

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
(повна назва)

Кафедра _____ Автоматизації проектування обчислювальної техніки _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Методи моделювання приміщень розумного будинку

засобами доповненої реальності

(тема)

Виконав:

здобувач _____ 2025 _____ року навчання,

групи _____ СКСм-23-2 _____

Воронін Р.А.

(прізвище, ініціали)

Спеціальність _____

123 – Комп'ютерна інженерія

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Спеціалізовані

комп'ютерні системи

(повна назва освітньої програми)

Керівник: _____ доц. Ларченко Л.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри



(підпис)

Чумаченко С.В.

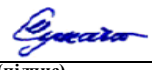
(прізвище, ініціали)

2025

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Автоматизації проектування обчислювальної техніки
Рівень вищої освіти другий (магістерський)
Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)

“ ” 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві Вороніну Роману Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи моделювання приміщень розумного будинку
засобами доповненої реальності

затверджена наказом по університету від " 08 " 11 2024 р. № 1189 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 22.01.2025

3. Вихідні дані до роботи

Технології для AR візуалізації

Мова програмування Javascript

Середовище розробки Webstorm

4. Перелік питань, що потрібно опрацювати в роботі

1 Огляд концепції розумного будинку та технологій доповненої реальності

2 Аналіз методів та інструментів для створення AR-застосунків

3 Методи моделювання приміщень засобами доповненої реальності

4 Розробка AR-застосунку для моделювання приміщень у розумному будинку

5 Результати тестування розробленого AR-застосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 19 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	Дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	02.09.2024-05.09.2024	
2	Аналіз предметної області	06.09.2024-29.09.2024	
3	Аналіз джерел з проблемної галузі	30.09.2024-07.10.2024	
4	Дослідження існуючих методів моделювання приміщень засобами доповненої реальності	08.10.2024-11.10.2024	
5	Аналіз інструментів для створення застосунків з використанням доповненої реальності	12.10.2024-20.10.2024	
6	Програмна реалізація застосунку моделювання приміщень	21.10.2024-19.11.2024	
7	Тестування розробленого застосунку	20.11.2024-28.11.2024	
8	Оформлення пояснювальної записки	29.12.2024-14.12.2024	
9	Оформлення графічного матеріалу	15.12.2024-01.01.2025	
10	Перевірка виконаного проекту керівником	02.01.2025-10.01.2025	
	Захист кваліфікаційної роботи	22.01.25	

Дата видачі завдання 02.09.2024

Студент 
(підпис)

Керівник роботи 
(підпис)

доц. Ларченко Л.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 71 сторінок, 19 рисунків, 3 таблиці, 12 джерел за переліком посилань.

РОЗУМНИЙ БУДИНОК, ДОПОВНЕНА РЕАЛЬНІСТЬ,
МОДЕЛЮВАННЯ, GOOGLE MODEL VIEWER, UNITY, WEBXR API, 3D-
МОДЕЛІ, REACT, JAVASCRIPT

У кваліфікаційній роботі досліджено методи моделювання приміщень розумного будинку за допомогою технологій доповненої реальності (AR). Проведено аналіз ключових підходів до впровадження AR у сфері проєктування та управління інфраструктурою розумного будинку.

Розглянуто популярні інструменти для створення AR-додатків: Google Model Viewer, Unity та WebXR API з AR.js. Проведено їх детальний порівняльний аналіз з акцентом на продуктивність, функціональність та зручність інтеграції. Досліджено методи маркерного та немаркерного моделювання, а також використання технологій SLAM та хмар точок у AR-середовищі.

На основі проведених досліджень було розроблено AR-застосунок для моделювання приміщень. Вибрано технологічний стек, що включає React, Google Model Viewer та Node.js. Описано архітектуру застосунку, алгоритм його роботи, а також реалізовано ключові функції, такі як інтерактивний вибір 3D-моделей, керування AR-сценою та обробка користувацьких подій.

Проведено тестування застосунку, проаналізовано його продуктивність та стабільність, виявлено переваги та обмеження реалізованого рішення. Отримані результати підтверджують ефективність використання AR-технологій у проєктуванні та моделюванні приміщень розумного будинку.

ABSTRACT

The explanatory note contains 71 pages, 19 figures, 3 tables, and 12 references.

SMART HOME, AUGMENTED REALITY, MODELING, GOOGLE MODEL VIEWER, UNITY, WEBXR API, 3D MODELS, REACT, JAVASCRIPT

The qualification work investigates methods for modeling smart home environments using augmented reality (AR) technologies. An analysis of key approaches to implementing AR in the design and management of smart home infrastructure was conducted.

Popular tools for creating AR applications, such as Google Model Viewer, Unity, and WebXR API with AR.js, were reviewed. A detailed comparative analysis was carried out, focusing on performance, functionality, and ease of integration. Marker-based and markerless modeling methods were studied, along with SLAM and point cloud technologies in AR environments.

Based on the research, an AR application was developed for modeling smart home environments. The chosen technology stack includes React, Google Model Viewer, and Node.js. The architecture of the application, its operational algorithm, and key functions such as interactive 3D model selection, AR scene control, and user event handling were described and implemented.

The application was tested, its performance and stability were analyzed, and the strengths and limitations of the implemented solution were identified. The results confirm the effectiveness of AR technologies in designing and modeling smart home environments.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
1 СУЧАСНІ ПІДХОДИ ДО ПРОЄКТУВАННЯ РОЗУМНИХ БУДИНКІВ ТА ВИКОРИСТАННЯ ДОПОВНЕНОЇ РЕАЛЬНОСТІ	11
1.1 Концепція розумного будинку	11
1.2 Технології доповненої реальності та їх основні характеристики.....	16
1.3 Сучасні тренди впровадження AR у розумні будинки	22
1.4 Мета та постановка завдання	24
2 АНАЛІЗ ТЕХНОЛОГІЙ І МЕТОДІВ МОДЕЛЮВАННЯ ПРИМІЩЕНЬ У СИСТЕМАХ ДОПОВНЕНОЇ РЕАЛЬНОСТІ	26
2.1 Аналіз сучасних інструментів для створення AR-додатків	26
2.1.1 Google Model Viewer	27
2.1.2 Unity	30
2.1.3 WebXR API та AR.js	33
2.1.4 Порівняння інструментів для створення AR-застосунків	35
2.2 Методи моделювання за допомогою маркерів	37
2.2.1 Алгоритми розпізнавання маркерів	42
2.2.2 Методи обчислення позиції та орієнтації.....	45
2.2.3 Переваги та обмеження маркерного моделювання	47
2.3 Методи моделювання без використання маркерів	48
2.4 Технології використання хмар точок і SLAM у моделюванні	51
2.5 Порівняння методів моделювання AR для розумних будинків	54
3 РЕАЛІЗАЦІЯ AR-ЗАСТОСУНКУ ДЛЯ МОДЕЛЮВАННЯ ПРИМІЩЕНЬ У РОЗУМНОМУ БУДИНКУ	56
3.1 Вибір технологій та обґрунтування їх використання	56
3.2 Архітектура та структура застосунку	59

3.3 Алгоритм роботи застосунку.....	60
3.4 Ключові функції застосунку.....	62
4 ТЕСТУВАННЯ ЗАСТОСУНКУ ДЛЯ МОДЕЛЮВАННЯ ПРИМІЩЕНЬ РОЗУМНОГО БУДИНКУ.....	64
ВИСНОВКИ	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ДОДАТОК А	73
ДОДАТОК Б	83
ДОДАТОК В.....	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) – програмний інтерфейс додатків

AR (Augmented Reality) – доповнена реальність

CLI (Command Line Interface) – інтерфейс командного рядка

DOM (Document Object Model) – об'єктна модель документа

GLB (GL Binary) – бінарний формат передачі тривимірних моделей

GLTF (GL Transmission Format) – формат передачі тривимірних моделей

HDRI (High Dynamic Range Imaging) – освітлення з високим динамічним діапазоном

JSON (JavaScript Object Notation) – формат обміну даними

LOD (Level of Detail) – рівень деталей

MR (Mixed Reality) – змішана реальність

NPM (Node Package Manager) – менеджер пакетів Node.js

PBR (Physically Based Rendering) – фізично коректний рендеринг

PhysX – система фізичної симуляції від NVIDIA

PnP (Perspective-n-Point) – метод визначення просторового положення об'єктів за координатами точок

SDK (Software Development Kit) – набір інструментів для розробки програмного забезпечення

SLAM (Simultaneous Localization and Mapping) – одночасна локалізація та побудова карти

USDZ (Universal Scene Description Zip) – формат передачі тривимірних моделей від Apple

VR (Virtual Reality) – віртуальна реальність

ВСТУП

Методи моделювання приміщень розумного будинку за допомогою технологій доповненої реальності (AR) є одним із ключових напрямів досліджень у сфері сучасних інформаційних технологій. Використання AR дозволяє інтегрувати віртуальні об'єкти в реальне середовище, забезпечуючи точне моделювання, інтерактивність та зручність у процесах проєктування, управління та експлуатації приміщень. Завдяки розвитку вебтехнологій та зростанню обчислювальних можливостей, технології AR стали більш доступними та широко застосовуються у проєктах розумного будинку.

Особливу важливість у процесі впровадження AR-технологій відіграють методи моделювання, які визначають способи створення та обробки AR-сцен, точність позиціонування віртуальних об'єктів та стабільність їх інтеграції у фізичний простір. Серед них виділяють маркерні та немаркерні методи моделювання, а також технології, що базуються на алгоритмах SLAM (Simultaneous Localization and Mapping) та хмарах точок. Кожен з цих методів має свої переваги, обмеження та особливості застосування, що робить їх об'єктом детального дослідження.

У кваліфікаційній роботі акцент зроблено саме на дослідженні методів моделювання AR-сцен та їх практичному застосуванні для створення AR-застосунка, що дозволяє моделювати приміщення розумного будинку. Окрім цього, було проведено аналіз інструментів, що дозволяють реалізовувати сучасні методи, зокрема Google Model Viewer, Unity та WebXR API з AR.js.

Особлива увага приділена порівняльному аналізу методів моделювання для визначення їхньої ефективності у контексті конкретних завдань проєктування розумного будинку. Дослідження охоплює алгоритми розпізнавання маркерів, методи обчислення позиції та орієнтації AR-об'єктів, а також принципи роботи систем, що використовують технології SLAM.

Метою кваліфікаційної роботи є дослідження методів моделювання приміщень розумного будинку за допомогою технологій доповненої реальності та розробка AR-застосунку для візуалізації та взаємодії з тривимірними моделями у реальному просторі.

Результати кваліфікаційної роботи мають практичне застосування у сфері будівництва, архітектури, інтер'єрного дизайну та управління інфраструктурою розумного будинку. Використання методів моделювання AR дозволяє оптимізувати процеси візуалізації та управління приміщеннями, підвищити точність моделювання та покращити користувацький досвід.

1 СУЧАСНІ ПІДХОДИ ДО ПРОЄКТУВАННЯ РОЗУМНИХ БУДИНКІВ ТА ВИКОРИСТАННЯ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

У розділі наводиться аналіз процесу проєктування систем розумного будинку та використання технологій доповненої реальності. Проаналізовано основні компоненти, які забезпечують функціонування сучасних розумних будинків, а також ключові тенденції розвитку технологій у цій сфері. Розглянуто особливості застосування AR у проєктуванні приміщень та управлінні інфраструктурою розумного будинку. На основі проведеного аналізу сформовано загальні принципи, які відповідають сучасним вимогам і завданням дослідження, сформульовано мету та постановку завдання.

1.1 Концепція розумного будинку

Розвиток технологій і зростаюча урбанізація значно впливають на те, як люди організовують своє житло. Зміни у соціальній структурі суспільства, зокрема збільшення кількості міського населення, а також необхідність оптимізації ресурсів та підвищення рівня комфорту життя, спонукають до активного впровадження інноваційних рішень у побуті. Одним із найактуальніших трендів сучасності є концепція розумного будинку – інтелектуальної системи, що забезпечує автоматизований контроль та управління різноманітними функціями житла. Завдяки використанню сучасних технологій, таких як сенсори, виконавчі пристрої, мобільні застосунки та штучний інтелект, розумний будинок здатен адаптуватися до потреб користувачів, виконуючи рутинні завдання без їхнього втручання. Це сприяє не тільки підвищенню комфорту життя, але й дозволяє мінімізувати витрати часу на повсякденні справи, що особливо важливо у сучасному динамічному світі.

Ідея розумного будинку сформувалася на перетині кількох ключових

технологічних напрямів, зокрема Інтернету речей (IoT), автоматизованих систем управління, хмарних обчислень і штучного інтелекту (AI). Інтернет речей став основою для створення інтегрованої інфраструктури, де різні пристрої можуть обмінюватися даними між собою та виконувати завдання у координації. Автоматизовані системи управління дозволяють здійснювати контроль за параметрами середовища, такими як температура, вологість, освітлення, а також керувати побутовою технікою та безпекою житла. Хмарні обчислення забезпечують централізоване зберігання даних і можливість дистанційного доступу, що відкриває нові горизонти для інтеграції систем. Нарешті, штучний інтелект дозволяє створювати адаптивні сценарії поведінки системи, коли вона самостійно "вчиться" на основі даних про звички користувача та оптимізує свою роботу відповідно до його потреб.

Історія розвитку концепції розумного будинку бере свій початок ще в середині XX століття, коли з'явилися перші пристрої для автоматизації окремих функцій житла. У 1960-х роках були розроблені програмовані термостати, що дозволяли контролювати температуру в приміщенні. Наступним кроком стала поява аналогових систем безпеки у 1980-х роках, які використовували датчики руху та сигналізацію для попередження про вторгнення. Проте значні обмеження даних технологій, зокрема їхня локальність і відсутність інтеграції, не дозволяли створити повноцінну систему розумного будинку. Лише на початку XXI століття з розвитком цифрових технологій та бездротового зв'язку з'явилися можливості для об'єднання всіх компонентів у єдину мережу. Важливим етапом стало впровадження Інтернету речей, завдяки чому пристрої отримали можливість "спілкуватися" між собою через локальні та глобальні мережі. Поширення смартфонів і мобільних застосунків зробило управління системами розумного будинку доступним для широкого загалу, а розвиток голосових асистентів, таких як Amazon Alexa та Google Home, спростив взаємодію користувачів із системою[1].

