор, медичну, освітню сфери та відкрили інноваційні способи їхнього застосування по всьому світу.

Вебзастосунки з використанням технологій 3D-графіки та доповненої реальністю є актуальною та перспективною темою для дослідження та розробки. Використання таких технологій дозволить покращити якість програмного продукту та забезпечити їх більш ефективну роботу.

**Мета й завдання роботи.** Метою випускної кваліфікаційної роботи є реалізація власного вебзастосунку з використанням елементів 3D-графіки та доповненої реальності. З метою досягнення цієї мети було поставлено такі завдання:

- ознайомитися із концепціями 3D-графіки та доповненої реальності;
- дослідити наявні засоби для їхньої імплементації у вебзастосунках;
- виокремити переваги та недоліки використання технологій для роботи з 3D і AR;
- використати набуті знання для створення вебзастосунку.

**Об'єкт, методи й засоби розроблення.** Об'єктом розроблення є вебзастосунок для електронної комерції із такими функціональними характеристиками як: перегляд товарів у вигляді 3D-моделей, інтерактивна зміна зображення на товарі, перегляд предметів за допомогою доповненої реальності, використання штучного інтелекту при створенні візерунків, можливість завантаження отриманого результату.

Для розроблення проєкту обрано послідовний метод розробки програмного забезпечення – водоспадну модель.

Інструментами розроблення системи було обрано мову програмування JavaScript, безкоштовний редактор та відлагоджувач коду Visual Studio Code 1.77, Node.js, відкриту JavaScript бібліотека для створення інтерфейсів користувача React, програмний пакет для створення 3D-моделей React Three Fiber, бібліотеку для роботи з доповненою реальністю AR.js, бібліотеку для стилізування TailwindCSS, бібліотеку для анімації інтерфейсів Framer Motion, модель штучного інтелекту для генерації зображень з текстового опису DALL·E 2, засоби для розробки серверної частини вебзастосунків Node.js та npm, система контролю версій Git, вебсервіс GitHub.

**Можливі сфери застосування.** Запит на розробку вебзастосунків з використанням 3D-графіки та доповненої реальності стає все більш актуальним в сучасному інтернет-середовищі. Розвиток технологій 3D-графіки та доповненої реальності відкриває широкі можливості для

подальшого впровадження цих технологій у різні сфери життя, вони можуть застосовуватись у архітектурі та будівництві – для візуалізації проектів будівель та споруд в 3D-форматі, що дозволяє забезпечити більш точне розуміння будівельних проектів для замовників та інвесторів, у промисловості – для моделювання та тестування нових продуктів в 3D-форматі, що дозволяє забезпечити більш точні результати тестування перед випуском продукту на ринок, для освіти та науки – для використання 3D-моделей та доповненої реальності для покращення процесу навчання та досліджень в різних галузях науки та техніки, для медіа та розваги – для створення більш реалістичних та захопливих ігрових та розважальних додатків, які можуть привернути увагу більшої кількості користувачів та забезпечити кращий дохід для розробників, у туризмі та готельній справі – для створення вебзастосунків з використанням доповненої реальності, які можуть допомогти туристам ознайомитися з новими місцями та готелями до їх приїзду.

**Взаємозв'язок з іншими роботами.** Проект магістерської кваліфікаційної роботи частково пов'язаний із системою, яка була розроблена у ході виконання практичної частини для балаклавської кваліфікаційної роботи, оскільки в обох випадках відбувається розробка вебзастосунку електронної комерції. У майбутньому планується поєднання двох проєктів.

## РОЗДІЛ 1. ЗАГАЛЬНИЙ ОГЛЯД 3D-ГРАФІКИ ТА АНІМАЦІЇ

### 1.1 Історія виникнення та розвитку комп'ютерної графіки

Вважається, що термін «комп'ютерна графіка» був створений Вільямом Феттером – американським дизайнером, який з 1960-го року розпочав дослідження основ комп'ютерної анімації, використовуючи каркас фігури людського тіла від. Ним була створена тривимірна модель «Boeing Man», яка стала «пілотом» у короткометражній анімації 1964-го року. Сам Вільям Феттер присвоює авторство терміну своєму колезі Верну Хадсону.

У середині 1960-х років Девід Еванс та Айван Сазерленд заснували кафедру комп'ютерної графіки в Університеті Юти, США, де було розроблено базові технології 3D-графіки, такі як затемнення по Гуро, Фонгу та Бліну, текстуркування і алгоритми виявлення невидимої поверхні. Експерти, в тому числі такі видатні фахівці як Джим Блінн, Буй Тіонг Фонг і Анрі Гуро, працювали на кафедрі та зробили значний вклад у розвиток алгоритмів текстуркування і затемнення. У 1962-му році Сазерленд створив програму Sketchpad, яка передувала всім сучасним 3D-редакторам та САД-системам. Її визначальною особливістю було введення концепції об'єктів та екземплярів, що дозволяло їх зберігати, редагувати і використовувати повторно в інших роботах. Програма також автоматично промальовувала фігури з заданими координатами за допомогою світлового пера на дисплеї [1].

У 1968 році Айваном Сазерлендом та Девідом Евансом засновано компанію Evans & Sutherland. Компанія сфокусувала свою діяльність на виробленні новітнього обладнання, що підтримувало роботу систем, спроектованих в університеті, де працювали її засновники. Значна частина цих алгоритмів була пізніше реалізована в головному апаратному забезпеченні, а саме – геометричний конвеєр, дисплей з кріпленням до голови, кадровий буфер та авіаційний тренажер. Більшість працівників компанії були або колишніми випускниками, або студентами кафедри, серед яких ми сьогодні

можемо зустріти і Джима Кларка, засновника компанії Silicon Graphics, і Едвіна Кетмелла, який став президентом і технічним директором відомої анімаційної студії Pixar, на рахунок якої – перший повнометражний мультфільм, виконаний у редакторах та програмах для тривимірного простору, а також Джона Варнока, одного із основоположників Adobe Systems [2].

Приблизно у той самий час, група фахівців з СРСР, на чолі з Миколою Константиновим, використовуючи обчислювальну машину БЕСМ-4, змогли змодельовати рухи кішки з використанням систем диференціальних рівнянь другого порядку. Завдяки цьому, у 1968-му році було створено перший анімований комп'ютерний персонаж – «Кішечка», який з'явився на екранах завдяки друку кадрів на принтері, які потім були об'єднані в стрічку [3].

Вимоги з різних галузей людської діяльності росли швидкими темпами, тому було необхідно вдосконалювати технічні можливості та оптимізувати алгоритми для обробки графічних об'єктів, а також розробити зручні інструменти для їхнього використання.

В першу чергу, запит йшов від кіноіндустрії, яка потребувала значного технічного розвитку. У 1973-му році з'явився фільм «Дикий Захід», у якому використовувалась цифрова обробка зображень. Після цього вперше були використані каркасні тривимірні зображення у продовженні фільму в 1976-му році. Автори картини, Едвін Кетмелл та Фред Парке, у 1972-му році створили експериментальну роботу «Комп'ютерна анімація руки», відкрила новий етап розвитку сфери 3D-графіки. У короткометражному фільмі застосовувалися новітні техніки для генерування комп'ютером руки та обличчя – 3D рендеринг, який до сьогоднішнього дня використовується у відеоіграх, фільмах та спецефектах.

Якщо ж говорити про іншу, не менш важливу, галузь – автомобілебудування, то знаменною подією стала розробка П'єром Безьє у 1962 році на замовлення компанії Renault надзвичайно простого і збірного опису будь-яких форм площини, необхідних для автоматизованої роботи

механізмів по обробленню листового металу. Його система параметрично заданих кривих залягла у фундамент багатьох графічних програм.

На початку 1970-х років виникла необхідність перенесення графічних тривимірних зображення, що найбільш точно відображали реальний світ, на комп'ютер. За вирішення цієї проблеми взявся Анрі Гуро, наступник Айвана Сезерленда. У 1971 році він розробив алгоритм, який дозволяв ретельно промальовувати тіні з плавним переходом між різними кольорами, залежно від кута падіння променів та віддаленості джерела світла. Цей метод створював ілюзію гладкої криволінійної поверхні, що складалася з полігонів із плоскими гранями. Він досягався за допомогою інтерполяції кольорів між сусідніми гранями. Через два роки Буй Тіорнг Фонг, колега Гуро, презентував свій метод тонування об'єктів, який базувався на інтерполяції нормалей поверхні та растеризації полігонів і розрахунку кольорів пікселів з використанням інтерпольованої нормалі та моделі відбивання світла. Також варто згадати демонстрація на конференції SIGGRAPH у 1975-му році моделі чайника, створеної Мартіном Ньюллем. У попередньому році він опублікував дисертацію «Алгоритм моделювання підрозбиттів при створенні вигнутих поверхонь на екрані комп'ютера», в якій вивчав такі основні питання, як накладання текстури, В-кубічні фрагменти і Z-буфер. У 1978-му році Джеймс Блінн розробив технологію правдоподібної візуалізації 3D-об'єктів, що називалася рельєфним текстурованням. Ця техніка полягала у створенні ілюзії глибини та текстури за допомогою світла та тіні. Рельєфне текстуровання використовувало особливі картини, що називаються бамп-мапи, які містять відтінки сірого. Ці картини використовувалися для створення малюнків, що мали вигляд рельєфних поверхонь з піднятими та затопленими ділянками. Досягнення Блінна дозволило створювати більш детальні та реалістичні 3D-зображення, що було величезним кроком вперед у розробці комп'ютерної графіки.

Після 1980-х років, розвиток тривимірної комп'ютерної графіки став настільки швидким, що він дозволив створювати візуальні ефекти та анімацію, які здавалися майже реальними.

У 1990-х роках з'явилися перші спеціалізовані графічні процесори, які значно підвищили продуктивність комп'ютерів в галузі 3D-графіки. Це дозволило створювати відеоігри зі збалансованим співвідношенням між графікою та продуктивністю. Крім того, з'явилася функція рендерингу в режимі реального часу, що робило можливою інтерактивну взаємодію з тривимірним середовищем [4].

У 2000-х роках комп'ютерна графіка продовжувала швидко розвиватися. Було започатковано нові технології, такі як шейдери, які дозволяли розраховувати освітлення та матеріали об'єктів в режимі реального часу. Це значно покращило рівень реалізму графіки в іграх та додатках.

З появою мобільних пристроїв та інтернету швидкість передачі даних значно підвищилася. Відповідно, розширилися області застосування 3D-графіки, такі як веб-дизайн, віртуальна реальність, архітектурне проектування та візуалізація даних.

Сьогодні, тривимірна комп'ютерна графіка застосовується в багатьох сферах життя, включаючи ігри, фільми, архітектуру, медицину, науку та інші. Розвиток технологій продовжується, і ми можемо очікувати ще більш складні та реалістичні візуальні ефекти в майбутньому. Однією з головних тенденцій у сфері тривимірної комп'ютерної графіки є збільшення реалізму та деталізації зображень, що стає можливим завдяки використанню більш потужних процесорів та відеокарт.

Також, в сучасних комп'ютерних іграх активно використовуються штучні інтелект та машинне навчання, що дозволяє створювати більш інтелектуальних та складних персонажів, які можуть пристосовуватися до дій гравця та змінювати свою поведінку відповідно до ситуації.

У майбутньому, можна очікувати подальший розвиток технологій тривимірної комп'ютерної графіки.

Алгоритми формування тривимірного зображення, що максимально наближене до реального, постійно вдосконалюються професіоналами з усього світу. Зрештою, оскільки 3D-графіка та анімації мають настільки великий діапазон застосування, що сьогодні важко уявити світ без них [5].

## **1.2 Створення об'ємного зображення**

### **1.2.1 Тривимірна графіка та її переваги порівняно із двовимірною**

У традиційному двовимірному середовищі побудова об'єкта відбувається на двох осях –  $X$  та  $Y$ , що дозволяє бачити лише одну сторону зображеного предмета. Однак у тривимірному просторі додатково використовується третя вісь –  $Z$ , яка передає інформацію про всі сторони предмета.

Перша перевага використання тривимірної графіки полягає у можливості анімації готової моделі. Після розміщення її в кадрі, фінальне зображення створюється за допомогою спеціальної програми – візуалізатора.

Друга перевага полягає у тому, що одна тривимірна модель використовуватися повторно в різних проектах, а також її можна легко змінювати, деформувати та надавати бажаного зовнішнього вигляду. Ці можливості не завжди доступні у звичайних двовимірних малюнках.

Третя перевага полягає в можливості деталізованого проектування моделей. Ця можливість дозволяє отримати детальний зображення об'єктів під час їхнього збільшення, що є важливим елементом для багатьох проектів.

Найбільш поширеним способом моделювання є полігональне. Полігональне моделювання – це метод створення тривимірних моделей, який базується на використанні геометричних примітивів у вигляді полігонів. Кожний полігон складається з вершин, які визначають його форму та положення в просторі. За допомогою великої кількості полігонів можна створювати детальні та реалістичні 3D-моделі.

Одним з переваг полігонального моделювання є можливість швидкого перетворення моделі з чотирикутників на трикутники для її візуалізації в реальному часі. Однак, збільшення кількості полігонів у моделі призводить до зниження її продуктивності, оскільки збільшується кількість точок, за якими їх будують, що, в свою чергу, вимагає більше ресурсів при обробці процесором. Це вимагає пошуку компромісів між продуктивністю та деталізацією моделі, що призводить до використання високополігональних та низькополігональних моделей залежно від призначення моделі. У комп'ютерних іграх застосовують низькополігональні моделі для досягнення оптимальної продуктивності при візуалізації в реальному часі [6].

Після завершення етапу моделювання отримуємо математичну модель, яка містить інформацію про геометричну форму об'єкта, але не містить інформації про його кольори та текстури. Для того, щоб надати моделі потрібний колір і здатність відображати текстури, використовують спеціальні зображення, які називають текстурами.

Текстура об'єкта – це двовимірне зображення, яке накладається на поверхню тривимірної моделі для того, щоб додати їй колір та деталізацію. Текстури створюються різними способами: процедурними – згенерованими за допомогою алгоритмів, або намальованими у графічному редакторі. Текстури задають лише зображення та колір об'єкта, тоді як здатність віддзеркалення, переломлення, рельєф і прозорість визначаються за допомогою властивостей матеріалу. З точки зору тривимірної графіки, матеріал – це математична модель, що описує параметри поверхні. Перед тим, як накласти текстуру на модель, необхідно зробити її розгортку, тобто, уявити поверхню моделі у вигляді проекції на площину, щоб знайти відповідні точки для накладання текстур. Загалом, процес створення тривимірної моделі складається з кількох етапів.

Спочатку потрібно відмалювати ескіз моделі або знайти відповідне зображення, на основі якого буде створена модель. Далі, на основі цього ескізу або зображення, проводиться моделювання геометричної форми об'єкта.

Потім необхідно створити розгортку, щоб уявити поверхню моделі у вигляді проекції на площину. Після цього використовуються текстури, які накладаються на модель, щоб надати їй колір і додаткові ефекти. Налаштування параметрів матеріалу, таких як текстури, відображення, заломлення, прозорість і так далі, завершують процес створення тривимірної моделі [7].

Після виконання цих етапів модель готова до візуалізації, або ж можна продовжувати роботу з налаштуванням ригінгу (rigging) моделі чи створенням анімації.

Після створення тривимірної моделі, необхідно розмістити її в сцені з іншими об'єктами, додати камеру та освітлення, щоб отримати фінальне зображення. Фізична модель, яка враховує відображення, заломлення та розсіювання променів світла, використовується для прорахунку цього зображення. У традиційному 2D-малюванні художник самостійно малює відблиски, тіні та інші деталі, а в 3D-графіці автор створює сцену з урахуванням геометрії, матеріалів освітлення, властивостей камери, після чого візуалізатор виконує розрахунки та відображає кінцеве зображення [8].

### **1.2.2 Аналіз сучасних методів 3D-моделювання**

У 3D-системах координати складаються з трьох вимірів, і зміна одного параметра може вплинути на інші. Деякі системи мають вбудовані технології аналізу, які автоматично обчислюють такі фізичні характеристики об'єктів, як момент інерції або вага.

У тривимірному просторі виділяють три основні типи моделювання:

- каркасне;
- твердотільне;
- поверхневе.

Каркасна модель складається з вершин, з'єднаних лініями для утворення каркасу об'єкта. Каркасна модель не містить поверхні об'єкта та інформації про товщину, текстуру чи кольори, вона представляє лише його геометричну структуру.