Сучасний розумний будинок функціонує як єдина інтегрована

екосистема, у якій взаємодіють сенсори, виконавчі пристрої, центральний контролер і програмне забезпечення. Сенсори відповідають за збір інформації про стан приміщення: температурні датчики контролюють рівень тепла, датчики освітлення – рівень світла в кімнаті, а датчики руху виявляють присутність людей. Ці дані передаються центральному контролеру, який обробляє інформацію, приймає рішення та надсилає команди виконавчим пристроям. Виконавчі пристрої реалізують отримані команди, наприклад, вмикають або вимикають світло, регулюють роботу систем опалення чи відкривають двері. Програмне забезпечення надає користувачу можливість керувати системою як у ручному режимі, так і автоматично, через налаштування сценаріїв. У поєднанні ці компоненти забезпечують злагоджену роботу системи, що відповідає індивідуальним потребам користувача та створює максимально комфортне середовище для проживання.

Серед ключових переваг розумного будинку можна виділити автоматизацію побутових процесів, підвищення енергоефективності, покращення безпеки житла та можливість персоналізації функцій системи. Автоматизація дозволяє оптимізувати роботу пристроїв, мінімізуючи споживання ресурсів і витрати часу користувача. Наприклад, система може автоматично вимкнути освітлення в кімнаті, якщо датчики руху не фіксують присутності людей. Енергоефективність досягається шляхом точного контролю за використанням електроенергії та тепла, що дозволяє знизити витрати на комунальні послуги. Підвищення безпеки житла забезпечується завдяки інтеграції систем відеоспостереження, датчиків руху та інтелектуальних замків, що дозволяють контролювати доступ до будинку в реальному часі.

Таким чином, концепція розумного будинку є результатом інтеграції новітніх технологій і відповідає викликам сучасного світу. Вона відкриває нові можливості для створення комфортного, безпечного та енергоефективного житла, що адаптується до потреб користувача та сприяє

раціональному використанню ресурсів. Попри певні виклики, такі як висока вартість впровадження та питання безпеки даних, розумний будинок продовжує розвиватися, набуваючи дедалі більшого поширення у світі.

Система розумного будинку базується на інтеграції сенсорів, контролера, виконавчих пристроїв, комунікаційних протоколів та програмного забезпечення, що забезпечує комфорт, безпеку й оптимізацію ресурсів.

Сенсори фіксують дані про стан середовища (температура, вологість, рух) і передають їх контролеру, який приймає рішення на основі заданих алгоритмів. Контролери можуть бути локальними для мінімізації затримок або хмарними для дистанційного управління.

Виконавчі пристрої (термостати, замки, освітлення) виконують команди контролера, працюючи автономно чи в рамках сценаріїв. Комунікаційні протоколи (Zigbee, Wi-Fi) забезпечують передачу даних між компонентами системи[2].

Програмне забезпечення дозволяє керувати системою через мобільні додатки чи веб-інтерфейси, підтримуючи інтеграцію з голосовими помічниками (Google Assistant, Alexa) для зручного управління. На рис. 1.1 приведено архітектуру системи розумного будинку.

Комунікаційні протоколи забезпечують передачу даних між усіма компонентами системи. Вибір протоколу залежить від конкретних вимог системи, таких як масштаб мережі, енергоефективність та швидкість обміну інформацією. Найпоширенішими є протоколи Zigbee, Z-Wave, Wi-Fi та Bluetooth. Zigbee та Z-Wave є енергоефективними рішеннями, що підходять для сенсорів і виконавчих пристроїв, які працюють від батарейок. Wi-Fi забезпечує високу швидкість передачі даних, що особливо важливо для потокового відео та систем відеоспостереження.

Програмне забезпечення є інтерфейсом взаємодії користувача із системою розумного будинку. Воно дозволяє налаштовувати сценарії автоматизації, контролювати роботу пристроїв у режимі реального часу та

отримувати звіти про стан системи. Програмне забезпечення, зазвичай, реалізується у вигляді мобільних застосунків або веб-інтерфейсів, які забезпечують доступ до системи з будь-якої точки світу. Інтеграція голосових помічників, таких як Google Assistant або Amazon Alexa, дозволяє керувати функціями будинку за допомогою голосових команд, що значно підвищує зручність використання.

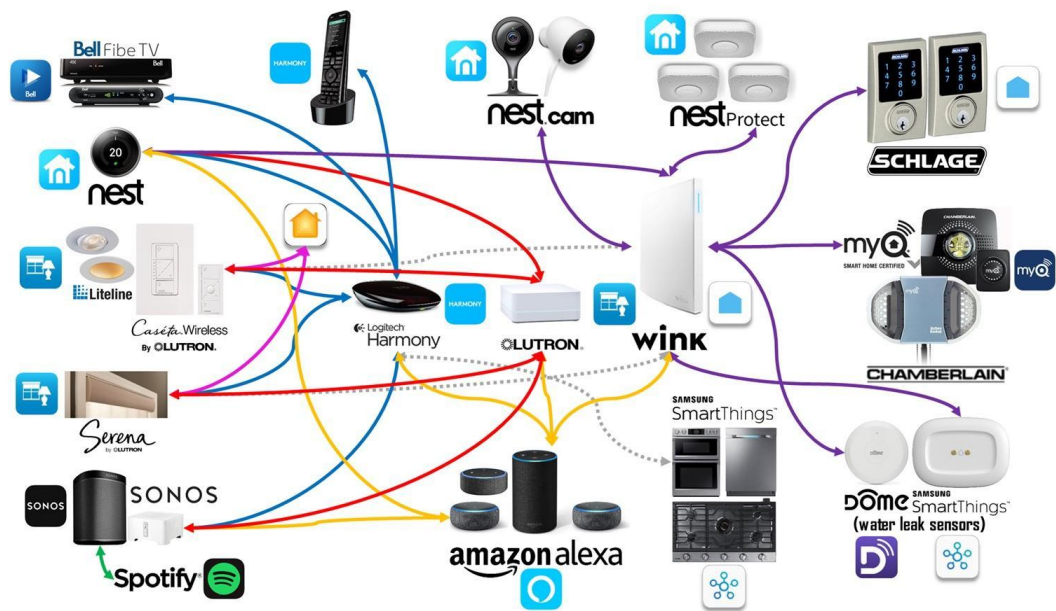


Рисунок 1.1 - Архітектура системи розумного будинку.

Впровадження технологій розумного будинку відкриває значні можливості для покращення якості життя, зокрема підвищення комфорту, енергоефективності та безпеки. Автоматизація рутинних завдань, таких як управління освітленням чи клімат-контролем, дозволяє зекономити час користувачів. Водночас централізоване управління через смартфон або голосового помічника забезпечує гнучкість і зручність.

Системи розумного будинку сприяють раціональному використанню ресурсів, зменшуючи споживання енергії на 20–30% завдяки оптимізації роботи термостатів, освітлення та інших пристроїв. Безпека житла значно зростає завдяки розумним замкам, камерам відеоспостереження та датчикам

руху чи диму, які своєчасно сповіщають про загрози.

Попри переваги, існують і недоліки. Значні початкові витрати на впровадження, залежність від стабільного інтернет-з'єднання та ризики для конфіденційності даних є основними викликами. Використання хмарних технологій створює ризик витоку інформації, що потребує надійного шифрування та регулярного оновлення системи.

Таким чином, технології розумного будинку мають великий потенціал, але для їх успішного впровадження необхідно враховувати виклики, пов'язані з вартістю, безпекою та стабільністю роботи.

1.2 Технології доповненої реальності та їх основні характеристики

Доповнена реальність (AR) – це технологія, яка дозволяє інтегрувати цифрові об'єкти в реальний світ у режимі реального часу. На відміну від віртуальної реальності (VR), яка повністю занурює користувача у штучно створене середовище, AR доповнює існуючу фізичну реальність віртуальними елементами, зберігаючи зв'язок із реальним світом. Такі технології знайшли широке застосування у різних галузях, від архітектури та дизайну до медицини, освіти та ігрової індустрії.

Сутність технологій доповненої реальності полягає у поєднанні кількох ключових складових, які забезпечують їх функціонування: апаратного забезпечення, програмних платформ, алгоритмів обробки даних і взаємодії з користувачем. Кожен із цих елементів відіграє важливу роль у створенні цілісного досвіду доповненої реальності.

Одним із ключових компонентів AR є апаратне забезпечення, яке включає пристрої введення та виведення, такі як камери, дисплеї, сенсори та носимі пристрої. Сучасні пристрої для роботи з AR можуть бути представлені у вигляді смартфонів, планшетів, окулярів доповненої реальності (наприклад, Microsoft HoloLens або Magic Leap) та інших носимих пристроїв. Камери та сенсори дозволяють фіксувати фізичне середовище,

відстежувати рухи користувача та його взаємодію з об'єктами реального світу. Наприклад, камери високої роздільної здатності збирають дані про навколишнє середовище, а гіроскопи та акселерометри визначають орієнтацію пристрою у просторі.

Ще одним важливим аспектом є програмне забезпечення, яке забезпечує обробку зібраних даних, розпізнавання об'єктів та відображення цифрових елементів на екрані пристрою. AR-додатки побудовані на основі спеціалізованих платформ і бібліотек, таких як ARKit (Apple), ARCore (Google) або Vuforia. Ці платформи забезпечують доступ до базових функцій, таких як відстеження поверхонь, розпізнавання об'єктів і керування 3D-моделями. Наприклад, ARKit використовує технології машинного зору для аналізу зображень із камери та побудови карти навколишнього середовища. Завдяки таким платформам розробники можуть створювати інтерактивні AR-додатки з мінімальними затратами часу та ресурсів.

Алгоритми обробки даних є критично важливим компонентом технологій AR. Однією з ключових задач доповненої реальності є точне поєднання віртуальних об'єктів із реальним середовищем, що вимагає високоточного визначення просторового положення пристрою та об'єктів у кадрі. Для цього використовуються алгоритми SLAM (Simultaneous Localization and Mapping), які одночасно локалізують пристрій у просторі та створюють карту навколишнього середовища. Ці алгоритми дозволяють визначати поверхні, такі як стіни, стелі чи підлоги, і коректно розташовувати віртуальні об'єкти у просторі. Крім того, технології комп'ютерного зору забезпечують розпізнавання реальних об'єктів і їх інтеграцію у сценарії доповненої реальності.

Для забезпечення інтерактивності та зручності користування велике значення мають інтерфейси взаємодії з користувачем. У системах AR зазвичай використовуються мультимодальні інтерфейси, які поєднують традиційне сенсорне введення (дотики, свайпи) із голосовим управлінням, жестами або навіть відстеженням погляду. Це дозволяє користувачам

інтуїтивно взаємодіяти з віртуальними елементами, створюючи унікальний досвід. Наприклад, окуляри доповненої реальності оснащені технологіями розпізнавання жестів, що дозволяє переміщувати або змінювати розміри об'єктів у просторі за допомогою рук.

Доповнена реальність також значно виграє від використання хмарних технологій, які дозволяють зберігати великі обсяги даних і виконувати складні обчислення на віддалених серверах. Завдяки цьому пристрої AR можуть залишатися легкими та портативними, оскільки основна обчислювальна робота виконується у хмарі. Наприклад, складні 3D-моделі чи текстури можуть передаватися на пристрій у реальному часі, що значно знижує вимоги до його обчислювальної потужності.

Таким чином, технології доповненої реальності є комплексним поєднанням апаратного та програмного забезпечення, алгоритмів обробки даних та інтерфейсів взаємодії з користувачем. Їх основні характеристики включають інтеграцію віртуальних об'єктів у реальний світ, використання сучасних алгоритмів для точного розташування елементів, а також можливість інтуїтивної взаємодії з ними. Усе це робить AR потужним інструментом для моделювання, який відкриває нові горизонти для вирішення складних завдань у багатьох галузях.

Технології доповненої реальності (AR) стають невід'ємною частиною сучасного моделювання, відкриваючи нові можливості для інтерактивності, візуалізації та точності. AR дозволяє накладати цифрові моделі на реальне середовище, що особливо корисно у проєктуванні приміщень розумного будинку. Ця технологія надає можливість оцінювати проєкти у масштабі 1:1, прогнозувати виклики під час реалізації та зменшувати ймовірність помилок.

Унікальною особливістю AR є інтерактивний досвід, де користувачі можуть змінювати розташування об'єктів за допомогою жестів чи відстеження рухів. Це підвищує залученість і забезпечує новий рівень точності під час роботи з моделями. Багатокористувацька співпраця, що підтримується хмарними технологіями, дозволяє координувати глобальні

проекти у реальному часі, забезпечуючи узгодженість рішень.

Інтеграція AR з технологіями штучного інтелекту та машинного навчання додає нових можливостей, таких як автоматизація сценаріїв та виявлення конструктивних недоліків. Це робить AR незамінним інструментом для створення адаптивних і функціональних рішень.

Доповнена реальність у моделюванні не тільки спрощує візуалізацію складних концепцій, але й дозволяє користувачам ефективніше співпрацювати та приймати рішення на основі точних даних. Її інтерактивність і здатність до інтеграції з іншими технологіями роблять AR незамінним інструментом для проектування майбутнього, відкриваючи нові горизонти в архітектурі, дизайні та інженерії.

Технології доповненої реальності відкривають значні можливості для покращення процесів проектування та управління в різних галузях. Завдяки своїй здатності інтегрувати віртуальні елементи у фізичний світ, AR забезпечує більш точну візуалізацію, спрощує аналіз складних структур і дозволяє підвищити ефективність робочих процесів. Застосування AR охоплює такі сфери, як архітектура, будівництво, управління інфраструктурою, дизайн інтер'єрів, промислове виробництво та освіта.

Однією з ключових галузей застосування AR є архітектурне проектування. Технології доповненої реальності дозволяють архітекторам створювати детальні тривимірні моделі будівель та візуалізувати їх у реальному середовищі ще до початку будівельних робіт. Це дає змогу замовникам та інженерам оцінити зовнішній вигляд, масштаби та особливості конструкції будівлі в реальному часі. Зокрема, AR дозволяє проводити аналіз взаємодії будівлі з навколишнім середовищем, що допомагає запобігти можливим проблемам, пов'язаним із ландшафтом, освітленням або іншими факторами. У результаті проекти стають більш деталізованими, а процес планування – значно точнішим.

Будівництво є ще однією сферою, де AR демонструє високу ефективність. Доповнена реальність використовується для контролю за

будівельними процесами, порівняння проєктних планів із фактичною реалізацією, а також для візуалізації прихованих елементів конструкцій. Наприклад, за допомогою AR-браузерів або спеціальних окулярів будівельники можуть накладати цифрові плани на фізичний об'єкт і перевіряти точність виконання робіт. Це зменшує ймовірність помилок, підвищує продуктивність та мінімізує витрати, пов'язані з переробкою.

У дизайні інтер'єрів офісних приміщень AR дозволяє візуалізувати розташування меблів, освітлювальних приладів та інших елементів декору у приміщенні ще до їх фактичного встановлення (рис. 1.2) Це особливо корисно для створення індивідуальних інтер'єрних рішень, оскільки замовник може бачити кінцевий результат у реальному масштабі та вносити зміни на етапі планування. Завдяки інтерактивним AR-застосункам користувачі можуть експериментувати з різними варіантами дизайну, вибирати кольорові гами та тести текстур у режимі реального часу, що значно спрощує процес прийняття рішень.

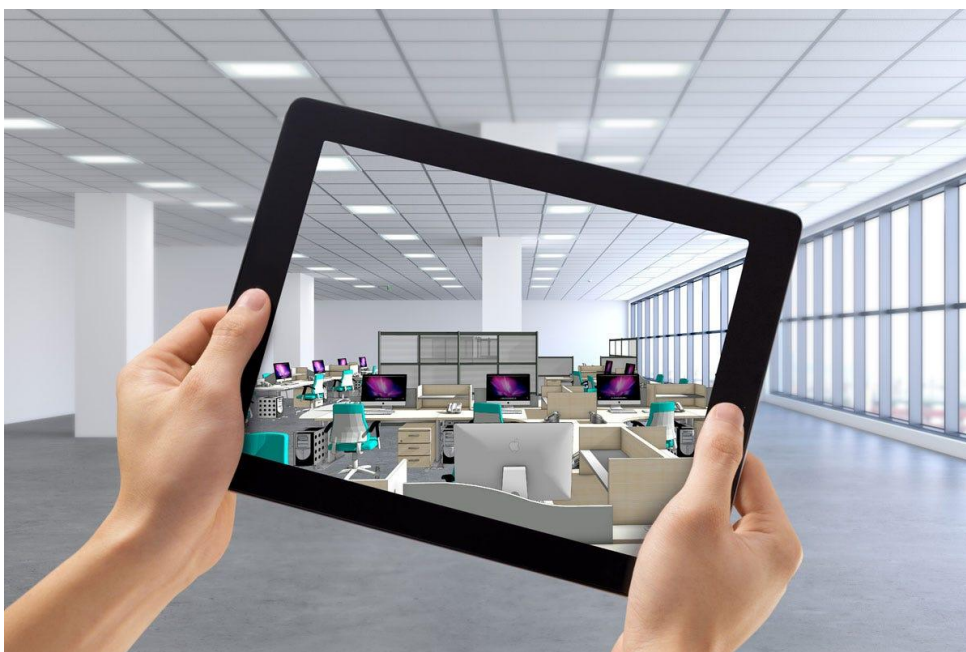


Рисунок 1.2 – Використання AR у моделюванні інтер'єрів офісних приміщень

Промислове виробництво та інженерія також активно використовують AR у процесах моделювання, планування та обслуговування обладнання. Доповнена реальність дозволяє створювати інтерактивні інструкції для складного обладнання, що полегшує його збірку, налаштування або ремонт. Наприклад, технічний персонал може використовувати AR-окуляри для перегляду покрокових інструкцій з обслуговування обладнання без необхідності відволікатися на традиційні посібники. Крім того, AR дозволяє проводити навчання працівників у безпечному віртуальному середовищі, де вони можуть відпрацьовувати навички роботи з обладнанням.

Освітня галузь активно застосовує AR для моделювання навчальних процесів (рис. 1.3). Використання доповненої реальності в освіті дозволяє студентам та учням взаємодіяти з віртуальними моделями, що відображають реальні об'єкти, процеси або явища. Це особливо ефективно у технічних дисциплінах, де віртуальні моделі можуть замінити фізичні макети або лабораторні установки. Наприклад, у медичній освіті студенти можуть досліджувати анатомію людини за допомогою інтерактивних AR-моделей, що значно спрощує вивчення складних структур організму.

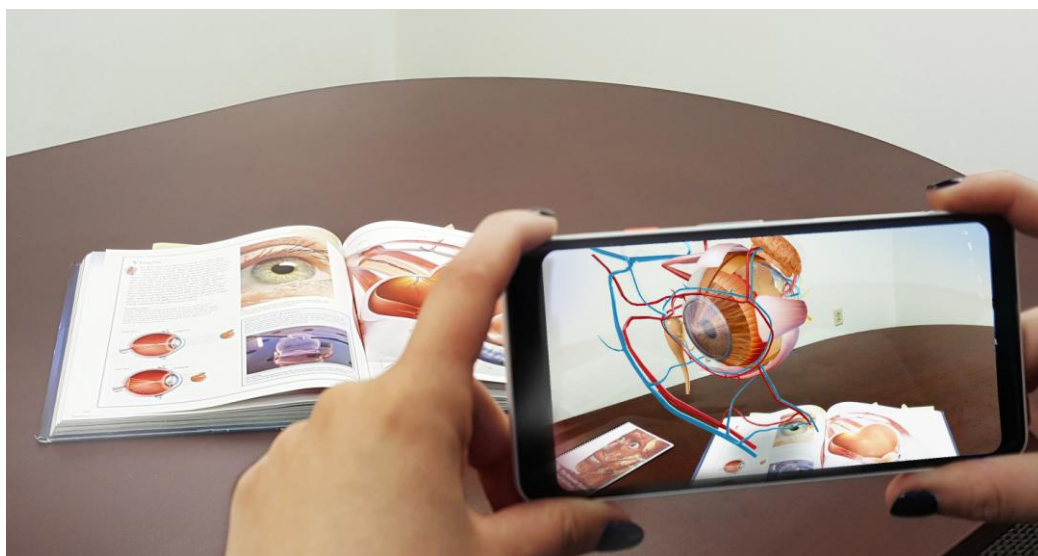


Рисунок 1.3 – Доповнена реальність в освітньому процесі

Сфери застосування AR у проєктуванні та управлінні постійно розширюються завдяки технологічному прогресу і зростанню потреб у точних та інноваційних рішеннях. Завдяки своїй універсальності, доповнена реальність дозволяє оптимізувати робочі процеси, покращити візуалізацію проєктів та забезпечити глибоку інтеграцію в існуючі інфраструктури. З огляду на це, AR має значний потенціал для подальшого розвитку й впровадження в ключових галузях, що потребують високого рівня точності та ефективності.

1.3 Сучасні тренди впровадження AR у розумні будинки

Впровадження технологій доповненої реальності (AR) у системи розумного будинку є перспективним напрямом, який поєднує автоматизацію житлових приміщень із можливостями візуалізації та інтерактивної взаємодії. Доповнена реальність дозволяє користувачам оптимізувати управління побутовими пристроями, візуалізувати інфраструктуру будинку та спрощувати процеси ремонту й обслуговування. Проте цей процес не позбавлений викликів, які впливають на темпи впровадження AR у розумні системи. Одночасно із викликами спостерігається активний розвиток технологічних трендів, які сприяють підвищенню функціональності та доступності цих рішень.

Одним із ключових викликів є висока вартість впровадження технологій AR. Створення повноцінної системи з використанням доповненої реальності потребує значних інвестицій у обладнання, програмне забезпечення та інфраструктуру. Пристрої, які підтримують AR, такі як окуляри HoloLens чи Magic Leap, залишаються досить дорогими, що обмежує їх доступність для масового споживача. Крім того, інтеграція AR з уже існуючими системами "розумного будинку" вимагає адаптації обладнання та розробки індивідуальних рішень, що також підвищує загальні витрати.

Ще одним викликом є технічні обмеження пристроїв AR. Незважаючи

на швидкий розвиток технологій, сучасні AR-пристрої досі мають обмежену продуктивність, що впливає на якість відображення 3D-об'єктів у реальному середовищі. Низька роздільна здатність візуалізації, обмежений кут огляду та необхідність постійного підключення до інтернету є основними технічними бар'єрами. Такі обмеження особливо критичні для складних систем розумного будинку, де потрібна точна інтеграція віртуальних об'єктів із реальним середовищем.

Особливим питанням є конфіденційність та безпека даних. Системи AR потребують великої кількості інформації про середовище користувача, що включає дані з камер, датчиків та пристроїв, інтегрованих у розумний будинок. Зберігання та обробка цих даних у хмарних сервісах створюють потенційні ризики витоку інформації або несанкціонованого доступу. Для вирішення цієї проблеми необхідно розробляти більш захищені платформи та впроваджувати сучасні методи шифрування й аутентифікації.

Попри виклики, сучасні тренди демонструють значний прогрес у впровадженні AR у розумні будинки. Одним із провідних трендів є інтеграція AR із системами штучного інтелекту (AI). Штучний інтелект дозволяє автоматизувати процеси управління та адаптувати AR-сценарії до потреб користувача. Наприклад, система може "навчатися" на основі поведінки мешканця, автоматично вмикати освітлення чи налаштовувати мікроклімат у кімнаті з урахуванням його вподобань.

Ще одним важливим трендом є розвиток легких і портативних пристроїв для AR. Сучасні смартфони та планшети вже підтримують функції доповненої реальності завдяки платформам ARKit і ARCore, що робить цю технологію більш доступною для широкого кола користувачів. Подальший розвиток окулярів AR із покращеною продуктивністю та зниженням вартості обладнання сприятиме поширенню технологій у системах розумного будинку.

Значний інтерес викликає інтеграція AR із Інтернетом речей (IoT). Поєднання AR та IoT дозволяє створити динамічні сценарії управління

пристроями на основі візуалізації їх роботи у реальному часі. Наприклад, користувач може отримати інформацію про стан побутових приладів або комунікаційних систем, просто навівши камеру смартфона на об'єкт. Це значно спрощує управління інфраструктурою будинку та підвищує ефективність процесів.

Попри наявні виклики, сучасний розвиток технологій робить доповнену реальність ключовим інструментом у впровадженні інноваційних рішень для "розумних будинків". Подальша інтеграція AR із системами штучного інтелекту, Інтернетом речей та розвиток доступного обладнання сприяють створенню більш зручних і функціональних середовищ, що відповідають вимогам користувачів і технологічних реалій майбутнього.

1.4 Мета та постановка завдання

Метою кваліфікаційної роботи є дослідження методів моделювання AR-сцен та розробка AR-застосунку для моделювання приміщень розумного будинку з використанням сучасних технологій доповненої реальності.

Об'єктом дослідження є технології доповненої реальності, що використовуються для моделювання та візуалізації приміщень розумного будинку.

Предметом дослідження є методи моделювання AR-сцен, інструментальні засоби для розробки AR-застосунків.

Для досягнення сформульованої мети визначені наступні задачі:

- огляд концепції розумного будинку та аналіз технологій доповненої реальності для моделювання приміщень;
- аналіз інструментів для розробки AR-застосунків;
- дослідження методів моделювання AR-сцен приміщень;
- розробка AR-застосунку для моделювання приміщень розумного будинку;
- вибір технологій, інструментів та бібліотек для розробки AR-

застосунок;

- розробка алгоритму роботи AR-застосунок та його архітектури;
- розробка програмного забезпечення AR-застосунок та його опис;
- тестування розробленого AR-застосунок.

2 АНАЛІЗ ТЕХНОЛОГІЙ І МЕТОДІВ МОДЕЛЮВАННЯ ПРИМІЩЕНЬ У СИСТЕМАХ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

У розділі розглядаються сучасні технології та методи моделювання приміщень за допомогою технологій доповненої реальності, їх структура, принципи роботи та особливості застосування.

2.1 Аналіз сучасних інструментів для створення AR-застосунків

Технології доповненої реальності AR постійно розвиваються й активно впроваджуються у різні сфери людської діяльності, зокрема архітектуру, дизайн інтер'єрів та системи розумного будинку. Основою будь-якого AR-рішення є інструменти та платформи, що забезпечують інтеграцію віртуальних об'єктів у реальне середовище. Вибір технології визначається низкою ключових факторів, серед яких технічні вимоги проєкту, рівень продуктивності системи та простота інтеграції для розробників і кінцевих користувачів.

На сучасному етапі розвитку AR можна виділити три популярні інструменти, які мають значний вплив на галузь завдяки своїй функціональності, доступності та продуктивності. Одним із них є Google Model Viewer – бібліотека, яка надає розробникам простий і ефективний спосіб впровадження 3D-моделей та AR-функціоналу у вебсередовище. Її ключовою особливістю є підтримка стандартних форматів 3D-моделей, таких як glTF/GLB для Android та USDZ для iOS, що забезпечує кросплатформенність і простоту налаштування.

Іншим важливим інструментом є Unity – універсальна платформа для створення інтерактивних AR-застосунків з підтримкою високоякісної 3D-графіки, анімації та складних обчислень. Unity інтегрується з нативними рішеннями, такими як ARKit від Apple та ARCore від Google, що робить її

одним із найпотужніших інструментів для розробки AR- застосунків. Попри складність навчання та високі вимоги до ресурсів, Unity забезпечує широкі можливості для реалізації функціонально насичених систем.