Серед способів створення каркасної моделі варто виділити ручне створення за допомогою спеціальних програм для моделювання, а також автоматичну генерацію з графічних даних, отриманих зі сканування об'єкта або з 3D-сканерів.

Можна виділити такі переваги використання каркасної моделі:

- каркасна модель має значно менший обсяг даних, ніж моделі з поверхнями, що дозволяє ефективніше використовувати ресурси обчислювальної системи;
- каркасні моделі досить зручно модифікувати та змінювати, що дозволяє швидко вносити оновлення до дизайну;
- каркасна модель надає просту та зрозумілу для користувача структуру об'єкта.

Серед недоліків каркасних моделей:

- відсутність інформації про текстуру та колір, що призводить до менш реалістичного вигляду;
- виникнення труднощів при створенні моделі зі складними поверхнями;
- неможливість застосування каркасних моделей для створення детальних анімаційних ефектів без додаткової обробки.

Часто цей метод використовується не в моделюванні, а при відображенні готових моделей, як один із методів візуалізації (рис. 1.1).

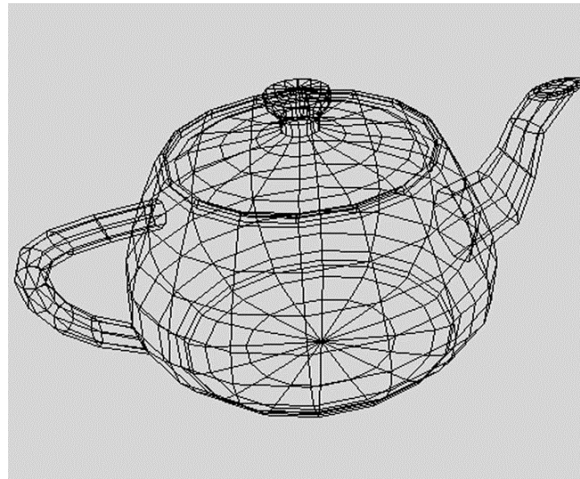


Рисунок 1.1 – Каркасна модель чайника

Наступним методом моделювання є поверхневе моделювання (рис. 1.2), яке визначається в термінах точок, ліній і поверхонь. Метод ґрунтується на тому, що об'єкти обмежуються поверхнями, які відокремлюють їх від довкілля. Ця оболонка зображена графічними поверхнями. Поверхня об'єкта обмежена контурами, які є результатом двох дотичних або поверхонь, що перетинаються. Вершини об'єктів задаються перетином трьох поверхонь.



Рисунок 1.2 – Поверхневе моделювання

При виборі поверхневого моделювання є певні плюси, порівнюючи із каркасним:

- можливість визначення складних криволінійних граней;

- здатність отримання тонових зображень;
- розпізнавання особливих побудов на поверхні, наприклад, отворів;
- отримання зображення високої якості.

В основі моделювання поверхні лежить математичне визначення: усяка поверхня може апроксимуватися многогранником, кожна грань якого складена з найпростішого плоского многокутника. Також в моделі допускаються поверхні другого порядку і поверхні, форма яких визначається завдяки різним способам апроксимації і інтерполяції. Будь-який об'єкт, сформований за допомогою поверхневого моделювання, має внутрішню і зовнішню сторону [9].

Є декілька типів поверхонь:

- базові геометричні поверхні, які виходять при переміщенні однієї плоскої кривої щодо іншої;
- поверхні обертання, створені при обертанні плоскої грані навколо деякої осі;
- поверхні об'єднання і перетинів, створені внаслідок об'єднання або перетину поверхонь;
- скульптурні поверхні, які ніяк не описуються математичними рівняннями і будуються лише при використанні сплайнів, що сполучають точки в просторі;
- аналітичні поверхні, описані математичним рівнянням [10].

Останній метод моделювання – твердотільне моделювання. Твердотільна модель (рис. 1.3) визначається термінами того тривимірного об'єму, який займає тіло, що її визначає. Твердотільне моделювання – найдосконаліший і найбільш достовірний метод створення реального об'єкта.



Рисунок 1.3 – Твердотільне моделювання

У цього методу є ряд переваг:

- можливість розділення внутрішньої і зовнішньої областей об'єкта;
- приховані лінії видаляються автоматично;
- автоматичне створення тривимірних розрізів об'єкта, що важливо при аналізуванні складних виробів;
- використання методів аналізу з автоматичним отриманням зображення конкретних властивостей;
- здатність отримання тонових зображень;

Є два способи створення твердотільних моделей:

- метод конструктивного представлення;
- метод граничного представлення.

Метод конструктивного представлення зводиться до того, що з базових складових елементів (твердотільних примітивів, які визначаються формою, розмірами, точкою прив'язки і орієнтацією) створюється твердотільна модель. Твердотільні моделі будуються за допомогою булевих операцій: об'єднання, перетин, різниця. У даного методу є свої переваги і недоліки. За допомогою методу конструктивного представлення можна забезпечити хорошу точність розрахунків фізичних властивостей об'єкта, однак цей метод відмінний від звичних способів моделювання і є складним у застосуванні без відповідного досвіду роботи.

Другий метод, метод граничного представлення, робить опис границь об'єкта і чітко задає межі, що описують тіло. Лише цей метод дозволяє створити точне представлення геометричного твердого тіла. При цьому підході потрібно задати контури або межі об'єкта і ескізи різних видів на об'єкт, а також вказати лінії зв'язків між цими видами для установки взаємовідповідності.

Порівнюючи обидва методи можна сказати, що метод конструктивного представлення більш зручний при первісній побудові моделі та займає менше місця в базі даних. Однак він, на відміну від методу граничного подання, не дуже зручний для побудови складних форм. Також, хоча в методі конструктивного представлення дані займають менше місця, обсяг розрахунків при відтворенні моделі виявляється більшим. З іншого боку, метод граничного представлення зберігає точний опис границь моделі, який займає багато пам'яті, але майже не вимагає обчислень для відтворення моделі. Перевагою методу граничного представлення також є простота перетворення у каркасну модель і в зворотному напрямку [11].

Поєднання конструктивного і граничного представлень є гібридною системою. Гібридне моделювання дозволяє поєднувати каркасну, твердотільну і поверхневу геометрії, а також використовувати параметричне моделювання.

## РОЗДІЛ 2. ІНСТРУМЕНТИ ДЛЯ ІНТЕГРАЦІЇ 3D-ГРАФІКИ У ВЕБ-СЕРЕДОВИЩЕ

### 2.1 Загальний огляд WebGL

WebGL — це JavaScript API, який дозволяє веброзробникам створювати інтерактивну 3D-графіку та анімацію безпосередньо в браузері. Він базується на специфікації OpenGL ES 2.0, широко використовуваному стандарті для створення 3D-графіки у вбудованих системах, і забезпечує низькорівневий інтерфейс для графічного обладнання сучасних комп'ютерів і мобільних пристроїв. WebGL розроблений Khronos Group, консорціумом технологічних компаній і організацій, які створили і інші відкриті стандарти, такі як OpenGL, OpenCL і Vulkan.

WebGL реалізовано як кросплатформену технологію. Це означає, що вона може працювати на широкому діапазоні пристроїв і операційних систем. Він працює, надаючи JavaScript API, який дозволяє розробникам створювати та керувати 3D-об'єктами, текстурами, шейдерами та іншими графічними ресурсами, які після цього компілюються та виконуються на графічному обладнанні пристрою. Це дозволяє створювати високоефективну 3D-графіку та анімацію в браузері без необхідності будь-яких плагінів або додаткового програмного забезпечення.

Однією з ключових особливостей WebGL є підтримка шейдерів, які є невеликими програмами, що працюють на графічному апаратному забезпеченні та дозволяють створювати складне освітлення, відображення текстур та інші візуальні ефекти. Шейдери написані на мові під назвою GLSL (OpenGL Shading Language) і можуть бути скомпільовані під час їхнього виконання, що дозволяє оновлювати та модифікувати 3D-сцену в реальному часі. WebGL надає низку вбудованих шейдерів для типових завдань, таких як відображення освітлення та тіней.

Іншою важливою особливістю WebGL є підтримка текстур, що є 2D- або 3D-зображеннями, які можна наносити на 3D-об'єкти для створення реалістичних матеріалів і поверхонь. Текстури можна завантажувати з файлів зображень або створювати динамічно за допомогою JavaScript, або ж комбінувати та маніпулювати різними способами для створення складних візуальних ефектів.

WebGL надає ряд інструментів і утиліт для роботи з 3D-графікою: бібліотеку матриць для маніпулювання 3D-перетвореннями, буферну систему для ефективного зберігання та передачі даних, а також ряд інструментів для налагодження та профілювання, які допомагають оптимізувати продуктивність. Він підтримує ряд пристроїв введення, таких як комп'ютерна миша, сенсорна панель, введення з клавіатури та ігрові контролери, використання яких забезпечує інтерактивний користувацький досвід у тривимірному режимі представлення об'єктів.

Однією з проблем роботи з WebGL є те, що це технологія низького рівня, яка вимагає певного рівня досвіду у комп'ютерній графіці та програмуванні. Щоб вирішити цю проблему, існує ряд бібліотек і фреймворків, побудованих на основі WebGL, таких як Three.js, Babylon.js і A-Frame, які забезпечують абстракції вищого рівня та утиліти для створення 3D-графіки та анімації [12].

## **2.2 Найпопулярніші JavaScript бібліотеки та фреймворки для використання тривимірної графіки**

Досить часто процес роботи із WebGL є доволі громіздким, особливо коли йдеться про освітлення, тіні, текстури, адже під час розробки моделі необхідно описати всі складові: точки, лінії, грані тощо.

Теоретично, візуалізація ідеї із залученням 3D-елементів потребує написання великої кількості коду, що не є ефективно з точки зору часу і продуктивності. Саме тому, для прискорення цього процесу розробили

бібліотеки та фреймворки. У наступних підрозділах розглянемо найпопулярніші з них.

### 2.2.1 Бібліотека Three.js

JavaScript бібліотеку – Three.js. Головним автором бібліотеки вважають Рікардо Кабелло, але участь у її розробці взяли багато професіоналів.

Бібліотека містить цілий ряд готових класів для реалізації та відображення тривимірної графіки у WebGL (рис. 2.1). Засоби Three.js дозволяють використовувати звичні терміни, не удаючись до написання шейдерів [13].

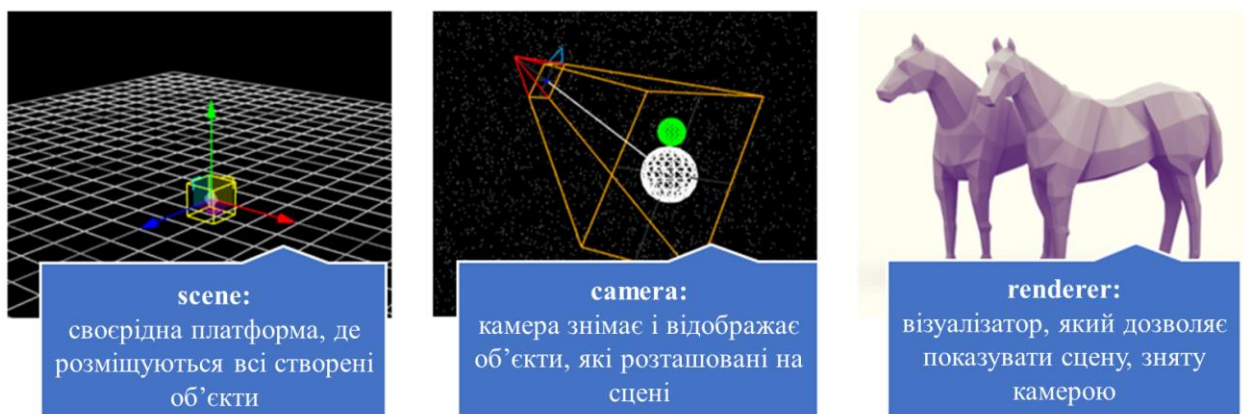


Рисунок 2.1 – Головні поняття, якими оперує бібліотека Three.js

У Three.js існує кілька типів камери (найчастіше застосовують перші дві):

- Perspective Camera;
- Stereo Camera;
- Orthographic Camera;
- Cube Camera.

Perspective Camera – найбільш поширений режим проекції, який використовується для візуалізації 3D-сцени (рис. 2.2). Перспективна камера призначена для імітації того, що бачить людське око. Вона сприймає об'єкти

в перспективній проекції, тобто, чим далі знаходиться об'єкт, тим меншим він здається. Камера приймає чотири аргументи (табл. 2.1):

Таблиця 2.1 – Аргументи камери

Аргумент	Застосування
Field Of View	Визначає кут, який можна бачити навколо центру камери.
Aspect ratio	Визначає співвідношення ширини до висоти екрану. При великих значеннях поля зору видимий розмір об'єктів швидко зменшується при віддаленні. При малих значеннях, навпаки, видимий розмір об'єктів слабо залежить від відстані.
Near & Far	Визначає мінімальну і максимальну відстані від камери, яка потрапляє в рендеринг. Дуже далекі точки не будуть вимальовуватися взагалі, як і точки, які знаходяться надто близько.

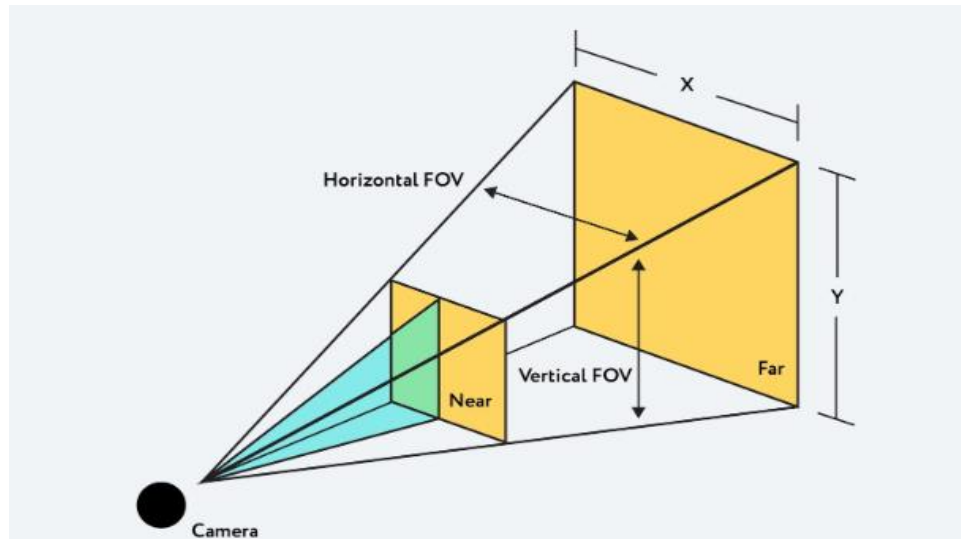
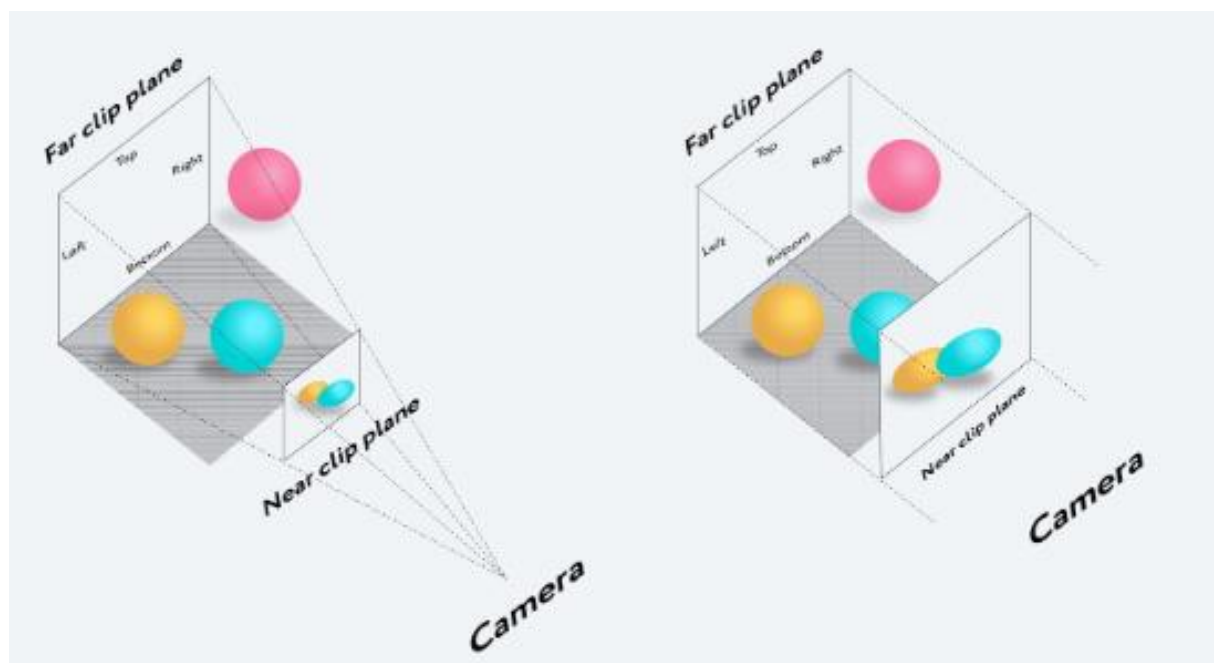


Рисунок 2.2 – Ілюстрація роботи перспективної камери

В режимі проектування Orthographic Camera (рис. 2.3.б) розмір об'єкта на відображеному зображенні залишається постійним, незалежно від його відстані від камери. Тобто, це камера, віддалена на нескінченну відстань від об'єктів. Усі перпендикулярні прямі залишаються перпендикулярними, а всі

паралельні – паралельними. Якщо рухати камеру, прямі і об’єкти не будуть спотворюватися. Це може бути корисним при відображенні 2D-сцен і елементів UI.