Третім інструментом, що заслуговує на увагу, є WebXR API у поєднанні з AR.js. Це рішення базується на сучасних вебтехнологіях, дозволяючи створювати AR-застосунки для браузерів без необхідності встановлення додаткового програмного забезпечення. WebXR API надає розробникам гнучкий доступ до функцій доповненої та віртуальної реальності, тоді як AR.js спрощує інтеграцію AR-сцен, забезпечуючи сумісність із широким спектром пристроїв.

2.1.1 Google Model Viewer

Google Model Viewer – це бібліотека з відкритим вихідним кодом, розроблена компанією Google для інтеграції 3D-моделей та функціоналу доповненої реальності (AR) у вебсередовищі. Вона є одним із ключових інструментів, що реалізують концепцію WebXR API, забезпечуючи високу якість рендерингу та простоту інтеграції для веброзробників. Створення AR-сцен на основі Model Viewer відбувається завдяки комбінації сучасних вебтехнологій, серед яких WebGL, Three.js і WebXR API, кожна з яких відіграє важливу роль у забезпеченні функціональності та продуктивності бібліотеки.

Архітектурною та технологічною основою для роботи Model Viewer є WebGL, що є низькорівневим API для рендерингу тривимірної графіки у веббраузері. WebGL забезпечує прямий доступ до GPU для обробки геометрії та текстур 3D-об'єктів. Завдяки цьому Model Viewer може виконувати апаратно прискорений рендеринг навіть на мобільних пристроях, забезпечуючи високу швидкість обробки та плавність взаємодії.

У процесі розробки Google Model Viewer використовує Three.js як проміжну бібліотеку для спрощення роботи з WebGL. Three.js забезпечує

більш високорівневий підхід до створення сцени, роботи з матеріалами, камерами та джерелами світла. При ініціалізації бібліотеки Model Viewer автоматично створює сцену, яка включає камеру, світло та обраний 3D-об'єкт.

Для завантаження 3D-моделей Google Model Viewer підтримує стандартизовані формати, серед яких:

- glTF (GL Transmission Format) – формат, що оптимізований для швидкого завантаження та обробки 3D-даних. Він є основним для Android-пристроїв завдяки його ефективності у передачі текстур, геометрії та анімацій;

- USDZ – формат, що підтримується екосистемою Apple і використовується для AR-функціоналу через Quick Look.

На рис 2.1 зображено приклад моделі у форматі USDZ.



Рисунок 2.1 – Модель у форматі USDZ

Важливою особливістю архітектури Model Viewer є підтримка PBR

(Physically Based Rendering) – методу рендерингу, що забезпечує фотореалістичну візуалізацію матеріалів за допомогою фізично коректного освітлення. Використання HDRI (High Dynamic Range Imaging) для налаштування освітлення дозволяє 3D-моделям виглядати природно за різних умов, автоматично адаптуючись до сцени.

Функціонал доповненої реальності в Model Viewer реалізується завдяки WebXR API, який надає веббраузеру доступ до сенсорів і камер пристрою. Під час активації режиму AR, Model Viewer ініціалізує WebXR-сесію, аналізує навколишнє середовище та розміщує віртуальний об'єкт у фізичному просторі. Процес включає кілька етапів:

- виявлення площини: WebXR API визначає поверхні у полі зору камери, на яких можна розташувати об'єкт;
- інтеграція освітлення: для забезпечення реалістичності об'єкт адаптується до умов освітлення навколишнього середовища;
- відображення та взаємодія: користувач може наближати, обертати та переміщати об'єкт за допомогою сенсорного управління або жестів.

Даний підхід забезпечує користувачеві інтуїтивно зрозумілу взаємодію з AR-сценами, одночасно підтримуючи високу продуктивність за рахунок апаратного прискорення.

Google Model Viewer активно використовує механізми оптимізації для роботи у вебсередовищі. Завдяки підтримці Level of Detail (LOD), бібліотека автоматично регулює деталізацію 3D-об'єктів залежно від відстані камери до моделі. Додатково застосовується фонове завантаження текстур і моделей, що мінімізує затримки та забезпечує плавне відображення сцени.

Особливу увагу приділено сумісності бібліотеки з мобільними пристроями. Завдяки використанню апаратного рендерингу та оптимізованої роботи з GPU, Model Viewer може функціонувати навіть на пристроях із обмеженими ресурсами без значного зниження продуктивності.

Основною перевагою Google Model Viewer є його простота інтеграції у вебпроекти. За допомогою HTML-тегу `<model-viewer>` розробник може

додати 3D-модель із мінімальним налаштуванням, що робить цей інструмент ідеальним для демонстрацій AR-функціоналу у браузері.

До переваг Google Model Viewer належать:

- підтримка стандартних 3D-форматів (glTF, USDZ);
- висока продуктивність завдяки WebGL;
- сумісність із WebXR для реалізації AR-функціоналу.

Серед обмежень можна виділити:

- неможливість створення складних AR-сцен із динамічними об'єктами;
- залежність від підтримки WebXR API у браузерах;
- обмежений набір інструментів для кастомізації взаємодії.

У підсумку Google Model Viewer є ефективним інструментом для швидкої та якісної інтеграції 3D-моделей у вебсередовище з підтримкою AR. Завдяки поєднанню WebGL, Three.js та WebXR API бібліотека забезпечує високу продуктивність та простоту впровадження, що робить її популярним рішенням для веборієнтованих AR-застосунків.

2.1.2 Unity

Unity – це одна з найпоширеніших платформ для розробки застосунків із доповненою реальністю (AR), яка поєднує в собі гнучкість, продуктивність і універсальність. Її багатофункціональність дозволяє створювати інтерактивні середовища, які успішно застосовуються у сфері освіти, архітектури, дизайну інтер'єрів, медицини та багатьох інших галузях. Unity надає розробникам потужний набір інструментів для моделювання, рендерингу, анімації та інтерактивності, що дозволяє створювати додатки з високоякісною графікою та складними механіками взаємодії.

На рис. 2.2 зображений користувацький інтерфейс платформи Unity.

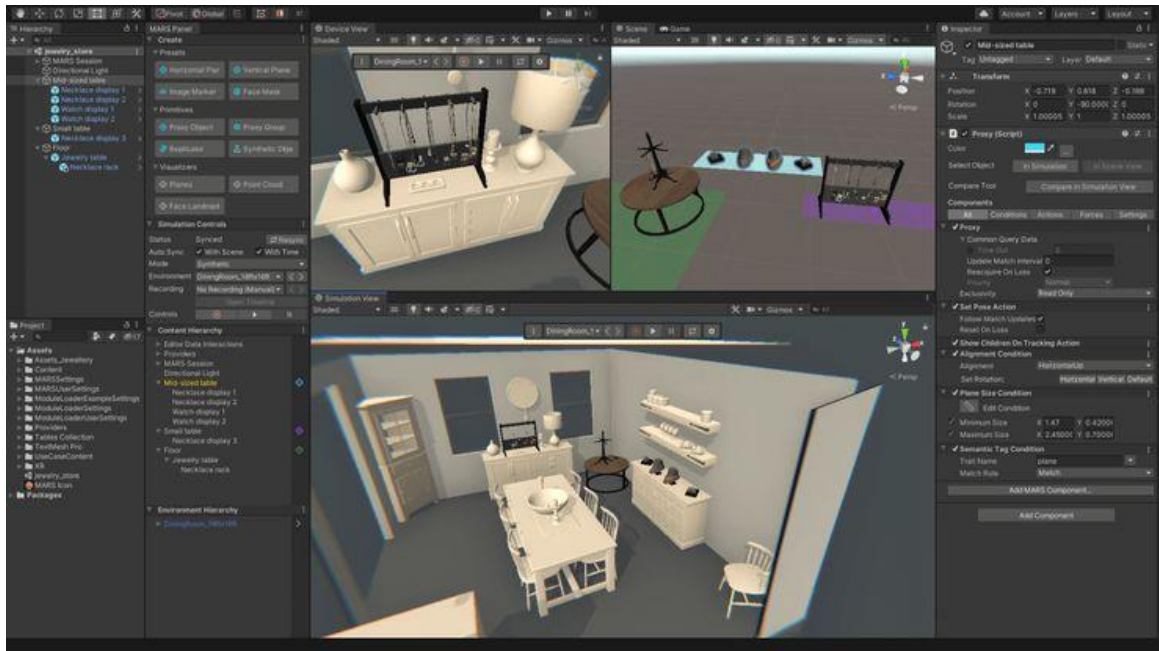


Рисунок 2.2 – Користувачський інтерфейс Unity

Основою платформи Unity є Unity Engine, система фізичного моделювання, що забезпечує всі аспекти роботи застосунку, включаючи рендеринг графіки, обробку фізики, обчислення анімацій і інтерактивність. Unity Engine підтримує різні графічні API, такі як OpenGL, Vulkan, DirectX і Metal, що дозволяє адаптувати застосунки до різних пристроїв і платформ[9].

Для AR-застосунків зазвичай використовується Universal Render Pipeline, який забезпечує баланс між продуктивністю та якістю графіки, особливо для мобільних пристроїв. Підтримка Physically Based Rendering (PBR) дозволяє досягати високого рівня реалістичності матеріалів за рахунок правильного відображення властивостей освітлення, текстур і віддзеркалень. Використання HDRI-освітлення додає моделям фотореалістичності, що особливо важливо для інтеграції віртуальних об'єктів у реальне середовище.

Важливим компонентом Unity є модуль фізичного моделювання PhysX, розроблений NVIDIA, що відповідає за симуляцію фізичних властивостей, таких як гравітація, тертя, зіткнення та еластичність. PhysX дозволяє забезпечити реалістичну взаємодію між об'єктами у тривимірному просторі. У контексті AR-застосунків це означає, що віртуальні об'єкти можуть

адекватно реагувати на фізичні сили та взаємодіяти з реальними поверхнями.

Анімація у Unity реалізована за допомогою систем контролю анімації і лінії часу. Контроль анімацій дозволяє створювати складні сценарії поведінки об'єктів, прив'язуючи їх до конкретних подій або станів, тоді як лінія часу забезпечує синхронізацію анімацій у часі. Для AR-застосунків це відкриває можливості створення інтерактивних сцен, де об'єкти адаптуються до дій користувача або змін у навколишньому середовищі.

Unity має потужну підтримку AR-фреймворків, таких як ARKit, ARCore і Vuforia. ARKit, розроблений компанією Apple, забезпечує точний трекінг обличчя, виявлення площин і освітлення для пристроїв iOS. ARCore, створений Google, орієнтований на Android-пристрої та пропонує функції виявлення поверхонь, трекінгу рухів і SLAM. Vuforia є кросплатформним рішенням, яке підтримує як маркерні, так і безмаркерні методи трекінгу, а також розпізнавання об'єктів і тексту.

Процес роботи Unity у контексті AR-застосунків починається з ідентифікації та аналізу фізичного середовища. Завдяки використанню даних із камер, акселерометрів і гіроскопів пристрою доповненої реальності, платформа створює тривимірну карту навколишнього простору. Алгоритми SLAM дозволяють визначити позицію пристрою та віртуальних об'єктів у реальному просторі. Unity автоматично налаштовує адаптивне освітлення, яке враховує умови реального середовища, забезпечуючи гармонійне поєднання віртуальних і реальних об'єктів.

Серед ключових переваг Unity можна виділити його кросплатформенність, гнучкість, високу продуктивність і розвинену екосистему, що включає тисячі плагінів і бібліотек для розширення базових можливостей платформи. Водночас до обмежень платформи належать її високі вимоги до апаратних ресурсів, складність освоєння для початківців і великий обсяг кінцевих додатків через значну кількість інтегрованих ресурсів.

Unity залишається одним із провідних інструментів для створення AR-

додатків завдяки своїй здатності поєднувати простоту інтеграції з передовими технологіями. Незважаючи на певні обмеження, платформа забезпечує стабільну основу для розробки інтерактивних рішень, що відповідають сучасним технологічним вимогам.

2.1.3 WebXR API та AR.js

WebXR API є сучасним стандартом для створення застосунків із доповненою та віртуальною реальністю у вебсередовищі, який поєднує в собі можливості роботи з тривимірною графікою, сенсорами пристрою та інтеграції віртуальних об'єктів у реальне середовище. Даний API замінив свого попередника WebVR, забезпечуючи ширший функціонал і сумісність із багатьма пристроями, включаючи смартфони, планшети, настільні комп'ютери та окуляри доповненої реальності. Основною перевагою WebXR API є можливість запуску застосунків через веб-браузер, що усуває необхідність у встановленні додаткового програмного забезпечення.

Одним із ключових аспектів WebXR API є інтеграція з сенсорами пристрою, такими як камери, гіроскопи, акселерометри та сенсори освітлення. Ця інтеграція дозволяє API аналізувати фізичне середовище в реальному часі та створювати тривимірні карти простору. Наприклад, камера пристрою може фіксувати поверхні, а алгоритми комп'ютерного зору обробляють ці дані, визначаючи їхню геометрію та орієнтацію. Гіроскоп і акселерометр забезпечують інформацію про положення пристрою, дозволяючи стабільно відображати віртуальні об'єкти навіть при зміні кута огляду.

WebXR API використовує WebGL як основне ядро для рендерингу 3D-графіки. WebGL є низькорівневим API, який забезпечує апаратно-прискорений доступ до GPU для обробки графічних даних. Завдяки цьому WebXR API може обробляти складні тривимірні сцени з високою продуктивністю, навіть на пристроях із обмеженими ресурсами. WebGL працює з популярними форматами 3D-моделей, такими як glTF/GLB, які

оптимізовані для швидкого завантаження й інтеграції в браузері[3].

Важливою частиною WebXR API є підтримка PBR, яка дозволяє створювати реалістичні матеріали з урахуванням освітлення, текстур, віддзеркалень і фізичних властивостей. Це забезпечує гармонійне поєднання віртуальних об'єктів із реальним середовищем, що особливо важливо для створення AR-застосунків.

Ще одним важливим компонентом WebXR API є алгоритми SLAM, які забезпечують точне визначення положення пристрою у просторі та одночасне створення карти фізичного середовища. SLAM дозволяє створювати стабільні AR-сцени, навіть якщо пристрій змінює своє положення або орієнтацію.

AR.js є бібліотекою з відкритим вихідним кодом, створеною для спрощення роботи з WebXR API та інтеграції AR-функціоналу в веб-застосунки. Вона дозволяє розробникам реалізовувати складні AR-сцени з мінімальними зусиллями, використовуючи високорівневі інструменти. Основною особливістю AR.js є її підтримка як маркерного, так і безмаркерного трекінгу.

Маркерний трекінг у AR.js базується на використанні спеціальних зображень або QR-кодів, які слугують точками прив'язки для віртуальних об'єктів. Камера пристрою фіксує маркер, а алгоритми комп'ютерного зору визначають його геометрію та положення у просторі. Безмаркерний трекінг використовує алгоритми SLAM, які дозволяють будувати тривимірну карту фізичного середовища на основі даних із камери пристрою.

AR.js також підтримує рендеринг 3D-об'єктів у форматах glTF/GLB. Ці формати оптимізовані для швидкого завантаження текстур і моделей, що забезпечує плавність роботи застосунків навіть на пристроях із обмеженими ресурсами. Крім того, бібліотека використовує PBR для створення фотореалістичних матеріалів, що додає віртуальним об'єктам природного вигляду.

WebXR API та AR.js працюють у тісному зв'язку, забезпечуючи

розробникам потужний набір інструментів для створення веборієнтованих AR-застосунків. WebXR API забезпечує доступ до низькорівневих функцій, таких як обробка сенсорних даних і рендеринг, тоді як AR.js спрощує реалізацію цих функцій, надаючи готові рішення для інтеграції AR-функціоналу.

Серед переваг названих технологій:

- доступність завдяки запуску застосунків через веб-браузер, без необхідності встановлення додаткового програмного забезпечення;
- сумісність із широким спектром пристроїв та платформ;
- продуктивність за рахунок використання WebGL для апаратно-прискореного рендерингу;
- гнучкість у підтримці маркерного та безмаркерного трекінгу.

Проте розглянуті технології мають і свої обмеження:

- залежність від сумісності версій браузера та пристрою;
- висока продуктивність завдяки WebGL та PBR;
- обмеження продуктивності на пристроях із низькими обчислювальними ресурсами;
- менші можливості для кастомізації порівняно з нативними рішеннями.

2.1.4 Порівняння інструментів для створення AR-застосунків

Для обґрунтованого вибору відповідного рішення для розробки AR-застосунків важливо провести порівняльний аналіз інструментів, розглянутих у попередніх підрозділах: Google Model Viewer, Unity та WebXR API у поєднанні з AR.js. Кожен із цих інструментів має унікальні особливості, які впливають на їхню продуктивність, гнучкість та зручність використання.

Google Model Viewer є простим і зручним рішенням для швидкої інтеграції 3D-моделей у вебсередовище. Завдяки підтримці форматів glTF та USDZ бібліотека забезпечує високий рівень сумісності з сучасними браузерами й мобільними пристроями. Проте її функціональність обмежена у

створенні складних AR-сцен, оскільки вона орієнтована на базовий функціонал для відображення та взаємодії з моделями.

Unity пропонує широкий спектр можливостей для створення складних AR-застосунків із високим рівнем продуктивності та інтерактивності. Завдяки інтеграції з ARKit, ARCore та Vuforia, платформа дозволяє створювати нативні застосунки, які відзначаються стабільністю та реалістичністю. Однак високі вимоги до ресурсів і значний обсяг застосунків можуть обмежити використання Unity для проєктів, орієнтованих на вебкористувачів.

WebXR API у поєднанні з AR.js забезпечує оптимальне рішення для створення веборієнтованих AR-застосунків. Це рішення дозволяє запускати застосунки безпосередньо через браузер, забезпечуючи доступність для широкої аудиторії. Попри це, залежність від продуктивності пристроїв та обмежена кастомізація функціоналу можуть бути викликами для розробників. У таблиці приведено порівняльний аналіз ключових характеристик інструментів для розробки AR-застосунків.

Таблиця 2.1 – Порівняльний аналіз ключових характеристик інструментів для розробки AR-застосунків.

Критерій	Model Viewer	Unity	WebXR API та AR.js
Цільова аудиторія	Веброзробники, орієнтовані на просту інтеграцію	Розробники нативних застосунків з високою складністю	Веброзробники, які створюють інтерактивні AR-сцени
Платформа	Веб (підтримка браузерів Chrome, Safari)	iOS, Android, AR-окуляри	Веб (підтримка сучасних браузерів)
Підтримка 3D-форматів	glTF, USDZ	glTF, OBJ, FBX, USDZ	glTF, OBJ
Можливості освітлення	HDRI-освітлення, PBR	Фізично коректне освітлення, HDRP	PBR, базова підтримка освітлення
Тип трекінгу	Виявлення поверхонь	SLAM, виявлення облич, маркерний трекінг	Маркерний та безмаркерний трекінг

Апаратні вимоги	Працює навіть на низькопродуктивних пристроях	Потребує потужного обладнання	Залежить від браузера і пристрою
Можливості інтеграції	Вбудовується через HTML-тег <model-viewer>	Потрібне програмування на C#	Вбудовується через бібліотеки JavaScript
Обмеження	Обмежена взаємодія, відсутність кастомізації	Великий розмір застосунків, складність освоєння	Залежність від браузера, обмежений функціонал

2.2 Методи моделювання за допомогою маркерів

Моделювання за допомогою маркерів є одним із найстаріших і найпоширеніших методів реалізації AR-технологій. Цей підхід базується на використанні спеціальних маркерів, які розпізнаються камерою AR-пристрою, аналізуються алгоритмами комп'ютерного зору та використовуються як опорні точки для інтеграції віртуальних об'єктів у реальне середовище. Завдяки своїй доступності, високій точності та відносно невисоким вимогам до апаратного забезпечення, метод маркерного моделювання широко застосовується у навчальних, промислових, медичних та розважальних AR-системах.

Метод моделювання за допомогою маркерів спирається на фундаментальні принципи комп'ютерного зору, геометрії та оптики. Камера пристрою захоплює зображення, на якому присутній маркер, після чого зображення обробляється спеціалізованими алгоритмами. У результаті обробки визначаються ключові характеристики маркера: його орієнтація, масштаб, координати у просторі та відносне положення щодо камери.

Однією з ключових особливостей цього методу є використання маркерів як опорних елементів для визначення просторових параметрів сцени. Маркер зазвичай має контрастний малюнок (чорно-білий або кольоровий) та унікальні геометричні характеристики (рис. 2.3), що

дозволяють його легко ідентифікувати серед інших елементів сцени.

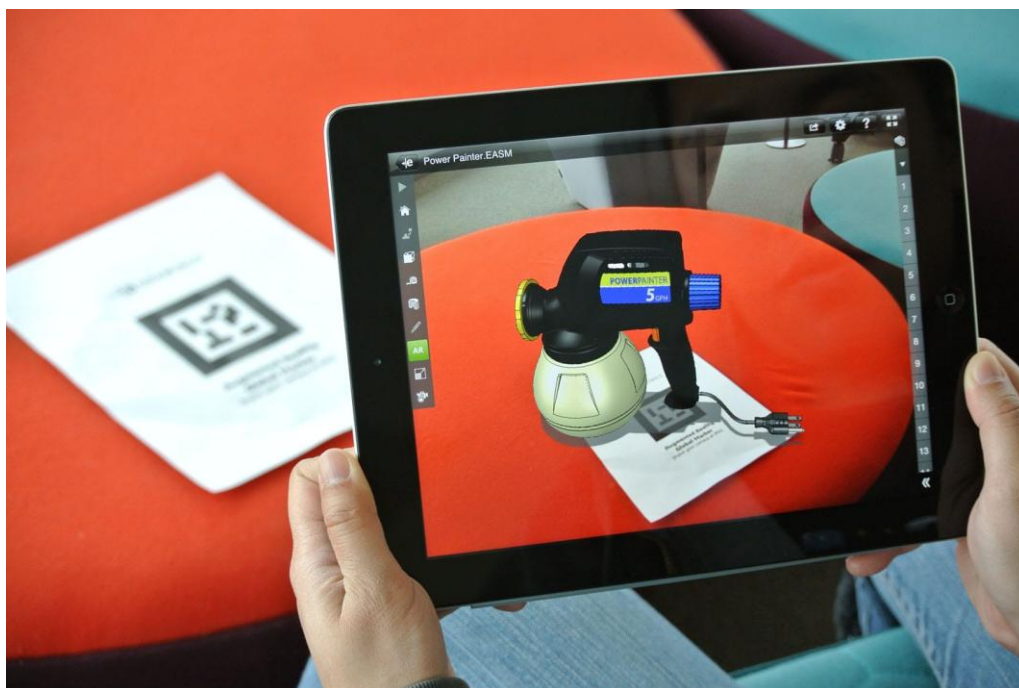


Рисунок 2.3 – Використання маркеру для позиціонування AR моделі

Обробка маркерів виконується у кілька етапів. Спочатку камера захоплює кадр, після чого відбувається попередня обробка зображення, яка включає фільтрацію шумів і підвищення контрастності. Наступним кроком є виділення маркера за допомогою алгоритмів комп'ютерного зору, таких як SIFT або ORB, які аналізують геометрію маркера, його межі та ключові точки. Після ідентифікації маркера виконується обчислення його просторових характеристик за допомогою математичних моделей, таких як перспектива проєкції або метод PnP (Perspective-n-Point).

Фінальним етапом є накладання віртуального об'єкта на маркер, що дозволяє забезпечити його стабільність і правильне позиціонування у просторі навіть при русі камери чи маркера.

У маркерному моделюванні використовуються різні типи маркерів, залежно від поставлених завдань та умов їх застосування. Класичними прикладами є чорно-білі маркери, QR-коди, геометричні фігури та

інтерактивні кольорові маркери. Чорно-білі маркери забезпечують високу точність розпізнавання завдяки чіткому контрасту між кольорами, тоді як QR-коди дозволяють зберігати додаткову інформацію про об'єкт чи сцену. Геометричні маркери, у свою чергу, є гнучкими у створенні й можуть бути адаптовані до специфічних умов проекту.

Інтерактивні кольорові маркери дозволяють реалізовувати більш складні сцени та підвищують стабільність відображення в умовах змішаного освітлення. Однак усі типи маркерів мають спільне обмеження: вони залежать від умов освітлення та якості камери пристрою.

Технічна сторона маркерного моделювання включає кілька ключових аспектів, які визначають його ефективність та стабільність. Один із них – це параметри камери, що використовуються для захоплення зображень. Роздільна здатність, чутливість до освітлення та частота кадрів безпосередньо впливають на якість розпізнавання маркера та стабільність роботи системи.

Також важливою складовою є алгоритми комп'ютерного зору. Використання алгоритмів, таких як ORB (Oriented FAST and Rotated BRIEF), дозволяє значно знизити обчислювальні витрати при розпізнаванні маркерів у реальному часі. З іншого боку, алгоритми на основі SIFT (Scale-Invariant Feature Transform) забезпечують високу точність навіть при частковому перекритті маркерів або змінах кута огляду[4].

Маркерний метод має низку значних переваг, серед яких висока точність, простота реалізації, мінімальні вимоги до апаратного забезпечення та сумісність із багатьма платформами. Проте є й певні обмеження, такі як залежність від якості камери, обмежений робочий діапазон і чутливість до умов освітлення.

Попри названі недоліки, маркерне моделювання залишається одним із найефективніших та доступних методів для створення AR-застосунків. Воно забезпечує стабільну роботу в широкому спектрі сценаріїв і є основою для багатьох сучасних AR-рішень.

У маркерному моделюванні важливим елементом є самі маркери, оскільки саме вони слугують основою для визначення положення, орієнтації та масштабу віртуальних об'єктів у просторі. Типи маркерів визначаються їх структурою, формою, способом кодування інформації та технологією, яка використовується для їх розпізнавання. Вибір типу маркера залежить від поставлених задач, доступного обладнання та специфічних вимог до продуктивності й точності системи.

Чорно-білі маркери є найпоширенішими завдяки своїй простоті та надійності. Вони складаються з контрастних геометричних фігур, найчастіше чорно-білих квадратів або прямокутників із унікальними візерунками (рис. 2.4). Ці маркери легко виявляються за допомогою алгоритмів комп'ютерного зору, оскільки контраст між чорними та білими ділянками дозволяє швидко ідентифікувати межі та орієнтацію маркера. Вони особливо ефективні при використанні в умовах із мінливим освітленням і на пристроях із низькою роздільною здатністю камери. До основних переваг цього типу можна віднести простоту виготовлення (їх можна надрукувати на звичайному папері) та мінімальні вимоги до обчислювальних ресурсів. Проте чорно-білі маркери мають обмеження у вигляді неможливості кодування значного обсягу інформації та чутливості до часткового перекриття маркера іншими об'єктами.

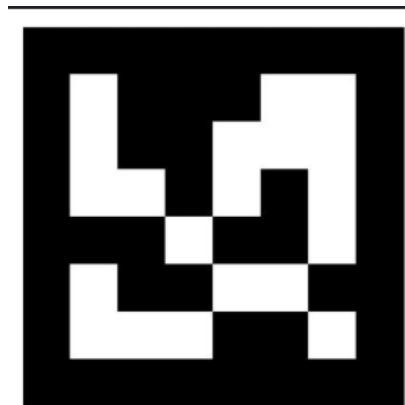


Рисунок 2.4 – Чорно-білий ARTag маркер

QR-коди також активно використовуються як маркери в системах доповненої реальності. Вони являють собою двовимірні коди, що складаються з чорно-білих квадратів (рис. 2.5), які можуть зберігати велику кількість інформації, включно з текстом, посиланнями, координатами та іншими даними. Завдяки вбудованим механізмам корекції помилок QR-коди можуть бути розпізнані навіть за умови незначного пошкодження або часткового перекриття. Це робить їх ідеальними для використання в застосунках, які вимагають передачі додаткових даних через маркер. Проте, порівняно з класичними чорно-білими маркерами, обчислювальні витрати на їх розпізнавання є вищими, а процес обробки займає більше часу.