а)

б)

Рисунок 2.3 – Порівняння камер: а – перспективна; б – ортографічна

Важливим інструментом є освітлення (рис. 2.4), без якого на сцені буде складатися враження, що глядач (користувач) знаходиться в темній кімнаті. Крім цього, за допомогою освітлення можна надати більшу реалістичність сцені. Технічно кожному освітленню можна задати колір. Приклади освітлення наведені у таблиці 2.2.

Таблиця 2.2 – Типи освітлення

Тип освітлення	Опис
Ambient Light	Фонове освітлення, яке використовується для освітлення всіх об’єктів сцени однаково; не може використовуватися для створення тіней, оскільки не має напрямку.

## Продовження таблиці 2.2 – Типи освітлення

Тип освітлення	Опис
Directional Light	Світло, що випромінюється в певному напрямку. Це світло буде вести себе так, ніби воно нескінченно далеко. Воно може відкидати тіні, оскільки направлене на конкретний об'єкт.
Point Light	Світло, що випромінюється з однієї точки в усіх напрямках. Звичайний випадок використання такого освітлення це повторення освітлення від простої лампочки.
Spot Light	Світло, яке випромінюється з однієї точки в одному напрямку вздовж конуса, що розширюється по мірі віддалення від джерела світла.

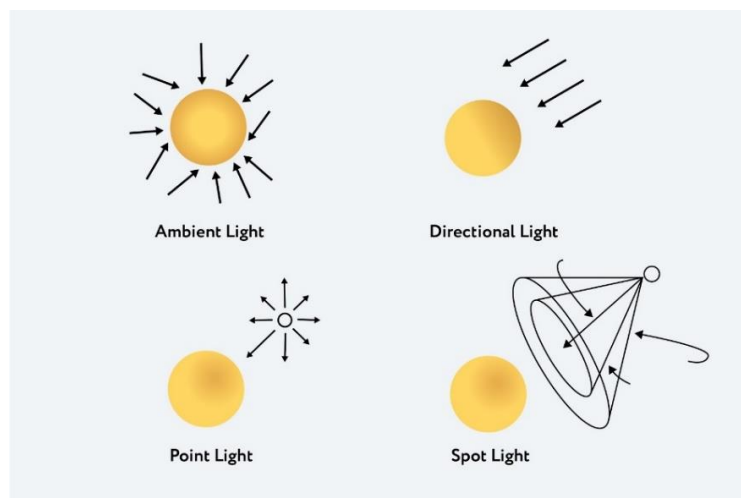


Рисунок 2.4 – Приклади освітлень

Тепер перейдемо до процесу створення об'єктів на сцені. Об'єкт, що створюється на сцені, називається Mesh. Цей клас приймає два аргументи:

- Geometry, який описує форму (положення вершин, межі, радіус та інші параметри)(рис. 2.5);
- Material, який описує зовнішній вигляд об'єктів (колір, текстура, прозорість іта інші характеристики.) (рис. 2.6).

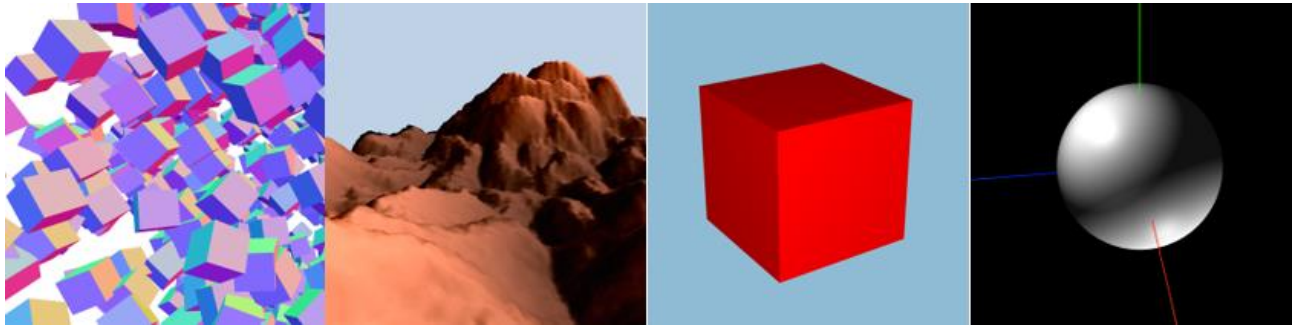


Рисунок 2.5 – Geometry приклади із офіційного сайту Three.js

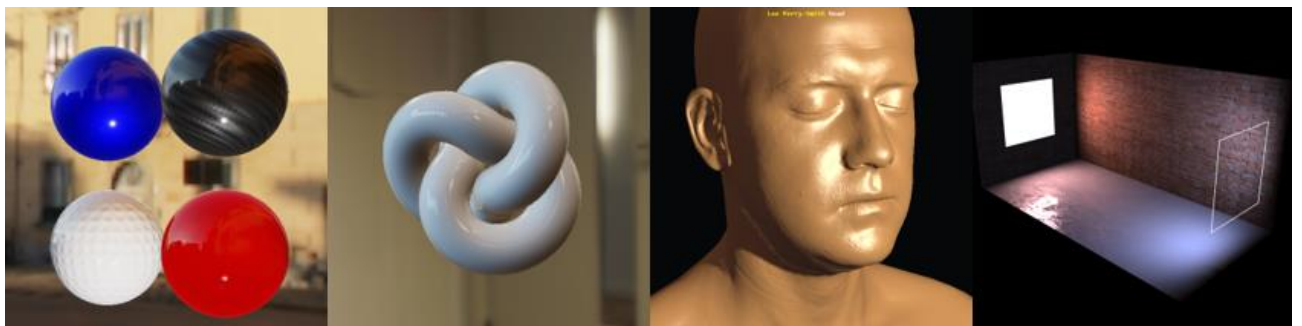


Рисунок 2.6 – Material приклади із офіційного сайту Three.js

Для наочності розглянемо процес створення засобами Three.js найпростіших фігур – куба та сфери.

Насамперед, потрібно перейти на офіційний сайт бібліотеки <https://threejs.org/> та завантажити її останню версію. Потім необхідно підключити бібліотеку в секції head або на початку секції body документа у проєкті:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>First Three.js app</title>
    <style>
      body { margin: 0; }
      canvas { width: 100%; height: 100% }
    </style>
  </head>
  <body>
    <script src="js/three.js"></script>
    <script>
      // Тут увесь Javascript код.
    </script>
  </body>
</html>

```

Після цього, аби відобразити створюваний об'єкт, треба створити сцену, додати камеру і налаштувати рендер. Додаємо сцену:

```
var scene = new THREE.Scene();
```

Додаємо перспективну камеру:

```
var camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 );
```

Камера приймає на себе чотири параметри, про які було згадано раніше:

- кут зору, або FOV (у нашому випадку, це стандартний кут 75°);
- співвідношення сторін, або aspect ratio;
- третім і четвертим параметром будуть мінімальна і максимальна відстань від камери, яка потрапить в рендеринг.

Додаємо і налаштовуємо рендер:

```
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
```

Отже, підсумовуючи зроблене: спочатку було створено об'єкт рендера, потім встановлено його розмір, відповідно до розміру видимої області і, зрештою, додано його на сторінку для створення порожнього елемента canvas, з яким будемо працювати. Після створення рендера вказуємо, де потрібно відобразити тег canvas. У нашому випадку ми додали його в тег body.

Для створення самого куба спочатку задаємо геометрію:

```
var geometry = new THREE.BoxGeometry( 10, 10, 10);
```

Куб створюється за допомогою класу `BoxGeometry`. Це клас, який містить у собі вершини і межі куба. Передаємо розміри:

- `width`, ширина куба, розмір сторін по осі X;
- `height`, висота куба, розмір сторін по осі Y;
- `depth`, глибина куба, розмір сторін по осі Z.

Щоб розфарбувати куб, задаємо матеріал:

```
var material = new THREE.MeshBasicMaterial( { color: 0x00ff00 } );
```

У нашому випадку заданий `MeshBasicMaterial` і переданий параметр кольору `0x00ff00`, тобто, зелений колір. Цей матеріал використовується для надання фігурі однорідного кольору. Недолік його в тому, що у фігурі зникає глибина. Але цей матеріал може стати в нагоді при відображенні каркасів за допомогою параметра `{wireframe: true}`.

Тепер нам потрібен об'єкт `Mesh`, який приймає геометрію, і застосовує до нього матеріал:

```
var cube = new THREE.Mesh( geometry, material );
scene.add( cube );
```

```
camera.position.z = 25;
```

Додаємо `Mesh` на сцену і віддаляємо камеру, оскільки всі об'єкти після методу `scene.add()` за замовчуванням додаються з координатами `(0,0,0)`, через те, що камера і куб будуть в одній точці.

Для того щоб анімувати куб, потрібно відмалювати все всередині циклу рендеринга, використовуючи `requestAnimationFrame`:

```
function render() {
  requestAnimationFrame( render );
  cube.rotation.x += 0.01;
  cube.rotation.y += 0.01;
  renderer.render( scene, camera );
}
render();
```

`requestAnimationFrame` – це запит до браузера про те, що ви хочете щось анімувати. Йому передається функція для виклику, тобто функція `render()`.

Тут же задаються параметри швидкості обертання. В результаті, цикл рендерить створену сцену 60 раз в секунду і змушує куб обертатися (рис. 2.7).

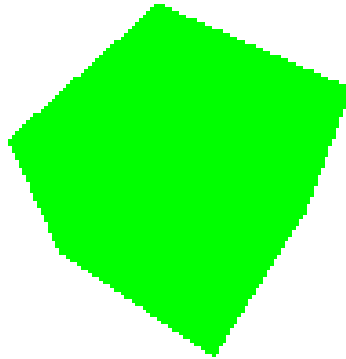


Рисунок 2.7 – Куб

Тепер намалюємо сферу.

```
var geometry = new THREE.SphereGeometry(1, 32, 32);
```

Для побудови сфери використовується клас SphereGeometry, який приймає на себе:

- радіус (значення за замовчуванням дорівнює 1);
- widthSegments, кількість горизонтальних сегментів (трикутників, мінімальним значенням є 3, значення за замовчуванням – 8);
- heightSegments, кількість вертикальних сегментів (мінімальним значенням є 2, значення за замовчуванням – 6).

Чим більша кількість трикутників, тим гладкішою буде поверхня сфери.

Далі спробуємо використовувати інший матеріал – MeshNormalMaterial, різнобарвний матеріал, який зіставляє вектори нормалей в RGB кольори:

```
var material = new THREE.MeshNormalMaterial();
var sphere = new THREE.Mesh( geometry, material );
scene.add( sphere );
camera.position.z = 3;
```

Видів цього матеріалу існує дуже багато. Деякі з них можна поєднувати і застосовувати одночасно до однієї фігури.

Останнім кроком задаємо цикл рендеринга:

```
function render() {  
  requestAnimationFrame( render );  
  sphere.rotation.x += 0.01;  
  sphere.rotation.y += 0.01;  
  renderer.render( scene, camera );  
}  
render();
```

І отримуємо сферу (рис. 2.8) [14]:



Рисунок 2.8 – Сфера

### 2.2.2 Фреймворк A-Frame

A-Frame — це вебфреймворк з відкритим кодом для створення 3D-графіки та віртуальної реальності в Інтернеті (рис. 2.9). Він був розроблений Mozilla в 2015 році і базується на HTML, CSS і JavaScript. A-Frame використовує декларативний синтаксис, який дозволяє розробникам створювати складні 3D-сцени за допомогою простих тегів HTML.

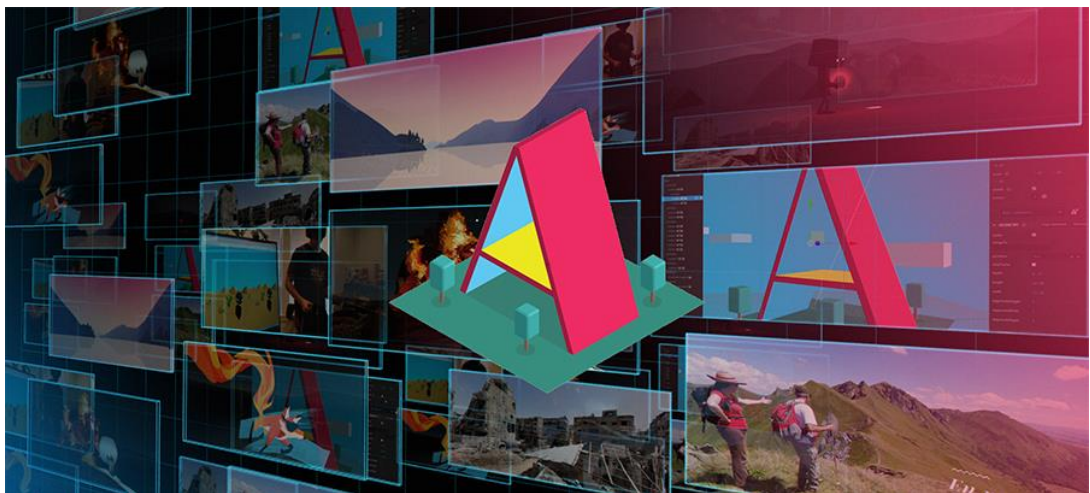


Рисунок 2.9 – Приклади роботи фреймворку A-Frame

A-Frame створено на основі Three.js. Він забезпечує абстракцію вищого рівня для роботи з Three.js, дозволяючи розробникам створювати 3D-сцени з меншою кількістю коду та спрощуючи цей процес. A-Frame також підтримує широкий спектр платформ віртуальної реальності включаючи Oculus Rift, HTC Vive, Google Cardboard і Samsung Gear VR та інші.

Важливою особливістю A-Frame є система компонентів. Компоненти — це багаторазові модулі, які можна приєднати до компонентів сцени, таких як об'єкти, світло, камери тощо. A-Frame постачається з набором вбудованих компонентів: положення, обертання, масштаб, матеріал і текстура тощо. Розробники можуть створювати власні спеціальні компоненти для розширення функціональності A-Frame.

A-Frame надає ряд примітивів, які є базовими формами, які використовуються для створення більш складних об'єктів. Ці примітиви містять такі елементи як ящик, сфера, циліндр, площина тощо. При розробці власного проєкту також можна імпортувати 3D-моделі, створені в іншому програмному забезпеченні, наприклад Blender або Maya, і використовувати їх у своїх сценах A-Frame.

A-Frame надає підтримку WebVR, стандарту для створення досвіду віртуальної реальності в Інтернеті. A-Frame містить в собі набір утиліт і компонентів для роботи з WebVR, серед них – елементи керування камерою, обробку введення тощо [15].

### **2.2.3 Фреймворк Babylon.js**

Babylon.js — це ще один популярний фреймворк JavaScript із відкритим кодом для створення 3D-графіки в Інтернеті (рис. 2.10). Він був створений Девідом Катухе і зараз підтримується командою розробників з Microsoft. Головна мета створення Babylon.js – реалізація простого та гнучкого у використанні фреймворку, із широким набором функцій для створення захоплюючих 3D-досвідів.



Рисунок 2.10 – Приклад роботи фреймворку Babylon.js

Babylon.js створено на основі WebGL, вебстандарту для створення 3D-графіки в браузері. Він надає ряд інструментів і утиліт для роботи з WebGL, включаючи графік сцени, систему матеріалів і потужний механізм анімації. Фреймворк підтримує широкий спектр імпортерів для 3D-моделей для роботи з популярними форматами, такими як .obj, .fbx і .glTF.

Фреймворк надає простий та інтуїтивно зрозумілий API для створення 3D-об'єктів у сцені та керування ними. Він містить ряд вбудованих матеріалів і текстур, що дозволяє легко створювати реалістичні об'єкти з мінімальними зусиллями.

Babylon.js надає інструментарій для інтерактивної роботи, такої як фізичне моделювання, системи частинок і ефекти постобробки. Він включає підтримку віртуальної реальності і доповненої реальності, що робить його популярним вибором для створення ефекту занурення в Інтернет.

Babylon.js має ще одну характеристику – високу продуктивність. Фреймворк розроблений з метою високої оптимізації та підтримки функцій створення екземплярів, що дозволяє відтворювати велику кількість об'єктів з мінімальними витратами. Оптимізатор мережі та компресор текстур забезпечують безперебійну роботу 3D-сцен на будь-яких пристроях [16].

#### 2.2.4 Рушій PlayCanvas

PlayCanvas — це ігровий механізм і платформа для створення 3D-графіки та інтерактивних програм у браузері. Він був створений командою

розробників із Лондона, Великобританія, і побудований на основі WebGL, веб-стандарту для створення 3D-графіки в браузері. PlayCanvas надає ряд інструментів і утиліт для роботи з WebGL, включаючи потужний редактор сцен, фізичний механізм і систему сценаріїв на основі JavaScript.

Платформа надає низку шаблонів і прикладів, які допомагають на початку роботи з ним, а також повну систему документації, яка охоплює всі аспекти механізму та платформи. Редактор сцен дозволяє створювати 3D-об'єкти та керувати ними візуально, не потребуючи жодних знань програмування. Також можна додати фізику та іншу поведінку до об'єктів за допомогою простої мови сценаріїв на основі JavaScript.

PlayCanvas цінується високою продуктивністю. Двигун розроблено для високої оптимізації з підтримкою таких функцій, як створення екземплярів і пакетне формування, що дозволяє відтворювати велику кількість об'єктів з мінімальними витратами. Він містить набір інструментів оптимізації, таких як компресор текстур і оптимізатор сітки, щоб забезпечити безперебійну роботу 3D-сцен на різних пристроях.

PlayCanvas пропонує функціонал, за допомогою якого можна підтримувати роботу пристроїв віртуальної реальності і доповненої реальності, а також він має вбудовану мережеву систему для кількох гравців. Платформа підтримує різні типи пристроїв введення: сенсорні екрани, ігрові контролери та контролери VR/AR, що робить її універсальним вибором для створення широкого спектру 3D-додатків та ігор.

Ще одним важливим аспектом PlayCanvas є його розширюваність. Платформа створена з використанням відкритих вебтехнологій (HTML5 і JavaScript), що означає, що її можна легко налаштувати та розширити за допомогою сторонніх бібліотек і інструментів. Платформа надає API для інтеграції із зовнішніми системами, наприклад, з платформами соціальних медіа та онлайн-сервісів [17].

## РОЗДІЛ 3 ЗАГАЛЬНИЙ ОГЛЯД ПОНЯТТЯ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

### 3.1 Історія розвитку доповненої реальності

Хоча сьогодні доповнена реальність є одним із флагманів сучасних технологій, нові функції постійно збагачують її використання, історія доповненої реальності набагато старша, ніж здається.

У 1960-х роках з'явився новий проєкт занурення в відео – Sensorama (рис. 3.1). Винайдений Мортонем Хейлігом у 1950-х роках, він мав звертатися до п'яти почуттів (звідси й назва), щоб залучити глядача до фільму на екрані. Прототип був завершений у 1962 році і включав кольоровий екран, вентилятори, випромінювачі запахів, стереозвукову систему та рухоме крісло. Ці різні елементи активувалися відповідно до фільму, який проєктується на екрані. Ця концепція ближча до віртуальної реальності (симуляції фізичної присутності в штучному середовищі), але це перший підхід до AR.

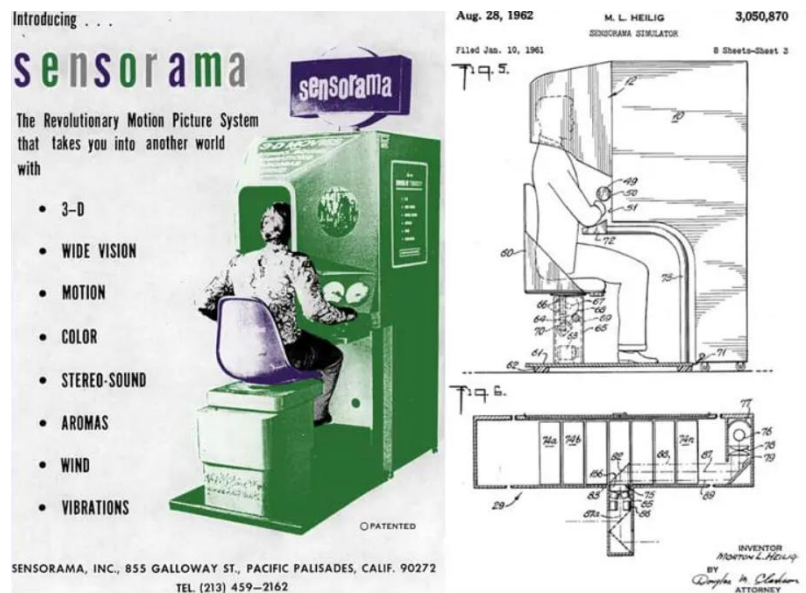


Рисунок 3.1 – Sensorama: перший проєкт доповненої реальності

Американського інженера Івана Сазерленда часто визнають одним із піонерів в історії доповненої реальності. Фактично, у 1960-х роках він розробив програмне забезпечення для тривимірного моделювання та візуального моделювання Sketchpad.

У 1968 році в Університеті Солт-Лейк-Сіті в штаті Юта, США, було розгорнуто пристрій «Тривимірний дисплей, який встановлюється на голову» – пара окулярів для перегляду зображень у 3D. Це були справжні предки Google Glass. Ця інсталяція була настільки важкою і вражаючою, що її прозвали «Демокловим мечем». Його потрібно було підвісити до стелі, щоб витримати його вагу, а користувачів довелося прив'язати до пристрою для кращого занурення, що зробило досвід досить незручним.

У 1980 році Стів Манн розробив EyeTap, шолом, який відображає віртуальну інформацію перед очима користувача. Це перша функціональна модель гарнітури доповненої реальності. EyeTap існує і сьогодні: його відточували, поки він не став таким же стриманим і мінімалістичним, як прості окуляри.

Протягом наступних десятиліть досягнення в історії доповненої реальності здебільшого обслуговували конкретні сектори, такі як авіація, військова оборона або промисловість. Хоча AR додали до кількох пристроїв у 2000-х роках, насправді доповнена реальність стала головною інновацією у рекламі в 2010-х [18].

У 2012 році Google запустив тест Google Glass, перший продукт AR, який продається у великих масштабах і робить цю технологію доступною для всіх. Однак цей продукт не зустрів свою аудиторію і знаменує собою один з найбільших провалів в історії доповненої реальності. Google швидко припинив виробництво цих окулярів на початку 2015 року. У 2019 році стався ще один поворот у пригоді Google Glass, веб-гігант вирішив відновити виробництво своїх окулярів, але цього разу тільки для професіоналів.

Snapchat знаменує собою поворотний момент в історії доповненої реальності. У соціальних мережах Snapchat демократизував AR для всіх

користувачів, іноді навіть не підозрюючи про це. Додаток вперше представив «геофільтри» у 2014 році, деякі фільтри, які змінювалися залежно від того, де ви були.

У 2015 році в соцмережі з'явилися «Об'єктиви». Ця функція аналізує обличчя користувача з його передньої камери та виводить на екран різні графічні елементи, якими потім можна поділитися на фотографіях та відео. Цей варіант зробив революцію у використанні соціальних мереж, і Instagram, Messenger або WhatsApp швидко скористалися цією новинкою, щоб конкурувати за кількістю користувачів.

Мобільний додаток «Pokemon GO!», випущений у 2016 році, поставив доповнену реальність в основу свого ігрового процесу. Гравці блукають у своєму реальному оточенні зі телефоном, щоб зловити покемонів або перемогти інших гравців під час бійок. Хоча в 2022 році гра вже не в топі завантажень, вона залишається дуже популярною і продовжує заробляти багато грошей для Niantic, компанії, яка розробила гру. Згодом на основі AR було розроблено ряд інших ігор: «Harry Potter: Wizards Unit», «Angry Birds AR: Isle of Pigs», «The Walking Dead: Our World» [19].

### **3.2 Як працює доповнена реальність**

Доповнена реальність — це технологія, яка накладає цифровий вміст на реальний світ, покращуючи при цьому користувацький досвід.

Доповнена реальність починається з пристрою, оснащеного камерою, наприклад, смартфона, планшета чи розумних окулярів, із завантаженим програмним забезпеченням AR. Коли користувач наводить на пристрій і дивиться на об'єкт, програмне забезпечення розпізнає його за допомогою технології комп'ютерного зору, яка аналізує відеопотік.

Потім пристрій завантажує інформацію про об'єкт з хмари, майже так само, як веббраузер завантажує сторінку через URL-адресу. Принципова відмінність полягає в тому, що інформація AR представлена у вигляді 3D

«досвіду», накладеного на об'єкт, а не в 2D-сторінці на екрані. Отже, те, що бачить користувач, є частково реальним, а частково цифровим.

AR може надавати уявлення про дані в реальному часі, що надходять від продуктів, і дозволяти користувачам керувати ними за допомогою сенсорного екрана, голосу або жесту. Наприклад, користувач може торкнутися кнопки зупинки на цифровому графічному накладенні в доповненій реальності або просто вимовити слово «стоп», щоб надіслати команду продукту через хмару. Оператор, який використовує гарнітуру AR для взаємодії з промисловим роботом, може побачити накладені дані про продуктивність робота та отримати доступ до його елементів керування.