Рисунок 2.5 – QR код з посиланням

Фідучіальні маркери є спеціалізованими маркерами з унікальними візерунками, які легко розпізнаються алгоритмами навіть у складних умовах. Вони часто використовуються в застосунках із високими вимогами до точності, таких як медичні тренажери або промислові системи контролю якості. Фідучіальні маркери забезпечують високу точність і стабільність навіть при мінливих умовах освітлення та значних нахилах камери. Однак вони вимагають більш складних алгоритмів для розпізнавання, що підвищує обчислювальні витрати.

Кожен із типів маркерів має свої переваги та обмеження, які роблять їх

ефективними в певних сценаріях. Чорно-білі маркери є універсальними для більшості стандартних задач, тоді як QR-коди забезпечують додаткові можливості завдяки передачі даних. Геометричні маркери підходять для застосунків, які вимагають швидкості та простоти обробки, тоді як кольорові маркери ефективні у складних сценах із динамічним фоном. Фідучіальні маркери забезпечують найвищу точність, але потребують додаткових ресурсів для обробки.

Таким чином, вибір типу маркера залежить від конкретних завдань, доступного апаратного забезпечення та вимог до точності та продуктивності системи. Вдало підібраний тип маркера забезпечує високу якість візуалізації, стабільність AR-сцен і ефективне використання ресурсів пристрою.

2.2.1 Алгоритми розпізнавання маркерів

Розпізнавання маркерів у системах доповненої реальності є критичним етапом, оскільки воно визначає точність позиціонування віртуальних об'єктів у реальному просторі. Алгоритми, що відповідають за цей процес, обробляють вхідне зображення, ідентифікують маркер, аналізують його особливості та обчислюють його просторові характеристики. Цей процес є багатоступеневим і включає кілька важливих етапів. Спочатку вхідне зображення, отримане з камери, піддається попередній обробці. Ця стадія включає фільтрацію шумів для зменшення артефактів, які можуть вплинути на точність розпізнавання, перетворення у відтінки сірого для спрощення подальшого аналізу та бінаризацію зображення за допомогою порогових значень, таких як метод Отсу (рис. 2.6). Ці операції дозволяють виділити контрастні ділянки, які можуть бути маркерами.

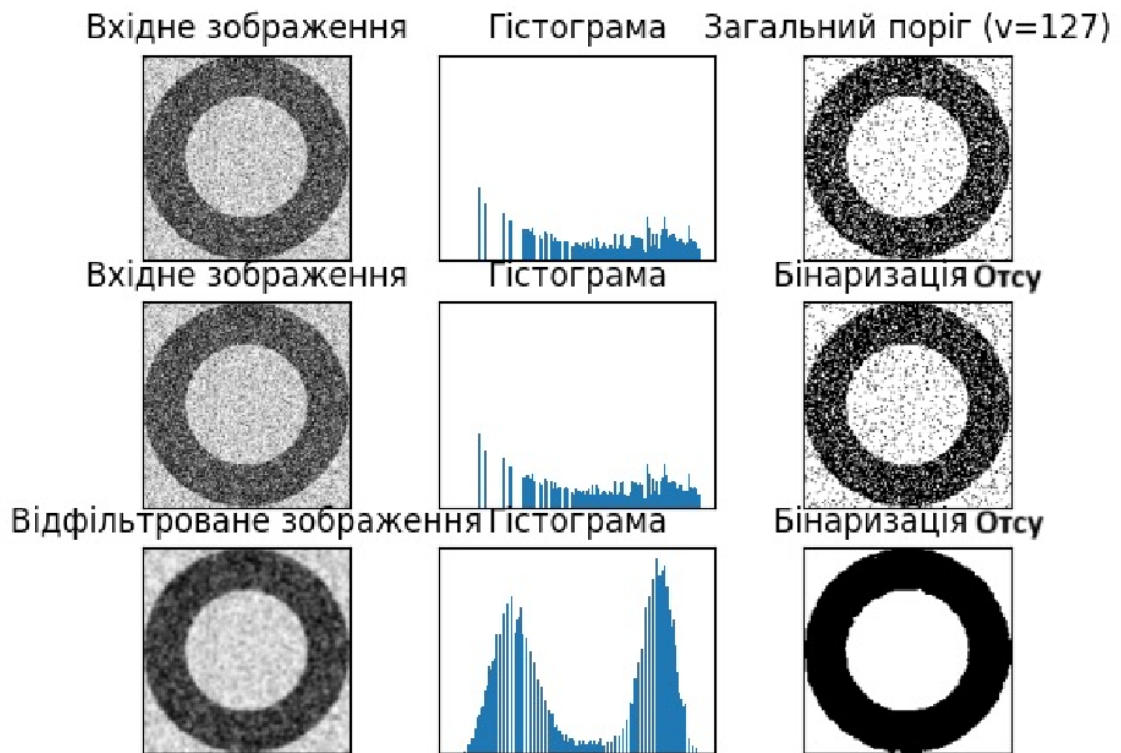


Рисунок 2.6 – Етапи обробки зображення методом Отсу

Після попередньої обробки алгоритми переходять до етапу виявлення контурів і ключових точок. Використання методів виявлення контурів дозволяє програмному забезпеченню виділити чіткі межі маркерів і проаналізувати їх геометрію. Додатково застосовуються алгоритми виявлення кутів, наприклад метод Харріса, які допомагають знайти точки перетину прямих ліній на маркері[5]. Ці ключові точки слугують орієнтирами для подальшої ідентифікації маркера та його орієнтації у просторі. На рис. 2.7 зображено візуалізацію метода Харріса.

Наступним етапом є ідентифікація маркера, яка відбувається шляхом порівняння знайденого візерунка з заздалегідь визначеною базою даних. Для цього застосовуються алгоритми порівняння шаблонів, які зіставляють отримане зображення з еталонним, а також методи порівняння характеристик, що аналізують унікальні ключові точки маркера. Це дозволяє системі точно ідентифікувати маркер, навіть якщо він частково перекритий або

спотворений через кут нахилу камери.

На етапі обчислення орієнтації та позиції маркера використовуються математичні моделі, такі як метод Perspective-n-Point, що мінімізує похибку між прогнозованими та реальними координатами точок на маркері.

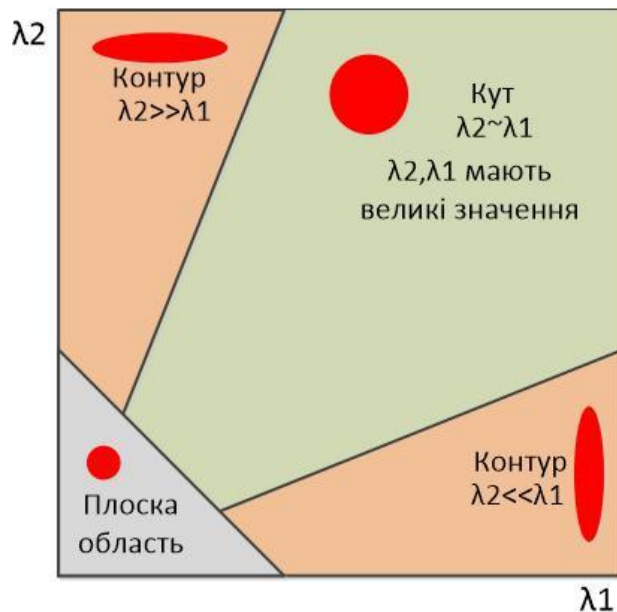


Рисунок 2.7 – Візуалізація алгоритму виявлення кутів Харріса

Це дозволяє точно визначати кут нахилу, положення та масштаб маркера у тривимірному просторі. Для додаткового згладжування результатів та усунення можливих стрибків у позиціонуванні використовуються фільтри, такі як фільтр Калмана[6]. Ці методи дозволяють забезпечити стабільність віртуальних об'єктів навіть у динамічних сценах, де камера або маркер перебувають у русі.

У сучасних системах доповненої реальності активно використовуються різні алгоритми для розпізнавання маркерів. Одним із найвідоміших є ARToolKit, який був одним із перших інструментів для розпізнавання маркерів і досі забезпечує високу точність завдяки використанню методів перспективної проєкції. Ще одним популярним інструментом є модуль ARUCO бібліотеки OpenCV, що підтримує ефективне виявлення маркерів у

реальному часі та має вбудовані механізми корекції помилок. AprilTag, у свою чергу, спеціалізується на роботі з фідуціальними маркерами і демонструє високу продуктивність навіть у випадках, коли маркер частково перекритий або має низьку роздільну здатність. Важливу роль у розвитку AR-інструментів відіграє і платформа Vuforia, яка використовує вдосконалені алгоритми для ідентифікації маркерів, включно з аналізом текстур і унікальних візерунків.

Ефективність алгоритмів розпізнавання маркерів залежить від кількох ключових факторів. Одним із них є освітлення, оскільки недостатнє або надмірне освітлення може значно вплинути на точність розпізнавання. Важливим фактором є роздільна здатність камери – чим вона вища, тим більш точними будуть результати розпізнавання. Також важливий кут огляду: великі кути нахилу камери можуть призводити до спотворень і втрати точності у визначенні координат. Крім того, якість самих маркерів, їх чіткість і відсутність пошкоджень відіграють важливу роль у надійності системи.

2.2.2 Методи обчислення позиції та орієнтації

Процес обчислення позиції та орієнтації маркера у просторі є одним із ключових етапів у маркерному моделюванні. Від точності цього процесу залежить правильне позиціонування віртуальних об'єктів у реальному середовищі та їхня стабільність під час взаємодії з користувачем. Сучасні алгоритми використовують математичні моделі, що базуються на принципах проєктивної геометрії, триангуляції та обробки зображень, для досягнення максимальної точності та швидкодії.

Процес обчислення включає три основні завдання:

- визначення просторових координат маркера;
- визначення його орієнтації відносно камери;
- стабілізація отриманих даних.

Перш за все система аналізує ключові точки маркера, що були виявлені на попередніх етапах розпізнавання. Точки використовуються для побудови математичної моделі, що дозволяє інтерпретувати їх у тривимірному просторі. Для цього застосовуються методи, що базуються на розв'язку задачі Perspective-n-Point (PnP)[7]. Задача полягає у визначенні просторових координат об'єкта за відомими двовимірними координатами точок на зображенні та тривимірними координатами відповідних точок на маркері.

Класичне рівняння для методу PnP можна виразити так:

$$p = K \cdot [R | t] \cdot P, \quad (2.1)$$

де p – двовимірні координати точки на зображенні, K – матриця внутрішніх параметрів камери, R – матриця обертання, що визначає орієнтацію маркера, t – вектор зміщення (translation vector), що визначає положення маркера відносно камери, P – тривимірні координати точки на маркері у його власній системі координат.

Використання цього рівняння дозволяє точно визначати положення та орієнтацію маркера у просторі. Важливо зазначити, що точність розв'язку задачі Perspective-n-Point залежить від кількості ключових точок, їх точності та якості зображення, що використовується для обробки.

Окрім PnP, у процесі обчислення можуть застосовуватися інші методи, такі як EPnP (Efficient PnP), що забезпечує оптимізацію обчислень і знижує навантаження на процесор, або UPnP (Uncalibrated PnP), який дозволяє виконувати розрахунки без попереднього калібрування камери. EPnP особливо ефективний у випадках, коли обробка повинна відбуватися в реальному часі, тоді як UPnP часто використовується у сценаріях, де точні параметри камери недоступні.

Після обчислення орієнтації та позиції маркера виконується фільтрація даних для усунення шумів і стабілізації результатів. Найпопулярнішим методом стабілізації є фільтр Калмана, який забезпечує згладжування

отриманих координат і усуває випадкові відхилення, що можуть виникати під час руху камери або маркера. Крім фільтра Калмана, можуть застосовуватися алгоритми на основі комплементарних фільтрів, які поєднують дані з різних джерел, таких як гіроскопи та акселерометри.

Слід зазначити, що обчислення позиції та орієнтації маркерів значною мірою залежать від умов експлуатації. Наприклад, якість освітлення, кут огляду камери, якість самої камери та характеристики маркера можуть суттєво вплинути на кінцевий результат. У випадках, коли кут огляду є надто великим або маркер частково перекритий іншими об'єктами, точність обчислень може знижуватися, а віртуальні об'єкти можуть почати «тремтіти» або відображатися у неправильному місці.

Використання сучасних математичних моделей, таких як PnP, EPnP та UPnP, у поєднанні з фільтрацією даних за допомогою фільтра Калмана, дозволяє досягти високої точності, стабільності та надійності роботи AR-систем. Точне визначення положення та орієнтації маркерів забезпечує коректне відображення віртуальних об'єктів, що є критично важливим для будь-якого AR-застосунку.

2.2.3 Переваги та обмеження маркерного моделювання

Маркерне моделювання є одним із найпоширеніших методів реалізації технологій доповненої реальності завдяки своїй доступності, точності та відносній простоті у впровадженні. Використання маркерів забезпечує стабільне позиціонування віртуальних об'єктів у просторі, що є критично важливим для багатьох AR-застосунків. Завдяки низьким обчислювальним вимогам і можливості працювати на пристроях із обмеженими ресурсами, маркерне моделювання залишається затребуваним рішенням у різних галузях.

Разом із тим, цей метод має свої обмеження. Він вимагає стабільних умов освітлення, чистих і непошкоджених маркерів, а також відсутності значного перекриття маркера іншими об'єктами. Крім того, масштабованість

маркерних систем часто є викликом, особливо в сценаріях із великою кількістю маркерів.

Нижче представлена таблиця 2.2, яка узагальнює ключові переваги та обмеження маркерного моделювання.

Таблиця 2.2 – Переваги та обмеження моделювання за допомогою моделювання

Параметр	Переваги	Обмеження
Точність	Висока точність визначення позиції та орієнтації.	Знижується при частковому перекритті маркера.
Продуктивність	Низькі обчислювальні вимоги.	Обмежена кількість маркерів у кадрі.
Освітлення	Стабільна робота при адекватному освітленні.	Чутливість до змін освітлення.
Доступність	Легке створення та друк маркерів.	Вимагає попереднього розміщення маркерів.
Стабільність	Висока стабільність при фіксованих умовах.	Тремтіння об'єктів при динамічних рухах.
Вартість	Низькі витрати на виробництво маркерів.	Додаткові витрати на масштабування системи.