Коли користувач рухається, розмір і орієнтація дисплея AR автоматично підлаштовуються під контекст, що змінюється. Нова графічна або текстова інформація потрапляє в поле зору, тоді як інша інформація виходить з поля зору. У промислових умовах користувачі в різних ролях, наприклад, оператор машин і технік з технічного обслуговування, можуть дивитися на один і той же об'єкт, але отримувати різні можливості AR, які відповідають їхнім потребам.

Тривимірний цифровий об'єкт, який знаходиться в хмарі – «цифровий близнюк» об'єкта, – служить мостом між реальним об'єктом і AR. Ця модель створюється або за допомогою комп'ютерного проектування, як правило, під час розробки продукту, або за допомогою технології, яка оцифровує фізичні об'єкти. Потім близнюк збирає інформацію з продукту, бізнес-систем та зовнішніх джерел, щоб відобразити поточну реальність продукту. Це засіб, за допомогою якого програмне забезпечення AR точно розміщує та масштабує актуальну інформацію про об'єкт [20].

### **3.3 Ключові технології доповненої реальності**

Технологія інтелектуального відображення, технологія тривимірної реєстрації та технологія інтелектуальної взаємодії становлять основне технологічне коло AR і відіграють важливу роль у його розвитку.

#### **3.3.1 Технологія інтелектуального дисплея**

Доповнена реальність базується на здатності проектувати цифровий вміст на поверхні реального світу, а для цього потрібні складні технології дисплея, які можуть точно відстежувати позицію та просторову орієнтацію користувача. Технологія інтелектуального відображення містить у собі ряд апаратних і програмних компонентів, які, працюючи разом, створюють ефект доповненої реальності.

Одним із ключових компонентів технології інтелектуального дисплея є сам дисплей. AR-дисплеї повинні мати можливість проектувати цифровий вміст на поверхні реального світу з високою точністю та роздільною здатністю. Для цього потрібні спеціальні дисплеї (наприклад, дисплеї OLED і microLED), які можуть створювати висококонтрастні зображення з низькою затримкою та високою частотою оновлення.

Іншим важливим компонентом технології інтелектуального дисплея є система відстеження. AR вимагає точного відстеження позиції та орієнтації у просторі користувача, щоб для коректного проектування цифрового вмісту на поверхні реального світу. Для підтримки цієї вимоги використовується комбінація датчиків, таких як камери, акселерометри та гіроскопи, які можуть відстежувати рухи користувача в режимі реального часу.

Технологія інтелектуального відображення включає складні алгоритми та програмне забезпечення, яке здатне обробляти дані датчика та створювати точну модель оточення користувача. Використовується одночасна локалізація та відображення (SLAM), яка будує 3D-карти навколишнього середовища на

основі даних датчиків, а також алгоритми комп'ютерного зору, які можуть розпізнавати та відстежувати об'єкти, що перебувають в полі зору користувача.

Щоб створити безперебійну роботу технології доповненої реальності, технологія інтелектуального дисплея повинна адаптуватися до мінливих умов освітлення та інших факторів навколишнього середовища. Для використовуються передові методи обробки зображень: зображення з широким динамічним діапазоном (HDR) і корекція кольорів, які налаштовують цифровий вміст відповідно до реального середовища.

Однією з найбільших проблем у розробці інтелектуальних технологій відображення є необхідність балансу продуктивності та енергоспоживання. Для доповненої реальності потрібне високопродуктивне апаратне та програмне забезпечення, яке може обробляти великі обсяги даних у режимі реального часу. В той же час, обране програмне забезпечення повинно балансувати з необхідністю мінімізувати енергоспоживання та подовжити термін служби акумулятора.

Незважаючи на ці проблеми, технологія інтелектуального дисплея досягла значних успіхів за останні роки завдяки прогресу в апаратних і програмних можливостях [21].

### **3.3.2 Технологія 3D-реєстрації**

Як одна з найважливіших технологій у системі доповненої реальності, технологія тривимірної реєстрації дозволяє точно накладати віртуальні зображення в реальне середовище. Основний потік технології 3D-реєстрації складається з двох етапів.

Спочатку визначається взаємозв'язок між віртуальним зображенням, моделлю та інформацією про напрямок і положення камери або пристрою відображення.

Після цього віртуальне відтворене зображення і модель точно проєктуються в реальне середовище. В результаті, їх можна об'єднати з реальним середовищем. Існують різні способи тривимірної реєстрації, такі як технологія реєстрації на основі апаратного трекера, технологія тривимірної реєстрації на основі комп'ютерного зору, технологія тривимірної реєстрації на основі бездротової мережі та змішана технологія реєстрації, серед яких дві перші найпопулярніші. Для технології тривимірної реєстрації, заснованої на комп'ютерному баченні, вона встановлює контрольну точку для визначення напрямку та положення реальної сцени камерою або дисплеєм.

### **3.3.3 Технологія інтелектуальної взаємодії**

Інтелектуальна інтерактивна технологія тісно пов'язана з технологією інтелектуального дисплея, технологією тривимірної реєстрації, ергономікою, когнітивною психологією та іншими дисциплінами. У системах доповненої реальності існує безліч інтелектуальних взаємодій, включаючи взаємодію з апаратним пристроєм, взаємодію з розташуванням, а також взаємодії на основі тегів або інших інформаційних взаємодій.

З розвитком технології інтелектуальної взаємодії доповнена реальність не тільки накладає віртуальну інформацію на реальні сцени, але й реалізує взаємодію між людьми та віртуальними об'єктами в реальних сценах. Ця взаємодія заснована на тому факті, що люди дають конкретні інструкції віртуальному об'єкту в сцені, і віртуальний об'єкт може зробити певний зворотний зв'язок, таким чином дозволяючи аудиторії програми доповненої реальності досягти кращого досвіду.

## **3.4 Види доповненої реальності**

Загалом, можна виокремити 6 видів доповненої реальності. Розглянемо детальніше кожен з них.

### 3.4.1 Доповнена реальність на основі маркерів

Доповнена реальність (AR) на основі маркерів — це популярна техніка, яка використовує попередньо визначені маркери або зображення для запуску цифрового вмісту в реальному середовищі.

Основний принцип AR на основі маркерів полягає у використанні камери для виявлення попередньо визначеного маркера або зображення, яке потім ініціює відображення цифрового вмісту. Маркер служить точкою відліку або прив'язкою в реальному світі, дозволяючи системі AR накладати віртуальні об'єкти або інформацію таким чином, що вони фізично розташовувалися в тому самому просторі, що й маркер.

Щоб досягти цього ефекту, системи AR на основі маркерів зазвичай використовують алгоритми комп'ютерного зору для виявлення та відстеження маркера в режимі реального часу. Це дозволяє додатку AR постійно оновлювати положення та орієнтацію віртуального вмісту щодо маркера, створюючи ілюзію прямої інтеграції між фізичним і цифровим світами.

Однією з переваг AR на основі маркерів є його відносна простота та легкість використання. За рахунок того, що маркер служить орієнтиром, користувачам легко зрозуміти, як працює система і як взаємодіяти з віртуальним контентом. Це призвело до широкого використання AR на основі маркерів у сферах ігор, реклами та освіти.

Ще однією перевагою AR на основі маркерів є його високий рівень точності. Оскільки маркер визначений заздалегідь, система доволі легко його розпізнати. Цим досягається високий рівень точності відстеження та мінімізуються помилки відстеження.

Одним із обмежень AR на основі маркерів є його залежність від попередньо визначених маркерів або зображень. Система AR може розпізнавати та відстежувати лише ті об'єкти, які були спеціально розроблені для цієї мети, обмежуючи діапазон потенційних застосувань. Крім того, AR на

основі маркерів може бути чутливим до змін освітлення та навколишнього середовища, що теж впливає на точність відстеження.

### **3.4.2 Безмаркерна доповнена реальність**

Доповнена реальність без позначок — це техніка, яка дозволяє відображати віртуальні об'єкти або інформацію в реальному середовищі без необхідності використання попередньо визначених маркерів або зображень.

Основний принцип доповненої реальності без позначок полягає у використанні алгоритмів комп'ютерного зору для аналізу реального середовища та визначення об'єктів або функцій, які можуть служити опорними точками для віртуального вмісту. Цими об'єктами можуть бути стіни, підлога та меблі, а також природні елементи, такі як кути, краї та візерунки текстури.

Для підтримки коректної роботи, системи AR без позначок зазвичай використовують комбінацію відстеження на основі камери та відстеження на основі датчиків для виявлення та моніторингу положення та орієнтації об'єктів реального світу.

Однією з переваг доповненої реальності без позначок є її здатність працювати з широким діапазоном середовищ реального світу без необхідності використання попередньо визначених маркерів або зображень, що забезпечує більшу гнучкість і універсальність у створенні доповненої реальності, а також потенціал для більш природної та інтуїтивно зрозумілої взаємодії між користувачем і віртуальним вмістом.

Ще однією перевагою AR без позначок є його потенціал для підвищення точності. Використовуючи об'єкти реального світу як контрольні точки, AR без позначок може досягти більш високого рівня точності відстеження та мінімізувати помилки, особливо в складних або динамічних середовищах.

Проблемою доповненої реальності без позначок є потреба в передових алгоритмах комп'ютерного зору та потужності обробки.

### **3.4.3 Доповнена реальність на основі розташування**

Доповнена реальність на основі розташування – це тип AR, який використовує географічне розташування користувача як основу для створення віртуального вмісту. Використовуючи GPS, Wi-Fi або інші технології визначення місцезнаходження, доповнена реальність на основі місцезнаходження може накладати віртуальну інформацію на середовище реального світу залежно від поточного місцезнаходження користувача.

Важливою рисою доповненої реальності на основі визначення місця розташування є її здатність створювати персоналізовані та контекстуальні враження. Використовуючи місцезнаходження користувача як основу для віртуального вмісту, доповнена реальність на основі місцезнаходження може надавати інформацію, розваги чи інший вміст, який адаптований до конкретних інтересів або потреб користувача.

Доповнену реальність на основі розташування можна використовувати для навігації, задаючи користувачам напрямки в реальному часі, вказуючи визначні місця та іншу інформацію під час навігації фізичним простором.

Однак існують також проблеми із запровадженням доповненої реальності на основі місцезнаходження, як-от забезпечення точних даних про місцезнаходження, керування проблемами конфіденційності даних і розробка контенту, який буде цікавим і корисним для користувачів.

### **3.4.4 Накладання доповненої реальності**

Накладання AR створює альтернативний вигляд об'єкта і може використовуватися для часткової або повної заміни вихідного вигляду об'єкта. Це означає, що ця технологія замінює або весь вигляд об'єкта, або його частину на розширений вигляд.

У цьому типі доповненої реальності розпізнавання об'єктів є надзвичайно важливим. Якщо програма не зможе визначити, що вона

дивиться, вона не зможе замінити оригінальний вигляд віртуальним. Накладання AR забезпечує кілька переглядів цільового об'єкта з можливістю виділення додаткової релевантної інформації про цей об'єкт.

### **3.4.5 Доповнена реальність на основі проєкції**

AR на основі проєкцій – це тип технології доповненої реальності, яка використовує проєктори для відображення цифрового вмісту на фізичних поверхнях, таких як стіни, підлога або столи. Цифровий вміст зазвичай накладається на зображення камери фізичного середовища, що дозволяє користувачеві бачити як реальний світ, так і віртуальні об'єкти. Віртуальні об'єкти можуть бути інтерактивними та реагувати на поведінку користувача, наприклад, жести рук або фізичні об'єкти, розміщені на поверхні.

Однією з ключових переваг доповненої реальності на основі проєкцій є її здатність створювати масштабне та захоплююче середовище. AR на основі проєкцій можна використовувати для створення інтерактивних виставок, інсталяцій або захоплюючих вистав, які дозволяють користувачам досліджувати віртуальні об'єкти у фізичному просторі та взаємодіяти з ними. Цю технологію також можна використовувати в роздрібних магазинах для створення інтерактивних дисплеїв або в музеях для надання додаткового контексту чи інформації про експонати.

### **3.4.6 Окреслення доповненої реальності**

Окреслення AR – це тип доповненої реальності, який накладає цифровий контур на фізичний світ, підкреслюючи лінії та межі, які можуть бути невидимі для людського ока. Ця технологія дозволяє користувачам візуалізувати приховані або складні структури та взаємодіяти з ними більш інтуїтивно зрозумілим способом.

Окреслення AR працює за допомогою комп'ютерного зору та алгоритмів розпізнавання об'єктів для визначення ліній і меж у реальному часі. Технологія може розпізнавати форму та положення об'єктів у фізичному середовищі, а потім накладати на них цифровий контур. Це накладання може надати користувачам додаткову інформацію про об'єкт, наприклад, його розміри або положення його компонентів.

Одне із застосувань AR-контур – це промислове обслуговування та ремонт. За допомогою AR-контур технічні спеціалісти можуть накладати цифровий контур на машину чи іншу складну структуру, висвітлюючи розташування компонентів і полегшуючи виявлення та діагностику проблем. Ця технологія може заощадити час і підвищити точність процесу технічного обслуговування та ремонту, підвищити продуктивність і скоротити час простою [22].

### **3.5 Різниця між доповненою та віртуальною реальністю**

Доповнена реальність (AR) і віртуальна реальність (VR) — це технології, які покращують або імітують реальність, але роблять це принципово різними способами.

AR додає цифрові елементи в реальний світ, накладаючи їх на фізичне середовище користувача. Ця технологія зазвичай передбачає використання смартфона, планшета або гарнітури AR, які використовують камери для зйомки реального світу, а потім додають до нього цифрову інформацію. AR можна використовувати для покращення користувацького в межах реального світу, для надання додаткової інформації про середовище та створення нових форм взаємодії між користувачами та цифровими об'єктами.

VR, з іншого боку, повністю занурює користувача в цифровий світ, часто використовуючи гарнітуру або інше спеціалізоване обладнання для створення імітованого середовища. У VR користувачі можуть взаємодіяти з

цифровими об'єктами, пересуватися у віртуальному просторі та переживати події та середовища, які неможливо відчути в реальному світі.

Хоча AR і VR мають певну схожість, між цими двома технологіями є кілька ключових відмінностей. Однією з основних відмінностей є рівень занурення. AR додає цифрові елементи до реального світу, тоді як VR створює абсолютно новий цифровий світ, який користувач може досліджувати. Доповнена реальність зазвичай дає менш відчутній ефект, ніж віртуальна реальність, оскільки користувач усе ще усвідомлює своє фізичне оточення.

Ще одна відмінність — рівень взаємодії. Доповнена реальність часто передбачає прості дії, такі як торкання екрану смартфона або проведення пальцем по екрану смартфона, тоді як VR працює із більш складними командами, такими як фізичні рухи та жести. У VR користувачі можуть взаємодіяти з об'єктами та середовищем більш природним та інтуїтивно зрозумілим способом.

AR і VR також відрізняються за сферою їх застосування. AR часто використовується в таких галузях, як роздрібна торгівля, реклама та ігри, де вона може покращити реальний досвід і надати нові форми взаємодії з цифровим контентом. VR, з іншого боку, часто використовується в таких сферах, як освіта, тренінги та симуляція, де створює середовище для навчання та практики нових умінь.

Нарешті, існують відмінності в апаратному забезпеченні, необхідному для роботи для AR і VR. Дію AR часто можна випробувати за допомогою смартфона або планшета, тоді як VR зазвичай вимагає більш спеціалізованого обладнання, наприклад, гарнітури або спеціальної системи VR.

Загалом, хоча AR і VR мають певну схожість, це принципово різні технології, які пропонують користувачам унікальний і відмінний досвід [23].

### 3.6 Головні тренди доповненої реальності

У цьому підрозділі розглянуто найбільш популярні тренди доповненої реальності.

Мобільна доповнена реальність. Мобільна доповнена реальність має багато очевидних переваг, які роблять її центром уваги багатьох технологій. Велика кількість користувачів уже мають смартфони з підтримкою AR, тому їм не потрібно докладати ніяких зусиль, щоб придбати дорогу та громіздку гарнітуру. Мобільні пристрої за своєю природою дуже портативні і їх легко застосовувати практично в будь-яку просторі.

Підтримка внутрішньої та зовнішньої навігації. У 2022 році AR-навігація стала більш плавною та досяжно. Розвиток таких технологій як антени BLE (Bluetooth Low Energy), Wi-Fi RTT (Round-Trip Time) та UWB (Ultra-Wide Band), робить навігацію в приміщенні більш ефективною, ніж у попередні роки. Одним з найкорисніших застосувань цієї технології є відображення напрямків AR у великих приміщеннях, таких як розподільні центри, торгові центри та аеропорти.

Охорона здоров'я та доповнена реальність. За даними багатьох досліджень, доповнена реальність і штучний інтелект трансформують традиційну бізнес-модель охорони здоров'я, пропонуючи рішення з підтримкою AR/MR і діагностичні інструменти на основі ШІ. Наприклад, Microsoft HoloLens 2 може надавати інформацію хірургу, дозволяючи йому використовувати обидві руки під час процедури.

У зв'язку з продовженням обмежень, пов'язаних із Covid-19, використання технологій доповненої реальності стає все більш важливим для вирішення таких проблем, як складність віддаленої підтримки пацієнтів та збільшення навантаження на лікарні. Це включає як рішення для телехірургічних операцій, так і програми для психічного здоров'я, які допомагають людям підтримувати психологічну рівновагу.

У поєднанні з алгоритмами машинного навчання, технологія AR може стати ефективним варіантом виявлення захворювань. Ще у 2020 році Google оголосила про розробку мікроскопа на основі AR для Міністерства оборони США, щоб підвищити точність діагностики та лікування раку. Такий гібридний пристрій використовує камеру для зйомки зображень в режимі реального часу, які потім обробляються за допомогою комп'ютерної діагностики для негайного відображення результатів і діагностики захворювань на ранній стадії.

Покупки з доповненою реальністю. Доповнена реальність має численні можливості в секторі роздрібною торгівлі. Однією з таких можливостей, яка зараз реалізується, є картування магазинів. Завдяки доповненій реальності вказівки на екрані можуть допомогти користувачеві орієнтуватися в макеті магазину, щоб знайти потрібний товар. Dent Reality – одна з компаній, що розробляють системи AR-карт для магазинів. Все, що користувач повинен зробити, це підняти свій телефон і слідувати вказівкам доповненої реальності через проходи до місця призначення: предмета, який він шукає.

Доповнена реальність також дуже корисна, щоб допомогти клієнтам «тестувати» продукти перед покупкою. Оскільки технології мобільної віртуальної приміркової вдосконалюються, вдосконалюються і розумні дзеркала в магазинах [24].

## РОЗДІЛ 4 ОГЛЯД ІНСТРУМЕНТІВ РОЗРОБКИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

На даний момент існує велика кількість засобів імплементації AR у різноманітні проєкти, переглянемо найбільш поширені.

### 4.1 Платформа Vuforia

За допомогою платформи Vuforia, розробники створюють доповнену реальність, доступ до якої можна отримати через веббраузер за допомогою мобільних пристроїв та настільних комп'ютерів.

Vuforia надає повний набір інструментів і функцій для створення доповненої реальності, в тому числі розпізнавання зображень, відстеження та 3D-рендеринг. Його технологія розпізнавання зображень дозволяє розробникам ідентифікувати та відстежувати об'єкти реального світу, дозволяючи їм створювати інтерактивний досвід, який поєднує фізичний і цифровий світи. Vuforia пропонує низку інструментів 3D-рендерінгу для створення та відображення високоякісних 3D-моделей та анімацій в рамках свого роботи AR.

Vuforia підтримується багатьма платформами, iOS, Android та Інтернет серед них. Vuforia надає просте у використанні вебсередовище розробки, яке дозволяє розробникам створювати та публікувати AR-контент без необхідності встановлювати програмне забезпечення чи SDK.

Vuforia надає детально прописані документацію та інструкції для кращого розуміння роботи із середовищем, а численна спільнота розробників допомагають та підтримають колег під час їхньої роботи над проєктами із доповненою реальністю [25].

## 4.2 Бібліотека AR.js

AR.js — це бібліотека з відкритим вихідним кодом для створення досвіду доповненої реальності в Інтернеті. Він розроблений для роботи в будь-якому браузері, який підтримує WebRTC і WebGL, і може використовуватися з будь-яким смартфоном або планшетом з камерою. AR.js використовує AR на основі маркерів, де маркер — це попередньо визначене зображення, яке запускає відображення віртуального вмісту під час перегляду через камеру.

AR.js створено на основі three.js, популярної бібліотеки JavaScript для створення 3D-графіки в Інтернеті. Він використовує алгоритми комп'ютерного зору для відстеження положення та орієнтації камери в реальному часі, що дозволяє віртуальним об'єктам виглядати так, ніби вони прив'язані до реального світу.