2.3 Методи моделювання без використання маркерів

Методи моделювання без використання маркерів (markerless AR) є сучасним напрямом у технологіях доповненої реальності, що динамічно розвивається. Методи базуються на розпізнаванні об'єктів, поверхонь або особливостей середовища за допомогою комп'ютерного зору, алгоритмів машинного навчання та сенсорних даних, отриманих від камер, акселерометрів і гіроскопів. Ці методи дозволяють створювати AR-застосунки, які не потребують фізичних маркерів, що значно розширює можливості їх застосування.

Однією з ключових особливостей безмаркерного моделювання є використання природних особливостей середовища для розпізнавання та відстеження. Алгоритми аналізують текстури, кути, форми та інші унікальні характеристики об'єктів у кадрі, щоб створити карту середовища та визначити орієнтацію пристрою у просторі. Для цього часто використовуються алгоритми SLAM, які дозволяють одночасно будувати карту середовища та відстежувати положення пристрою у ньому (рис. 2.8).

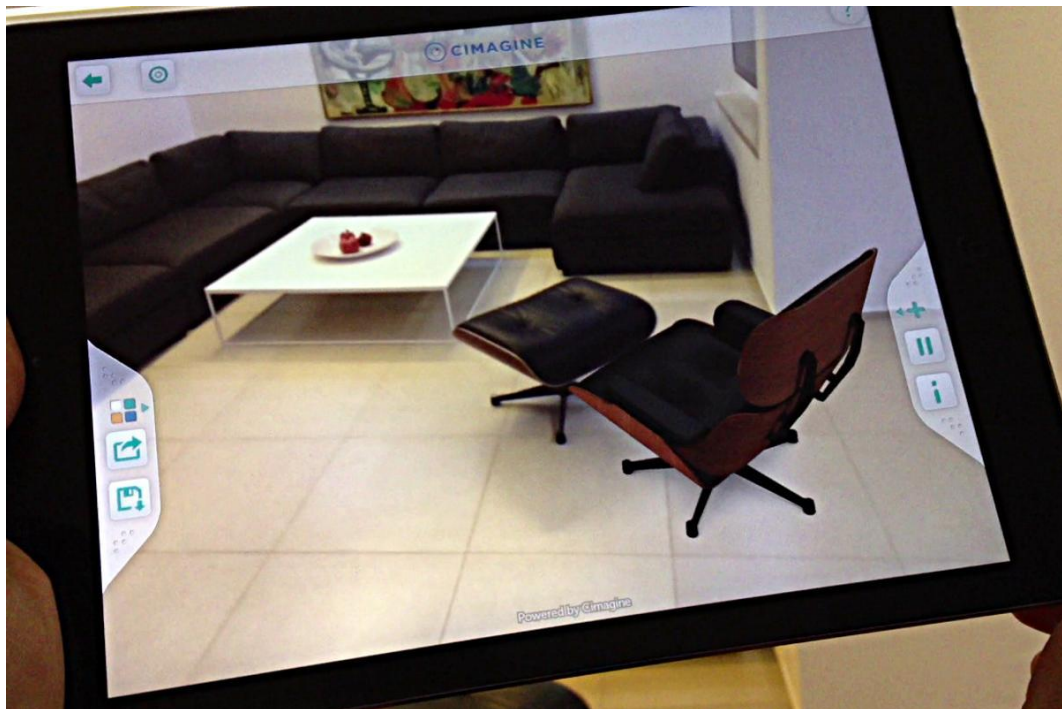


Рисунок 2.8 – Модель в доповненій реальності розміщена на карті середовища

SLAM є одним із найефективніших підходів у безмаркерному моделюванні. Цей метод використовує дані з камери та додаткових сенсорів для створення карти навколишнього середовища в реальному часі. Принцип роботи SLAM полягає у визначенні унікальних точок (feature points) на об'єктах та їх відстеженні під час руху пристрою. Ці точки допомагають алгоритму відстежувати зміни у просторі та оновлювати карту в реальному часі. Завдяки SLAM технології, AR-застосунки можуть працювати навіть у

складних середовищах із мінливими умовами освітлення та динамічними об'єктами.

Другим популярним методом у безмаркерному моделюванні є використання хмар точок (Point Clouds). Цей підхід базується на створенні тривимірної моделі простору з набору точок, отриманих із камер та сенсорів. Хмари точок дозволяють точно відображати геометрію середовища та забезпечують високу точність у позиціонуванні віртуальних об'єктів. Проте обробка хмар точок є доволі ресурсомісткою задачею, що може створювати виклики для пристроїв із обмеженими обчислювальними можливостями.

Алгоритми на основі глибокого навчання також активно застосовуються у безмаркерному моделюванні. Вони дозволяють системам навчатися розпізнавати об'єкти, поверхні та особливості середовища за допомогою попередньо навченої нейронної мережі. Нейронні мережі аналізують отримане зображення та визначають ключові точки, що дозволяє точно визначати положення та орієнтацію об'єктів.

Основною перевагою методів без використання маркерів є їхня гнучкість і незалежність від фізичних об'єктів у середовищі. Це дозволяє створювати інтерактивні AR-застосунки, які можуть працювати у будь-якому середовищі, без необхідності попереднього налаштування маркерів. Однак ці методи також мають свої обмеження. Вони вимагають значних обчислювальних ресурсів, а їх ефективність залежить від якості камери, умов освітлення та наявності достатньої кількості унікальних особливостей у кадрі.

Ефективність методів безмаркерного моделювання значною мірою залежить від таких факторів, як:

- точність алгоритмів комп'ютерного зору та машинного навчання;
- якість сенсорних даних, отриманих із камери, акселерометра та гіроскопа;
- швидкість обробки даних та можливості апаратного забезпечення;
- стабільність освітлення та мінімізація шумів на зображенні.

Методи моделювання без використання маркерів відкривають нові

горизонти у сфері доповненої реальності. Вони дозволяють створювати інтерактивні, динамічні та більш природні AR-середовища. Незважаючи на високі вимоги до обчислювальних ресурсів і залежність від якості вхідних даних, ці методи продовжують активно розвиватися, забезпечуючи все більш точне та стабільне позиціонування віртуальних об'єктів у реальному просторі.

2.4 Технології використання хмар точок і SLAM у моделюванні

Використання хмар точок і технологій SLAM (Simultaneous Localization and Mapping – одночасна локалізація та побудова карти) є одним із найбільш передових підходів у моделюванні приміщень за допомогою технологій доповненої реальності. Дані технології дозволяють досягти високої точності при визначенні положення об'єктів і побудові просторових карт реального світу у реальному часі. Вони забезпечують ефективне відстеження динамічних змін у середовищі та створюють основу для надійної інтеграції віртуальних елементів у фізичний простір (рис 2.9).

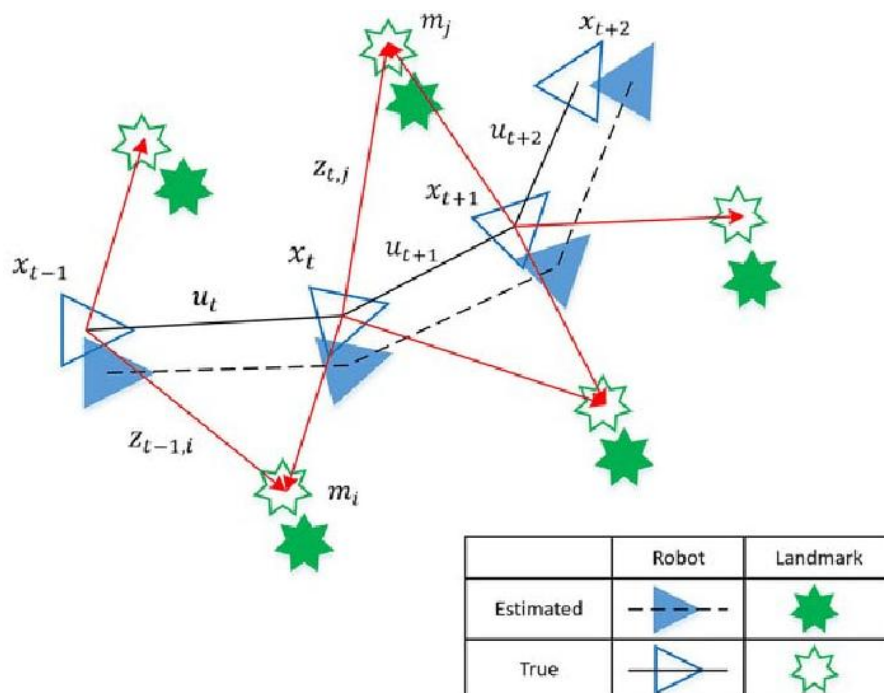


Рисунок 2.9 – Схематична діаграма SLAM

Хмари точок є набором точок у тривимірному просторі, що описують поверхню об'єктів і середовища. Кожна точка в хмарі має свої координати (x, y, z) і, у деяких випадках, додаткові властивості, такі як колір і яскравість. Для створення хмар точок використовуються різні технології, зокрема:

- лідарні сенсори (LiDAR): використовують лазерні промені для сканування простору й вимірювання відстані до об'єктів;
- стереокамери: аналізують різницю між двома зображеннями для обчислення глибини сцени;
- тоф-камери (Time-of-Flight): вимірюють час, необхідний світлу для відбиття від об'єкта та повернення до сенсора.

Хмари точок дозволяють створювати надзвичайно детальні тривимірні моделі середовища, що робить їх незамінними в застосунках, де важливе точне позиціонування та орієнтація. Проте, обробка хмар точок вимагає значних обчислювальних ресурсів, що може стати викликом для мобільних пристроїв та систем з обмеженими можливостями[10].

SLAM є технологією, що дозволяє пристрою одночасно будувати карту навколишнього середовища і визначати своє місцеположення у ньому. Цей підхід поєднує дані, отримані з камер, гіроскопів, акселерометрів та інших сенсорів, для побудови просторових карт у реальному часі. Основний принцип роботи SLAM можна розділити на кілька етапів:

1. Збір даних сенсорів. Камера та інші сенсори отримують інформацію про навколишнє середовище.
2. Виділення ключових точок. Алгоритм визначає унікальні точки на поверхнях об'єктів, які можна використовувати як орієнтири.
3. Локалізація. Визначається положення пристрою відносно ключових точок, що допомагає побудувати точну карту.
4. Оновлення карти. Карта середовища постійно оновлюється у міру того, як пристрій переміщується.

Технологія SLAM має кілька популярних реалізацій, серед яких:

- Visual SLAM (vSLAM) використовує камери для аналізу зображень і побудови карти;
- RGB-D SLAM поєднує дані RGB-зображень і глибинних сенсорів для точнішого моделювання;
- Lidar SLAM використовує дані, отримані з лідарів, для високоточних вимірювань відстаней.

Однією з ключових переваг SLAM є його здатність працювати у динамічних середовищах і адаптуватися до змін. Наприклад, якщо об'єкт або освітлення в кімнаті змінилися, алгоритм SLAM автоматично оновлює карту, забезпечуючи стабільну роботу системи. Однак, технологія також має свої обмеження. Вона залежить від якості даних сенсорів, продуктивності пристрою та ефективності алгоритму обробки.

Ефективність SLAM і хмар точок значною мірою залежить від якості сенсорів, продуктивності пристрою, стабільності освітлення та алгоритмів фільтрації даних.

Застосування хмар точок і SLAM у моделюванні приміщень відкриває нові можливості для інтеграції доповненої реальності у проекти, пов'язані з розумними будинками. Ці технології дозволяють створювати інтерактивні віртуальні середовища з високим рівнем деталізації, які можуть адаптуватися до змін у реальному світі.

Технології хмар точок і SLAM є одними з найефективніших рішень для моделювання приміщень у контексті доповненої реальності. Вони забезпечують високу точність, динамічну адаптацію та стабільність навіть у складних умовах середовища. Проте їх впровадження потребує потужного апаратного забезпечення та ефективних алгоритмів обробки даних, що може створювати певні виклики для розробників AR-систем.

2.5 Порівняння методів моделювання AR для розумних будинків

Різні методи AR-моделювання мають свої унікальні особливості, переваги та обмеження, які слід враховувати при виборі оптимального підходу для конкретного застосування. Нижче представлена порівняльна таблиця 2.3 методів маркерного, безмаркерного моделювання, а також технологій, що використовують хмари точок і SLAM.

Таблиця 2.3 – Порівняння методів моделювання в AR

Критерій	Маркерне моделювання	Безмаркерне моделювання	Хмари точок	SLAM
Просторова точність	До ± 1 см на коротких відстанях	Залежить від ключових точок	До $\pm 0,5$ см при високій щільності точок	До ± 1 см за стабільних умов
Час обробки кадру	<50 мс	100–200 мс	>200 мс	100–150 мс
Обчислювальні ресурси	Підходить для слабких пристроїв	Потрібен потужний процесор	Потрібен GPU	Ефективний CPU + GPU
Залежність від умов освітлення	Помірна чутливість до зміни освітлення	Висока чутливість до контрастності	Висока чутливість до відбиттів	Помірна чутливість
Стабільність у динаміці	Стабільно у статичних сценах	Може втрачати стабільність при швидкому русі	Стабільно при низькій швидкості	Висока стабільність у русі
Вартість впровадження	Прості маркери, дешеве обладнання	Залежить від сенсорів	Висока, лідар, сенсори високої точності	Залежить від сенсорів та камер

Таблиця 2.3 надає огляд основних характеристик методів AR-моделювання, дозволяючи порівняти їх за ключовими критеріями, такими як точність, час обробки, вимоги до обчислювальних ресурсів, чутливість до

освітлення, стабільність та вартість впровадження. Це дає змогу обрати оптимальний метод для конкретного завдання, враховуючи технічні обмеження, вимоги до результатів та бюджет.