AR.js підтримує роботу кількох маркерів, що дозволяє створювати складніші AR-проекти. Він також може працювати з доповненою реальністю на основі місцезнаходження, де місцезнаходження користувача використовується для запуску відображення віртуального вмісту.

AR.js має велику та активну спільноту розробників, які створили додаткові модулі доповненої реальності за допомогою бібліотеки. Вони підтримують різноманітні додаткові функції, такі як відтворення відео, візуалізація 3D-об'єктів і анімація.

AR.js сумісний з різними платформами та пристроями: пристрої Android та iOS, настільні комп'ютери та навіть деякі гарнітури VR. Він сумісний із популярними фреймворками веброботи — як React і Vue.js.

Одним із потенційних обмежень AR.js є його залежність від маркерів, що може частково обмежити його застосування. Однак AR.js постійно оновлюється та вдосконалюється, а регулярно додаються нові функції.

### 4.3 Платформа 8th Wall

8th Wall — це хмарна платформа для розробки досвіду доповненої реальності в Інтернеті. Він надає розробникам набір інструментів і ресурсів для створення AR-контенту, до якого можна отримати доступ через веб-браузер мобільного пристрою без необхідності використання програми чи спеціального обладнання. Технологія 8th Wall забезпечує безперебійну роботу вебдоповненої реальності на різних пристроях і платформах, та робить її доступним і універсальним рішенням для розробки доповненої реальності.

8th Wall працює на основі комп'ютерного бачення та технології розпізнавання зображень, які дозволяє точно відстежувати та ідентифікувати об'єкти в реальному світі. Наприклад, розробник може створити AR-гру, яка вимагає від гравця фізичного переміщення в просторі та взаємодії з об'єктами в навколишньому середовищі.

8th Wall доповнену реальність як на основі маркерів, так і без них. У AR на основі маркерів фізичний маркер, як-от QR-код або зображення, використовується для запуску AR. Безмаркерна AR, з іншого боку, використовує комп'ютерний зір та інші технології для виявлення та відстеження об'єктів у середовищі без необхідності використання маркера.

8th Wall надає ряд інструментів для розробників для створення та налаштування AR-контенту, включаючи вебредактор і бібліотеку готових AR-шаблонів і компонентів. 8th Wall підтримує популярні мови програмування та фреймворки, включаючи JavaScript і React.

Одним з унікальних аспектів 8th Wall є його інтеграція з такими вебтехнологіями, як WebXR і WebRTC. За рахунок цього функціоналу забезпечуюються такі функції, як спілкування в реальному часі та співпраця між декількома користувачами в одному середовищі AR [26].

## 4.4 Платформа ZapWorks

ZapWorks — це комплексна та потужна хмарна платформа розробки доповненої реальності. Розроблена компанією Zappar, ZapWorks пропонує зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє розробникам і дизайнерам легко створювати та публікувати свій AR-контент.

ZapWorks надає різноманітні потужні інструменти та функції, які допомагають користувачам створювати високоякісний контент AR – розпізнавання зображень, інтеграцію відео та аудіо, імпорт 3D-моделі та анімацію, сценарії тощо. Платформа пропонує ряд шаблонів із можливістю їхнього подальшого налаштування і готових компонентів, які легко змінювати та адаптувати відповідно до конкретних вимог проекту. ZapWorks підтримується багатьма платформами.

Окрім потужних інструментів розробки, ZapWorks пропонує функції аналітики та відстеження, які дозволяють користувачам відстежувати ефективність їхнього AR-контенту та отримувати розуміння взаємодії та поведінки користувачів. Цю інформацію потім можна використовувати для вдосконалення та вдосконалення майбутніх проектів AR.

## РОЗДІЛ 5 ПРАКТИЧНА ЧАСТИНА

### 5.1 Вибір моделі розробки

Перед початком створення програмного продукту було розглянуто декілька моделей розробки програмного забезпечення. Як результат, обрано методологію розробки ПЗ з використанням водоспадної моделі.

Водоспадна модель розробки програмного забезпечення (Waterfall model) – це лінійна послідовна модель, яка передбачає покрокове розроблення програмного продукту від концепції до завершення, з послідовним переходом від однієї фази до наступної.

Основні переваги водоспадної моделі розробки ПЗ полягають у простоті і зрозумілості, структурованості, передбачуваності результату на кожному етапі. Крім того, ця модель забезпечує зручне керування процесом розробки, передбачає створення детальної документації, що сприяє легкому розумінню концепції проекту у випадку подальшої роботи над ним.

Однак, у цієї моделі є і певні недоліки. Недоліки включають брак гнучкості та відсутність можливості зміни вимог на ранніх етапах розробки. Крім того, ця модель не передбачає взаємодії з клієнтом і не дає можливості отримати відгук на ранніх етапах розробки.

Незважаючи на недоліки, водоспадна модель розробки програмного забезпечення рекомендується для розробки вебзастосунків, зокрема інтернет-магазинів, оскільки такі проекти вимагають структурованого і послідовного підходу, з конкретними етапами, завершенням кожного етапу та чіткими критеріями переходу на наступний етап.

### 5.2 Огляд використаних технологій

У ході роботи на випускним кваліфікаційним проектом було розглянуто найбільш популярні засоби для впровадження 3D-графіки та доповненої

реальності у вебсередовище. Після аналізу переваг та недоліків було обрано такі засоби:

- React Three Fiber;
- Tailwind CSS;
- Framer Motion;
- DALL·E 2;
- AR.js (детальніше про AR.js у розділі **4.2 Бібліотека AR.js**);
- Node.js та npm.

React — це бібліотека JavaScript для створення користувацьких інтерфейсів, розроблена Facebook. Вона була вперше випущена в 2013 році і з тих пір стала однією з найпопулярніших бібліотек для створення веб-додатків. React відомий своїм декларативним підходом до створення інтерфейсів користувача. В основі React лежить концепція компонентів, які є багаторазово використовуваними фрагментами коду, що представляють частину інтерфейсу користувача. Компоненти можна комбінувати для створення складніших інтерфейсів, і їх можна повторно використовувати на кількох сторінках або програмах. React надає набір інструментів для керування станом компонентів. React використовує віртуальну DOM для керування рендерингом компонентів. Віртуальна DOM — це спрощене представлення фактичного DOM, що дозволяє React ефективно оновлювати інтерфейс, оновлюючи лише ті частини, які змінилися. Цей підхід до рендерингу відомий як рендеринг «точно вчасно» і є ключовою частиною продуктивності React.

Для збірки React проекту використовується Vite.

React Three Fiber — це бібліотека для створення 3D-графіки та анімації в браузері за допомогою React. Вона надає декларативний API, який дозволяє розробникам створювати 3D-сцени, використовуючи знайомий синтаксис і структуру компонентів React. Бібліотека побудована на основі Three.js (детальніше про Three.js у розділі **2.2.1 Бібліотека Three.js**) і надає спрощений API вищого рівня, який полегшує створення складних 3D-сцен. Для створення 3D-сцени за допомогою RTF необхідно використовувати JSX

синтаксис для створення React-компонентів, які можуть відображати 3D-об'єкти [27].

Tailwind CSS – це набір стилів CSS, який надає можливість швидко та ефективно стилізувати вебдодатки. Він пропонує атомізовану систему дизайну, де стилі можна задавати на рівні окремих елементів, класів та властивостей, забезпечуючи гнучкість та швидкість розробки. У Tailwind CSS використовується концепція «utility-first», коли стилі задаються за допомогою коротких класів, які описують конкретну властивість елемента (наприклад, ``bg-red-500`` для задання червоного кольору фону) [28].

Framer Motion – це високорівнева бібліотека анімації для React, яка надає потужні інструменти для створення різноманітних анімаційних ефектів та переходів на вебсторінках. Вона пропонує абстрактні компоненти, які можна використовувати для створення різних видів анімацій та взаємодії з користувачем. Framer Motion забезпечує легку інтеграцію з React-проектами та дозволяє використовувати декларативний підхід до створення анімацій. Бібліотека підтримує широкий спектр ефектів, включаючи повзунки, затемнення, витягування, переміщення та багато інших [29].

DALL·E 2 – це модель глибокого навчання штучного інтелекту розроблена в OpenAI з метою генерації зображень за запитом. Назва моделі DALLE походить від «Drawing AIs with Language Model and Efficient Training» (створення ШІ для малювання зображень за допомогою мовних моделей та ефективного тренування). DALLE використовує технології навчання без вчителя та генеративність глибоких нейронних мереж, щоб створювати вражаючі зображення на основі текстових описів. Ця модель здатна генерувати реалістичні зображення різних об'єктів та сцен за допомогою декількох мовних команд, що задаються користувачем. DALLE AI вивчає співвідношення між текстом та зображеннями, використовуючи технології машинного навчання, такі як трансформери та генеративні моделі зворотного розповсюдження помилок (GPT), які дозволяють йому розуміти текст та генерувати відповідні зображення [30].

Node.js – це безкоштовна та відкрита серверна платформа, побудована на базі двигуна V8 JavaScript заснованого на Chrome. Вона дозволяє виконувати JavaScript код на стороні сервера.

npm – це менеджер пакетів для програмного забезпечення, який виконує збір, публікацію та управління пакетами програмного забезпечення, що написані на мові JavaScript. Цей інструмент є одним з найбільш популярних інструментів, які використовуються для забезпечення ефективного розгортання пакетів у проектах на основі Node.js. npm забезпечує доступ до мільйонів безкоштовних та платних пакетів, які можуть бути встановлені у проект з використанням командної строки [31].

### 5.3 Специфікація користувацьких вимог

Зобразимо специфікацію користувацьких функціональних вимог у вигляді такої таблиці (таблиця 4.1):

Таблиця. 4.1 – Користувацькі функціональні вимоги

Назва	Опис
Перегляд головної сторінки	Користувач може переглядати головну сторінку інтернет-магазину із інформацією про функціонал сайту
Перегляд сторінки із налаштуванням	Користувач може перейти на сторінку конкретного товару для подальшого його редагування
Використання фільтрів	Користувач може використовувати фільтри для накладання зображення у вигляді логотипу та на всю поверхню об'єкта

## Продовження таблиці 4.1 – Користувацькі функціональні вимоги

Назва	Опис
Зміна кольору	Користувач може змінювати колір товарів за допомогою меню із палітрою кольорів
Накладення зображення	Користувач може завантажити зображення із локального пристрою для використання під редагування виду товару
Використання штучного інтелекту	Користувач може використовувати штучний інтелект при створенні дизайну товару вказуючи деталі текстом у діалоговому вікні
Завантаження результату	Після завершення редагування товару, користувач може завантажити отриманий результат у вигляді png файлу на локальний пристрій

#### 5.4 Архітектура системи

Під час розробки проєкту було обрано клієнт-серверну модель архітектури. Клієнт-серверна архітектура є однією з основних архітектурних парадигм, яка використовується при розробці програмного забезпечення. У цій моделі програмне забезпечення розбивається на дві основні складові: клієнтську і серверну частину. Клієнтська частина відповідає за інтерфейс користувача та взаємодію з ним, тоді як серверна частина забезпечує зберігання та обробку даних, виконання бізнес-логіки та інші функціональні можливості.

Однією з головних переваг клієнт-серверної архітектури є можливість розподілення обробки даних між сервером та клієнтом. Це дозволяє зменшити навантаження на окремі компоненти системи, підвищити їхню масштабованість та забезпечити більш високу продуктивність. Також така архітектура гарантує кращий рівень безпеки даних та більш ефективне управління доступом до них.

## 5.5 Тестування

Тестування системи проходило вручну, а також за допомогою інструменту тестування API Postman.

Результати тестування можна переглянути у таблиці 5.1.

Таблиця 5.1 – Результати тестування системи

Назва вимоги	Результати тестування
Перегляд головної сторінки	Пройдено
Перегляд сторінки із налаштуванням	Пройдено
Використання фільтрів	Пройдено
Зміна кольору	Пройдено
Накладення зображення	Пройдено
Використання штучного інтелекту	Пройдено
Завантаження результату	Пройдено

## 5.6 Опис кінцевої системи

Основні компоненти вебзастосунку описано у ДОДАТКУ А.

## 5.7 Способи покращення функціонування системи

У ході роботи над практичною частиною випускної кваліфікаційної роботи було проаналізовано способи її покращення та розширення у майбутньому. Серед ідей:

- розробка мобільної версії застосунку;
- збільшення списку можливих для покупки товарів: додавання елементів інтер'єру тощо;

- індивідуальна рекомендація товарів для користувача з використанням технології штучного інтелекту;
- застосування різних видів доповненої реальності;
- віртуальні тури приміщенням.

## ВИСНОВКИ

У результаті виконання випускної кваліфікаційної роботи опрацьовано теоретичний матеріал із поясненням принципів роботи 3D-графіки та доповненої реальності, досліджено основні сфери їхнього застосування, виявлено головні напрямки поширення технологій та актуальні тренди, переглянуто найбільш популярні інструменти для імплементації цих технологій у вебзастосунках, проаналізовано їхні переваги та недоліки.

На сьогоднішній день технології 3D-графіки та доповненої реальності перебувають у фазі активного розвитку та поступово впроваджуються практично у всі сфери людської діяльності: маркетинг, освіту, медицину, туризм, будівництво тощо. Вебзастосунки електронної комерції з використанням технологій 3D-графіки та доповненої реальності дозволяють значно покращити рівень якості користувацького досвіду та привернути увагу потенційних покупців. Засоби для імплементації 3D-графіки у вебсередовище надають можливість переглядати більш детальні та реалістичні моделі продуктів, реалізувати різноманітні ефекти та анімацію. Застосовуючи AR-технології, можна створювати функціонал із підтримкою інтерактивної взаємодії, який забезпечує унікальний досвід для користувача та підвищує ефективність бізнес-процесів.

Практичну частину реалізовано у вигляді вебзастосунку для електронної комерції, де користувачеві надається можливість переглянути тривимірну модель товару, змінювати колір його поверхні, логотип, використовувати власноруч завантажені зображення, згенерувати новий стиль за допомогою штучного інтелекту, зберегти отриманий результат локально, а також, використовуючи камеру смартфона чи камеру персонального комп'ютера, візуально представити об'єкт в реальному фізичному середовищі.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. The History of 3D Animation: a deep dive [Електронний ресурс] // InFocus Film School. – 2022. – Режим доступу до ресурсу: <https://infocusfilmschool.com/history-of-3d-animation/>.
2. History of 3D Computer Graphics [Електронний ресурс] // 3D Horse. – 2017. – Режим доступу до ресурсу: <https://www.3dhorse.com/blogs/3d/history-of-3d-computer-graphics>.
3. Храпаченко Д. М. Історія виникнення та розвитку комп'ютерної графіки [Електронний ресурс] / Дарина Миколаївна Храпаченко. – 2021. – Режим доступу до ресурсу: <https://phm.cuspu.edu.ua/nauka/naukovo-populiarni-publikatsii/891-istoriya-vynyknennya-ta-rozvytku-komp-yuternoyi-hrafiiky.html>.
4. 3D Animation: A Brief History and a Look to the Future [Електронний ресурс] // Skillshare blog. – 2022. – Режим доступу до ресурсу: <https://www.skillshare.com/en/blog/3d-animation-a-brief-history-and-a-look-to-the-future/>.
5. 3D Animation Evolution & History [Електронний ресурс] // Study.com. – 2022. – Режим доступу до ресурсу: <https://study.com/learn/lesson/3d-animation-evolution-history.html>.
6. What Are Polygons in 3D Modeling [Електронний ресурс] // CGI Furniture. – 2023. – Режим доступу до ресурсу: <https://cgifurniture.com/what-are-polygons-in-3d-modeling/>.
7. How to Make Textures for 3D models: Basics and Tips [Електронний ресурс] // 3D-Ace. – 2021. – Режим доступу до ресурсу: <https://3d-ace.com/blog/how-to-make-textures-for-3d-models-basics-and-tips/>.
8. The Stages of Creating a 3D Model [Електронний ресурс] // CGHero. – 2022. – Режим доступу до ресурсу: <https://cghero.com/articles/stages-of-creating-3d-model>.

9. The 4 Main Types of 3D Modeling – Simply Explained [Электронный ресурс] // All3DP. – 2020. – Режим доступа до ресурсу: <https://all3dp.com/2/types-of-3d-modeling/>.
10. What is 3D Surface Modeling? [Электронный ресурс] // ITS. – 2021. – Режим доступа до ресурсу: <https://it-s.com/what-is-3d-surface-modeling/>.
11. Glossary | Solid Modeling [Электронный ресурс] // Spatial Corp. – 2021. – Режим доступа до ресурсу: <https://www.spatial.com/resources/glossary/what-is-solid-modeling>.
12. WebGL: 2D and 3D graphics for the web [Электронный ресурс] // MDN. – 2023. – Режим доступа до ресурсу: [https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API).
13. Three.js – JavaScript 3D Library [Электронный ресурс] // three.js. – 2023. – Режим доступа до ресурсу: <https://threejs.org/>.
14. Working With 3D Models in Three.js [Электронный ресурс] // Better Programming. – 2022. – Режим доступа до ресурсу: <https://betterprogramming.pub/working-with-3d-model-in-three-js-e228621141af>.
15. 3D Models [Электронный ресурс] // A-FRAME. – 2023. – Режим доступа до ресурсу: <https://aframe.io/docs/1.4.0/introduction/models.html>.
16. Introduction to Babylon.js Features [Электронный ресурс] // babylon.js. – 2023. – Режим доступа до ресурсу: <https://doc.babylonjs.com/features/introductionToFeatures>.
17. User Manual [Электронный ресурс] // PlayCanvas Manual. – 2023. – Режим доступа до ресурсу: <https://developer.playcanvas.com/en/user-manual/>.
18. History of virtual reality: Timeline [Электронный ресурс] // Verdict. – 2020. – Режим доступа до ресурсу: <https://www.verdict.co.uk/history-virtual-reality-timeline/>.
19. History of VR - Timeline of Events and Tech Development [Электронный ресурс] // VirtualSpeech. – 2022. – Режим доступа до ресурсу: <https://virtualspeech.com/blog/history-of-vr>.

20. How Does Augmented Reality Work? [Электронный ресурс] // SmartTek Solutions. – 2023. – Режим доступа до ресурсу: <https://smarttek.solutions/blog/what-is-augmented-reality-ar-and-how-does-it-work/>.
21. An overview of augmented reality technology [Электронный ресурс] // IOPscience. – 2019. – Режим доступа до ресурсу: <https://iopscience.iop.org/article/10.1088/1742-6596/1237/2/022082/pdf>.
22. An overview of augmented reality technology [Электронный ресурс] / Y.Chen, Q. Wang, H. Chen, X. Song // ResearchGate. – 2019. – Режим доступа до ресурсу: [https://www.researchgate.net/publication/334420829\\_An\\_overview\\_of\\_augmented\\_reality\\_technology](https://www.researchgate.net/publication/334420829_An_overview_of_augmented_reality_technology).
23. Augmented Reality (AR) vs Virtual Reality (VR) [Электронный ресурс] // XMReality. – 2022. – Режим доступа до ресурсу: <https://www.xmreality.com/blog/augmented-reality-vs-virtual-reality>.
24. Emrich T. 23 Augmented Reality Trends to Keep an Eye on for 2023 [Электронный ресурс] / Tom Emrich // LinkedIn. – 2023. – Режим доступа до ресурсу: <https://www.linkedin.com/pulse/23-augmented-reality-trends-keep-eye-2023-tom-emrich>.
25. Vuforia Engine Overview [Электронный ресурс] // VuforiaLibrary – Режим доступа до ресурсу: <https://library.vuforia.com/getting-started/vuforia-features>.
26. Introduction [Электронный ресурс] // 8th Wall – Режим доступа до ресурсу: <https://www.8thwall.com/docs/>.
27. React Three Fiber Documentation [Электронный ресурс] // Pmndrs.docs – Режим доступа до ресурсу: <https://docs.pmnd.rs/react-three-fiber/getting-started/introduction>.
28. How To Use Tailwind CSS With React [Электронный ресурс] // Medium. – 2022. – Режим доступа до ресурсу: <https://medium.com/codingthesmartway-com-blog/how-to-use-tailwind-css-with-react-9dd78bbdc0e0>.

29. Framer Motion examples for React animations [Электронный ресурс] // Refine Dev. – 2022. – Режим доступа до ресурсу: <https://refine.dev/blog/framer-motion-react-animations/>.
30. How to Generate Images using React and the Dall-E 2 API [Электронный ресурс] // freeCodeCamp. – 2022. – Режим доступа до ресурсу: <https://www.freecodecamp.org/news/generate-images-using-react-and-dall-e-api-react-and-openai-api-tutorial/>.
31. Downloading and installing Node.js and npm [Электронный ресурс] // npm Docs – Режим доступа до ресурсу: <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>.

## ДОДАТОК А

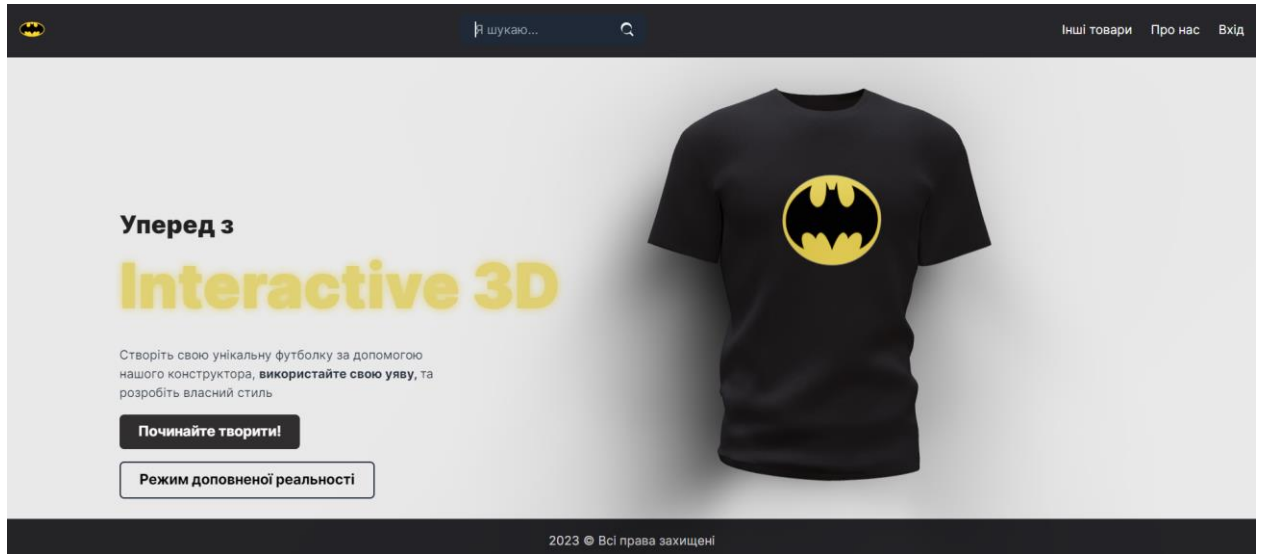


Рисунок А.1 – Головна сторінка



Рисунок А.2 – Режим доповненої реальності

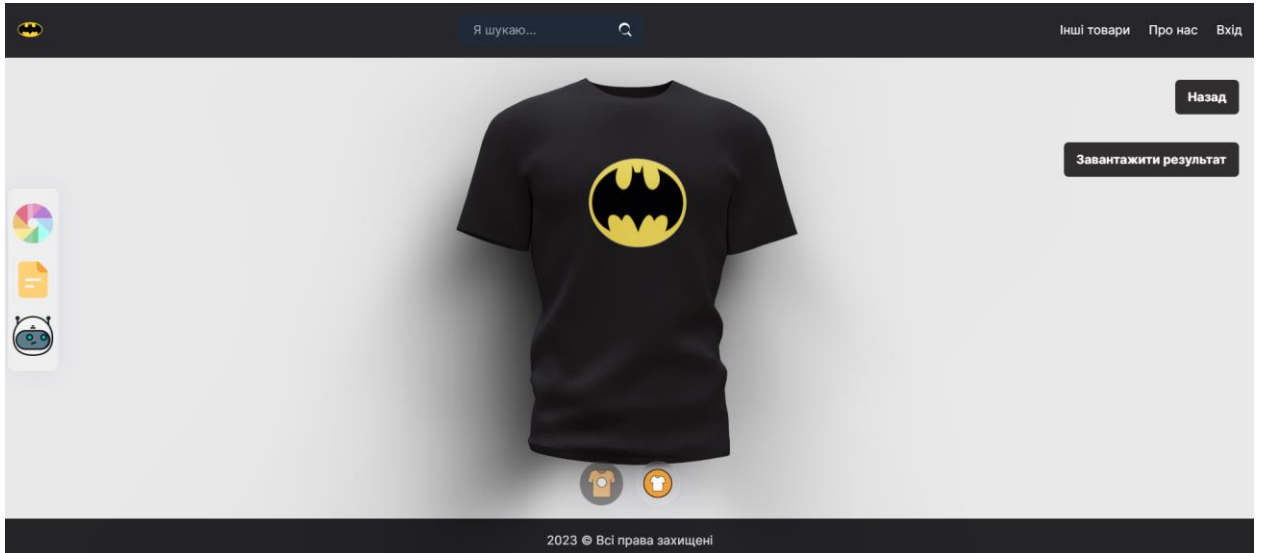
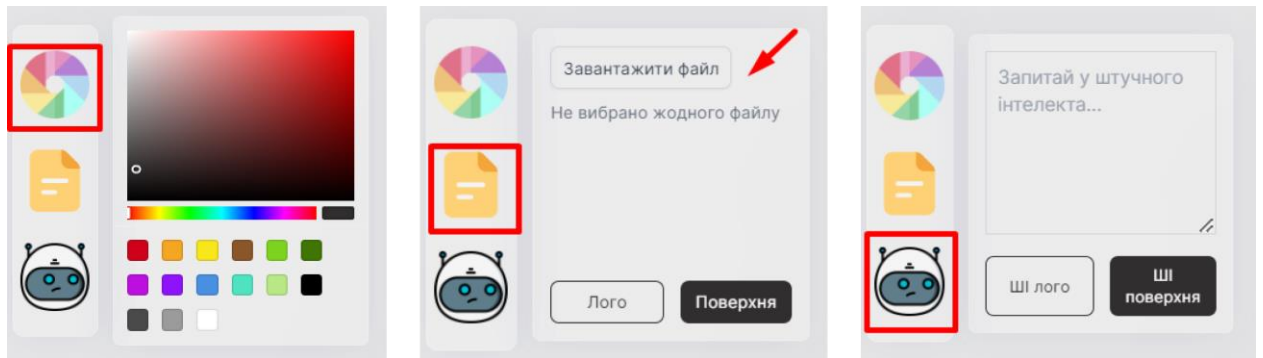


Рисунок А.3 – Сторінка налаштування вигляду одягу



а)

б)

в)

Рисунок А.4 – Меню: а – редагування кольору; б – завантаження користувачького файлу; в – використання штучного інтелекту



Рисунок А.5 – Робота фільтрів: фільтри вимкнені



Рисунок А.6 – Робота фільтрів: фільтр «Лого» активний



Рисунок А.7 – Робота фільтрів: фільтр «Поверхня» активний



Рисунок А.8 – Робота фільтрів: обидва фільтри активні

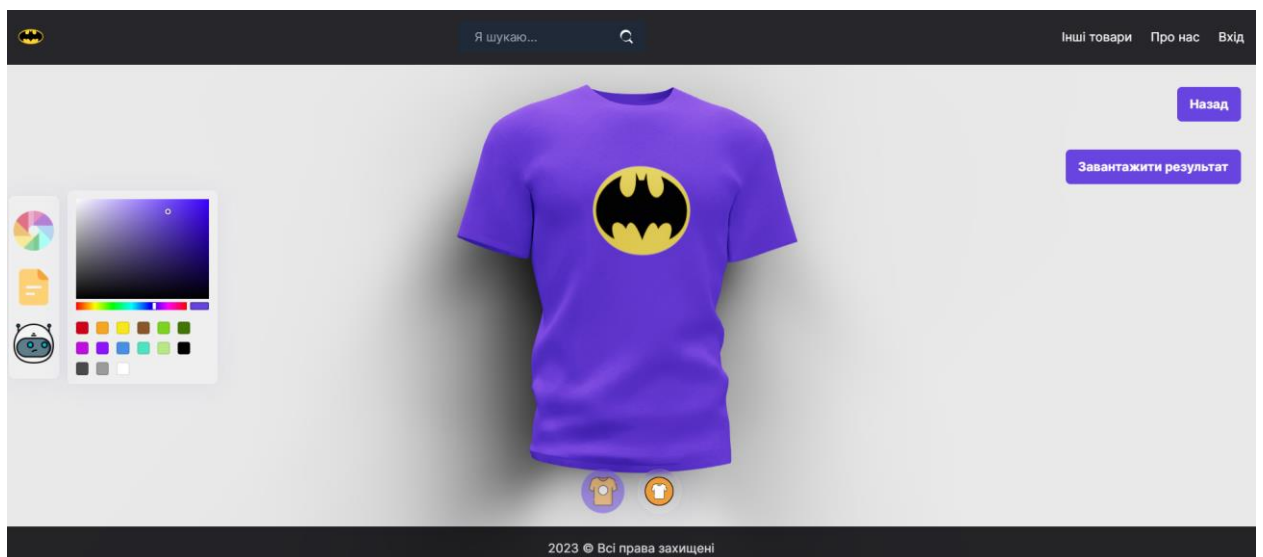


Рисунок А.9 – Редагування кольору

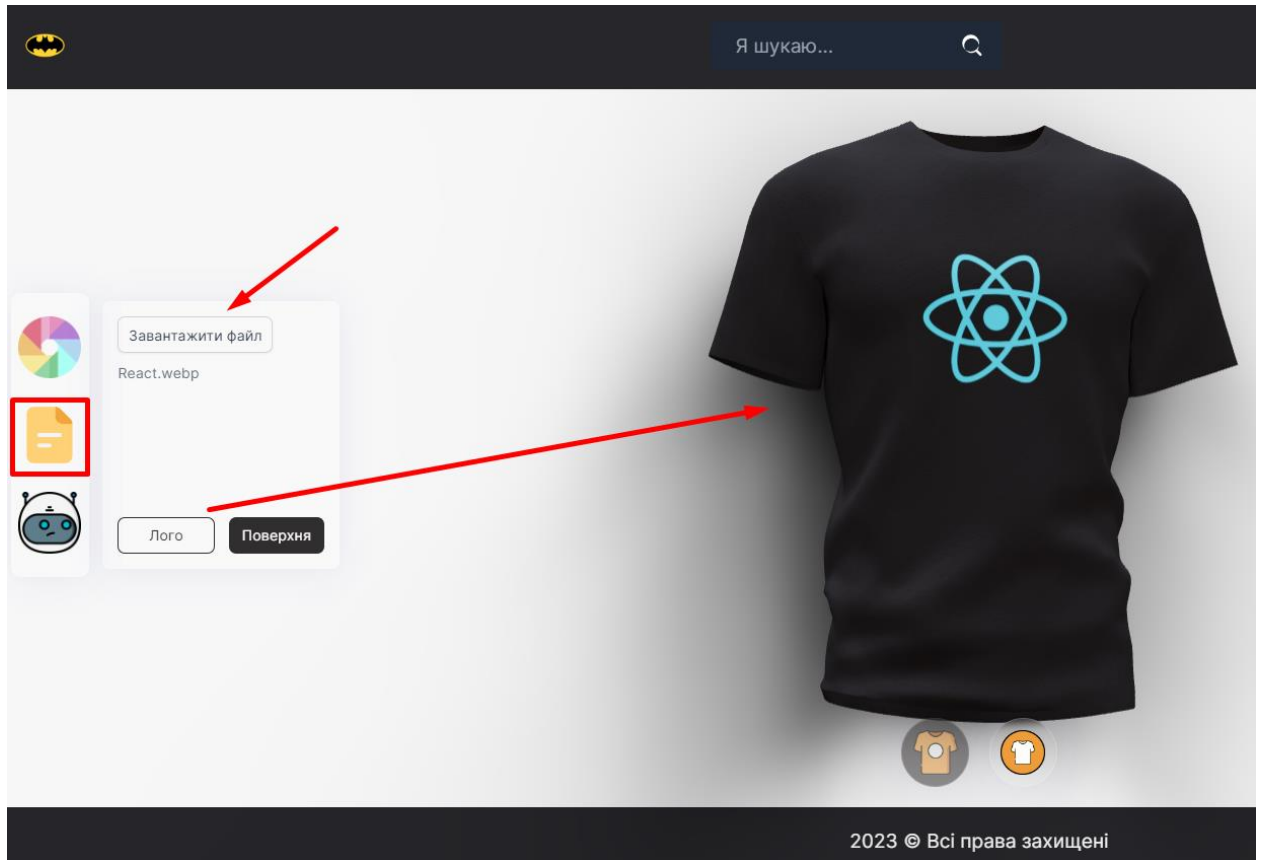


Рисунок А.10 – Завантаження користувацького файлу: накладання зображення у вигляді логотипу

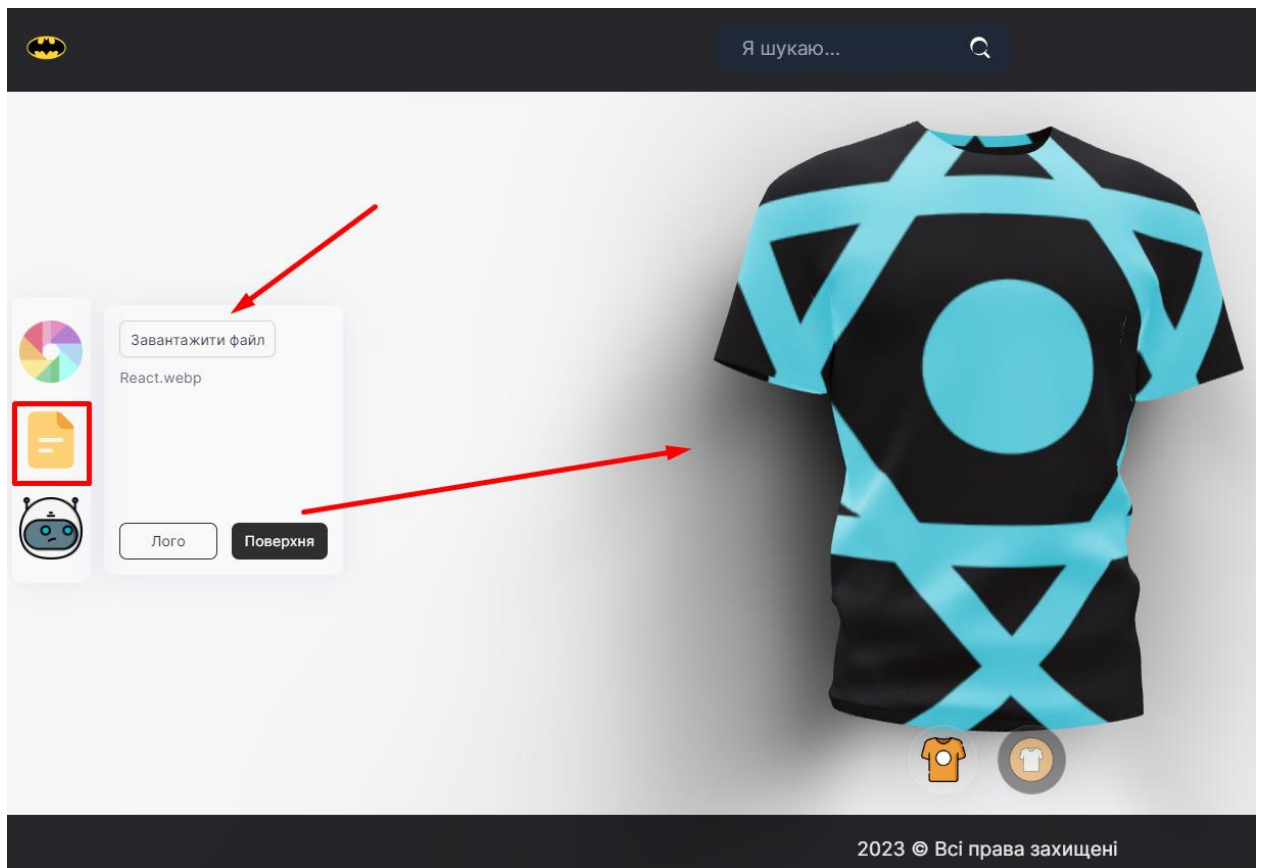


Рисунок А.11 – Завантаження користувацького файлу: накладання зображення на всю поверхню одягу

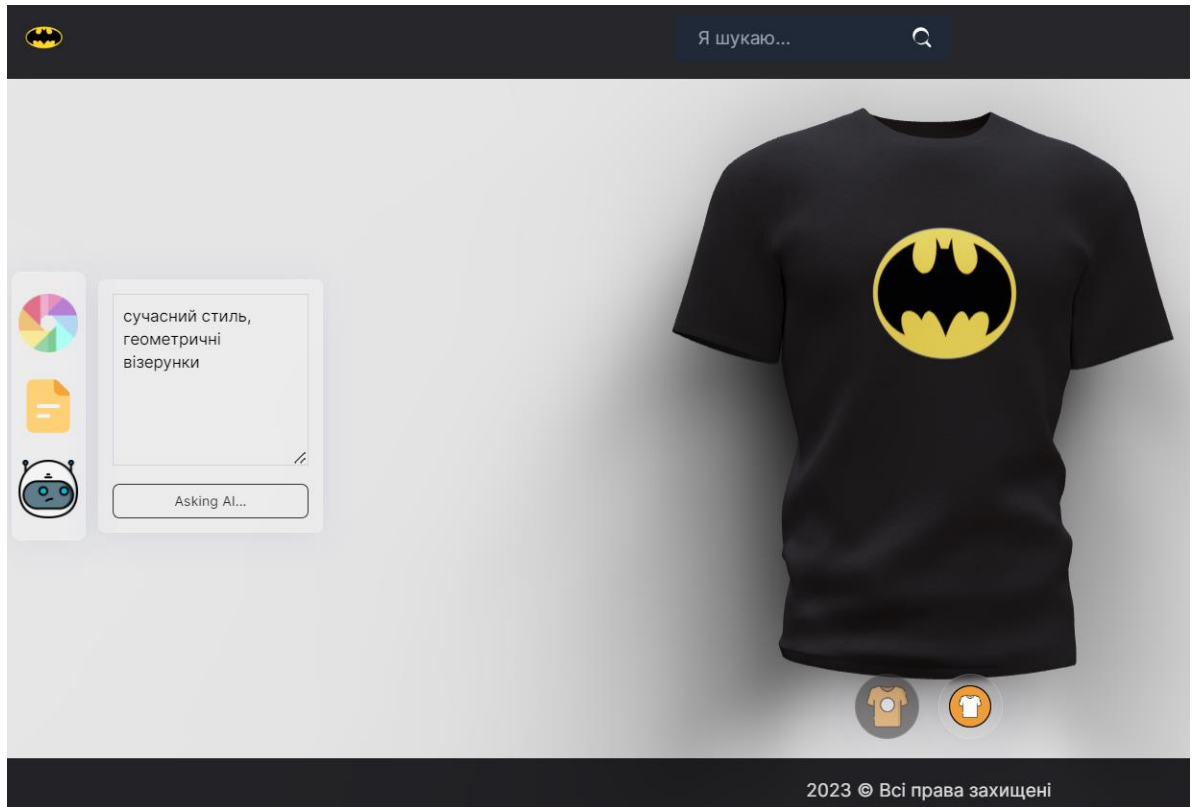


Рисунок А.12 – Використання штучного інтелекту

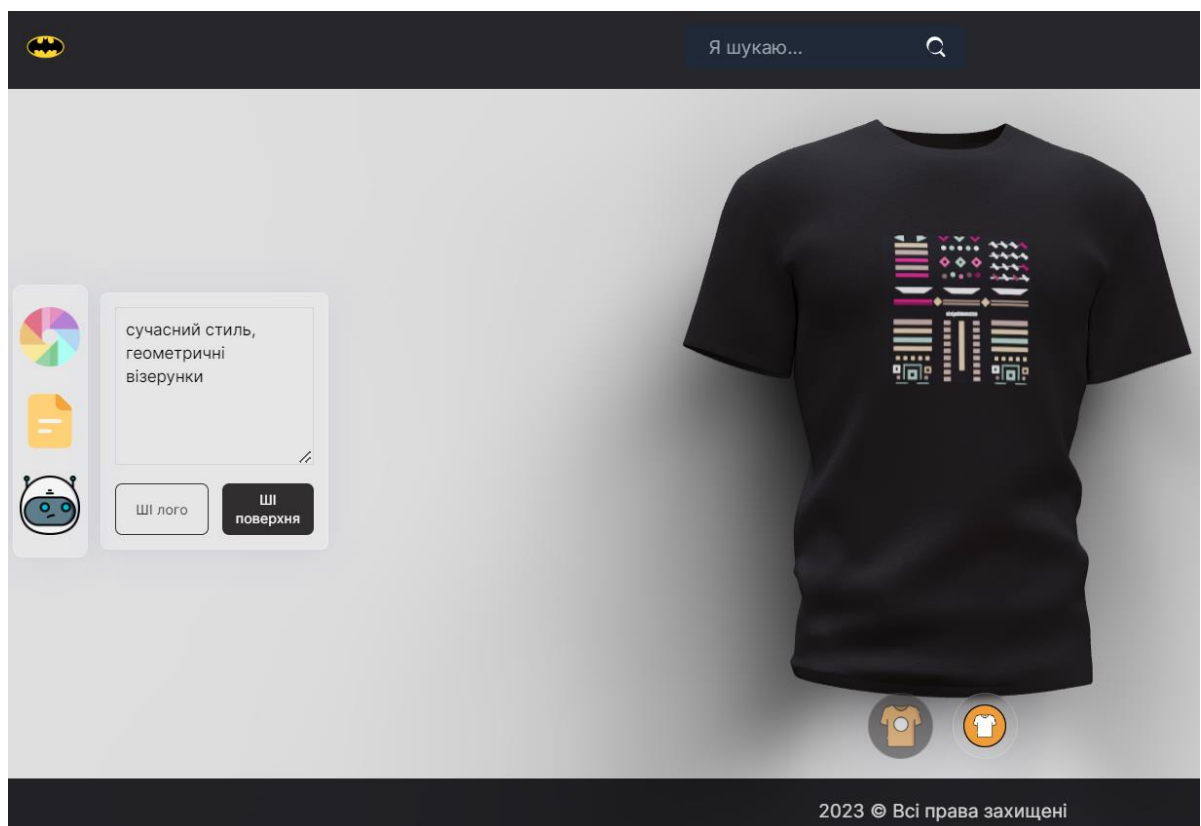


Рисунок А.13 – Використання штучного інтелекту: накладання зображення у вигляді логотипу

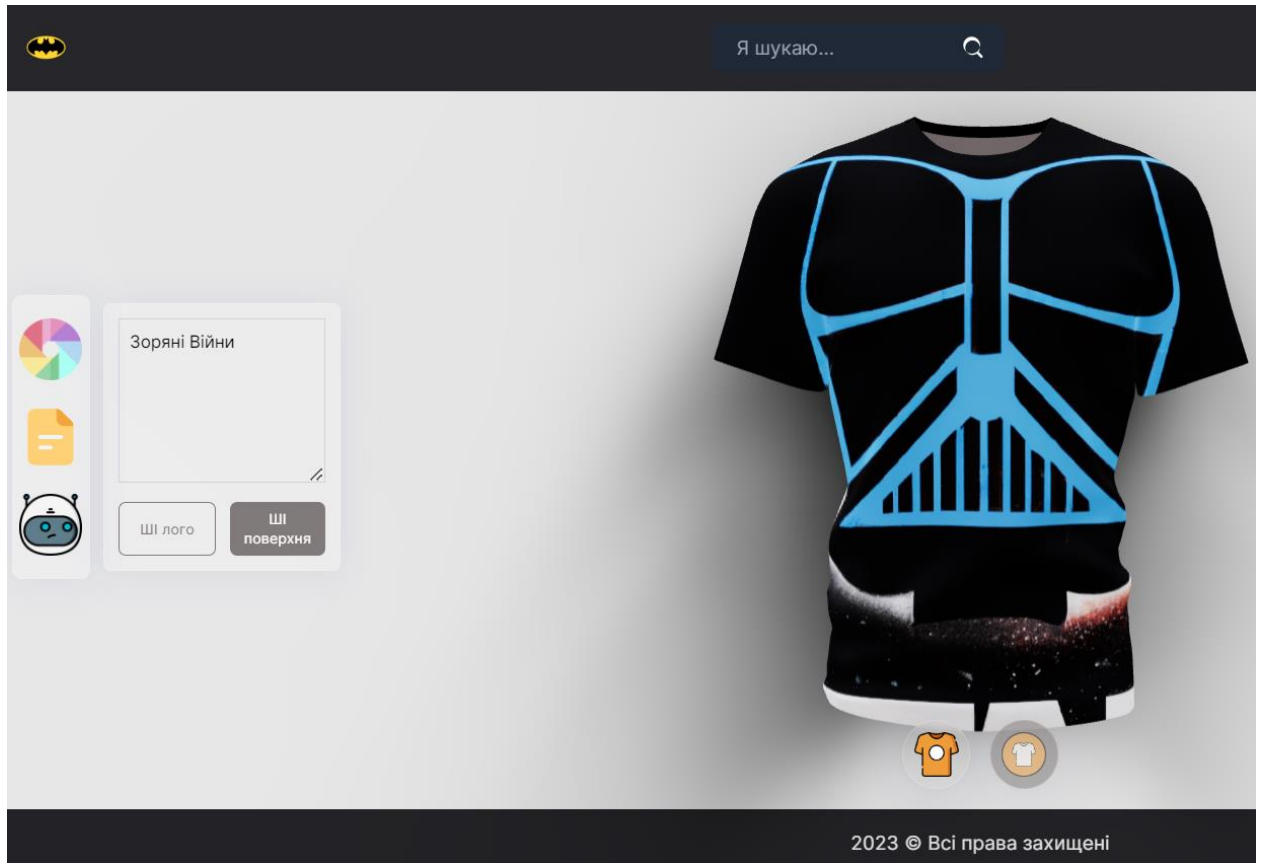


Рисунок А.14 – Використання штучного інтелекту: накладання зображення на всю поверхню одягу

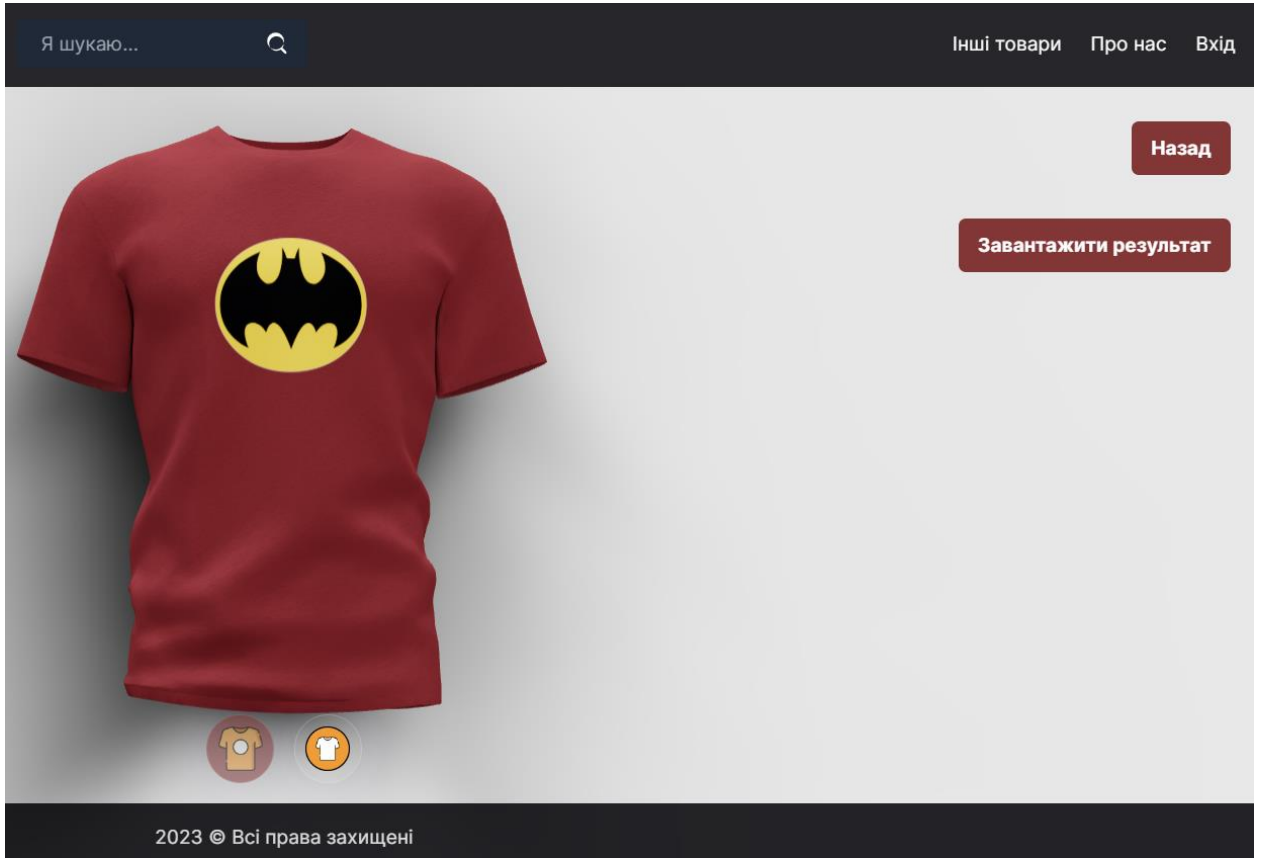


Рисунок А.15 – Завантаження результату на локальну машину



Рисунок А.16 – Перегляд завантаженого результату