З РЕАЛІЗАЦІЯ AR-ЗАСТОСУНКУ ДЛЯ МОДЕЛЮВАННЯ ПРИМІЩЕНЬ У РОЗУМНОМУ БУДИНКУ

У цьому розділі розглянуто ключові аспекти технічної реалізації AR-додатку, зокрема вибір технологій та інструментів, архітектурну структуру, алгоритмічний підхід та основні функціональні можливості. Описано важливі компоненти системи, їх взаємодію та логіку роботи, а також наведено фрагменти коду, що ілюструють ключові етапи розробки. Особливу увагу приділено тестуванню додатку, оцінці його продуктивності.

3.1 Вибір технологій та обґрунтування їх використання

Розробка AR-застосунку для моделювання приміщень у розумному будинку вимагає ретельного вибору технологій та інструментів, що забезпечують високу продуктивність, стабільність і гнучкість системи. Обрані технології повинні відповідати як технічним вимогам проєкту, так і стандартам веброзробки, а також забезпечувати можливість інтеграції сучасних AR-функцій.

Для розробки додатку було обрано бібліотеку React, яка є однією з найпоширеніших бібліотек для створення вебзастосунків. React забезпечує компонентний підхід до побудови інтерфейсу, що дозволяє легко масштабувати застосунок, а також підтримує ефективне оновлення DOM (Document Object Model) завдяки використанню віртуального DOM. Крім того, велика екосистема інструментів та бібліотек, доступних для React, значно спрощує розробку, тестування та підтримку застосунку[8].

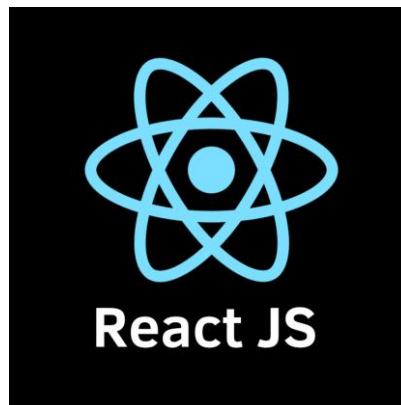


Рисунок 3.1 – Логотип Бібліотеки React

Ключовим елементом AR-функціональності є використання Google Model Viewer – вебкомпонента для інтеграції тривимірних моделей у веб-застосунки. Google Model Viewer підтримує формати GLTF і GLB, забезпечуючи високу якість відображення моделей із мінімальними вимогами до продуктивності пристрою. Додатковою перевагою є підтримка інтерактивності, включаючи можливість масштабування, обертання та переміщення моделей користувачем. Доповнена реальність у застосунку реалізована за допомогою Google Model Viewer, який використовує WebXR API для забезпечення інтеграції з апаратними можливостями пристроїв. Пряме звернення до WebXR API у коді відсутнє, оскільки функціональність роботи з AR-середовищем повністю реалізована всередині бібліотеки Google Model Viewer. Це дозволяє абстрагувати взаємодію з низькорівневими XR-сесіями, надаючи зручний інтерфейс для налаштування AR-сцен[11].

Мовою програмування додатку обрано JavaScript разом із сучасними можливостями стандарту ECMAScript. JavaScript забезпечує динамічну взаємодію між компонентами застосунку та підтримує асинхронне програмування, що є важливим для обробки подій у реальному часі. ECMAScript надає модульну структуру коду, що дозволяє організовувати його зрозуміло та підтримувано.

Середовище виконання Node.js використовується для управління залежностями та забезпечення серверної інфраструктури додатку. Node.js

дозволяє використовувати менеджер пакетів npm для інтеграції необхідних бібліотек, що забезпечує ефективне налаштування середовища розробки та підтримку залежностей.

Для стилізації інтерфейсу додатку застосовуються CSS та методи компонентної стилізації React. Це дозволяє створювати адаптивний дизайн, що забезпечує коректне відображення інтерфейсу на різних пристроях, а також надає можливість легко оновлювати та модифікувати стильові рішення у процесі розробки (рис. 3.2).

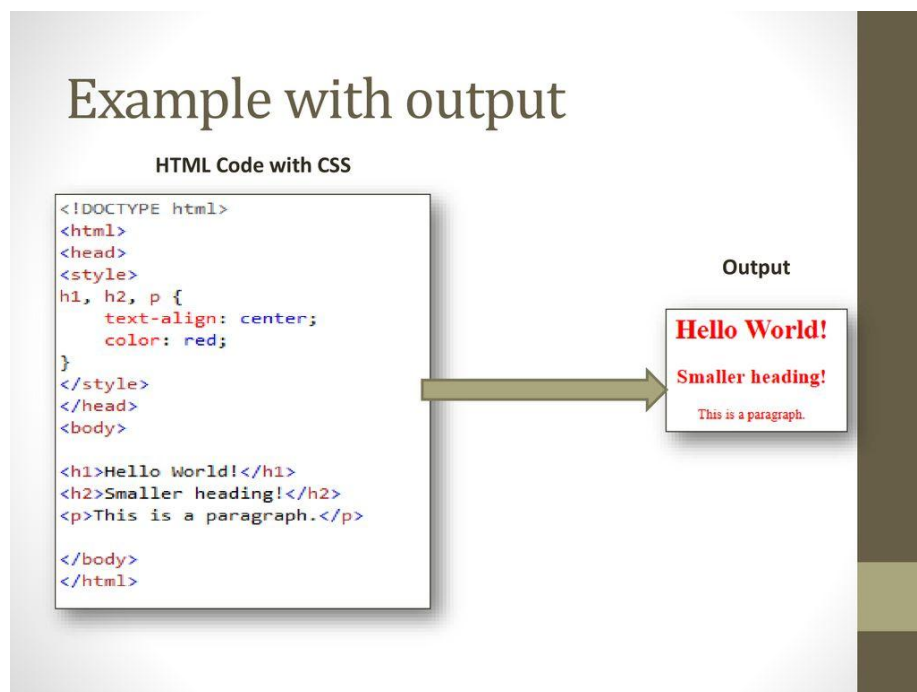


Рисунок 3.2 – Стилiзацiя за допомогою CSS

Таким чином, обраний технологiчний стек, що включає React, Google Model Viewer, JavaScript, Node.js та CSS, забезпечує стабiльнiсть, продуктивнiсть i масштабованiсть AR-застосунку для моделювання примiщень у розумному будинку. Кожна з технологiй вiдiграє важливу роль у досягненнi поставлених цiлей, забезпечуючи надiйну iнфраструктуру для iнтеграцiї AR-функцiональностi та пiдтримку динамiчних iнтерфейсiв[12].

3.2 Архітектура та структура застосунку

Реалізація AR-застосунку для моделювання приміщень у розумному будинку вимагає чітко визначеної архітектури, що забезпечує стабільність, продуктивність і легкість у підтримці системи. Архітектура застосунку ґрунтується на сучасних підходах до веброзробки, включаючи компонентний підхід, модульність і дотримання принципів розділення відповідальностей.

Архітектура застосунку базується на компонентній структурі, що дозволяє ізолювати окремі частини застосунку, забезпечуючи незалежність їхньої роботи та можливість подальшого масштабування. Ключовими компонентами є:

1. Головний компонент застосунку (App.js) – відповідає за ініціалізацію додатку, відображення основних елементів інтерфейсу та координацію роботи інших компонентів. У ньому відбувається підключення `<model-viewer>` для відображення 3D-моделей і управління їх параметрами.

2. Точка входу застосунку (index.js) – є місцем, де React-застосунок монтується в DOM. Цей файл відповідає за рендеринг основного компонента App.js у HTML-документі.

3. Структура HTML-документа (index.html) – забезпечує базовий каркас для React-застосунку та підключення необхідних ресурсів, включаючи CSS і скрипти. У цьому файлі створено основний контейнер для відображення компонентів React.

Комунікація між компонентами забезпечується через пропси (props) і стан (state). Дані передаються згори вниз (від батьківських компонентів до дочірніх), що дозволяє зберігати предбачуваність і прозорість у роботі компонентів.

Для роботи з 3D-моделями та AR-середовищем використовується компонент `<model-viewer>` у складі головного компонента. Цей елемент забезпечує відображення тривимірних моделей, їх рендеринг у реальному часі, а також інтерактивні можливості, такі як масштабування, обертання та

навігація.

Важливою частиною архітектури є система управління станами. У застосунку використовується локальний стан компонентів для зберігання даних, необхідних для відображення моделей і управління їх параметрами. Компонент App.js виступає як центральний вузол для управління станами, що дозволяє уникнути зайвої передачі даних між компонентами.

Інтерфейс користувача побудовано з урахуванням принципів адаптивного дизайну. CSS-стилі забезпечують коректне відображення компонентів на пристроях з різними роздільними здатностями екрану, а також підтримують кросбраузерну сумісність.

3.3 Алгоритм роботи застосунку

Ефективне функціонування AR-застосунку для моделювання приміщень у розумному будинку забезпечується чітко визначеним алгоритмом роботи, який охоплює етапи завантаження ресурсів, ініціалізації компонентів, обробки подій і візуалізації 3D-моделей у середовищі доповненої реальності. У цьому підрозділі буде представлено блок-схему алгоритму роботи застосунку та детально розглянуто ключові етапи його функціонування.

Алгоритм роботи застосунку можна умовно поділити на декілька основних етапів. Алгоритм роботи застосунку приведено у додатку Б.

1. Ініціалізація застосунку: завантаження ключових залежностей, рендеринг кореневого компонента App.js.
2. Ініціалізація компонента <model-viewer>: завантаження 3D-моделей та налаштування параметрів відображення.
3. Завантаження моделей: передача шляхів до 3D-моделей у компонент <model-viewer>.
4. Візуалізація сцени: побудова AR-сцени та інтерактивне відображення моделі на пристрої користувача.

5. Обробка подій користувача: реакція на дії користувача, такі як зміна моделі, обертання, масштабування.

6. Відображення результату: завершальна обробка даних та стабілізація візуалізації.

На етапі ініціалізації середовища React відбувається підключення головного компонента застосунку `App.js` до DOM. Після цього відбувається завантаження компонентів інтерфейсу користувача, включаючи елемент `<model-viewer>`, який є ядром AR-функціональності.

На етапі завантаження 3D-моделей відбувається передача шляхів до ресурсів (GLTF/GLB-файли) у властивості компонента `<model-viewer>`. Ці моделі зберігаються у папці `/models` і можуть бути обрані користувачем або автоматично завантажені під час ініціалізації застосунку.

Етап візуалізації AR-сцени включає обробку графічних даних і позиціонування об'єктів у просторі. Google Model Viewer відповідає за рендеринг моделей і інтеграцію їх у AR-середовище. Застосунок аналізує положення пристрою та камери, забезпечуючи точне відображення об'єктів у реальному просторі. Обробка подій користувача відбувається за допомогою слухачів подій, які відстежують такі дії, як дотики до екрана, масштабування, обертання та переміщення моделі. Дані події передаються компоненту `App.js`, де відбувається їх обробка та оновлення стану додатку.

На етапі оновлення стану компонентів зміни, викликані діями користувача, синхронізуються з внутрішнім станом додатку. Це забезпечує динамічне оновлення інтерфейсу та реактивне відображення змін на екрані.

Останній етап – відображення результатів, полягає у фінальній обробці даних і стабілізації AR-сцени для забезпечення плавного відображення та інтерактивності.

Кожен з етапів алгоритму взаємопов'язаний і забезпечує послідовне виконання операцій для досягнення стабільної та продуктивної роботи додатку.

3.4 Ключові функції застосунку

Файл `index.js` є точкою входу для React-застосунку. У ньому відбувається монтування головного компонента `App.js` у DOM.

У листингу 3.1 представлено код, який забезпечує ініціалізацію середовища React та підключення основного компонента `App`, який відповідає за бізнес-логіку застосунку.

Лістинг 3.1 - Ініціалізація застосунку та головного компонента

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

Компонент `<model-viewer>` є основою для відображення 3D-моделей. У файлі `App.js` даний елемент інтегровано як частину основного компонента.

- `src`: вказує шлях до 3D-моделі;
- `alt`: альтернативний текст для моделі;
- `auto-rotate`: автоматичне обертання моделі;
- `camera-controls`: дозволяє користувачу управляти камерою;
- `ar`: увімкнення режиму AR;
- `ar-modes`: визначає режими AR (WebXR, Scene Viewer, Quick Look).

У лістингу 3.2 наведено код відображення моделей за допомогою Google Model Viewer.

Лістинг 3.2 - Відображення 3D-моделей за допомогою Google Model Viewer

```
import React from 'react';
function App() {
  return (
    <div>
      <h1>AR Моделювання приміщень</h1>
```

```

    <model-viewer
      src="/models/lamp.glb"
      alt="3D модель лампи"
      auto-rotate
      camera-controls
      ar
      ar-modes="webxr scene-viewer quick-look"
      style={{ width: '100%', height: '500px' }}
    />
  </div>
);
}
export default App;

```

Важливою частиною застосунку є обробка подій користувача, таких як обертання, масштабування та вибір моделей. `onLoad`: Викликається після успішного завантаження моделі. `onARButtonClick`: Реагує на активацію AR-режиму користувачем. Ці обробники подій забезпечують зв'язок між користувацьким інтерфейсом та внутрішньою логікою застосунку.

У лістингу 3.3 наведені функції обробки користувацьких подій

Лістинг 3.3 - Обробка користувацьких подій

```

function handleModelLoad() {
  console.log('3D модель успішно завантажена');
}
function handleARButtonClick() {
  console.log('AR режим активовано');
}
<model-viewer
  src="/models/lamp.glb"
  ar
  camera-controls
  onLoad={handleModelLoad}
  onARButtonClick={handleARButtonClick}
/>

```

4 ТЕСТУВАННЯ ЗАСТОСУНКУ ДЛЯ МОДЕЛЮВАННЯ ПРИМІЩЕНЬ РОЗУМНОГО БУДИНКУ

Реалізований AR-застосунок для моделювання приміщень розумного будинку демонструє можливості інтеграції тривимірних моделей у реальний простір, а також забезпечує інтуїтивно зрозумілу взаємодію користувача з AR-сценами. У процесі тестування було підтверджено стабільність, високу продуктивність і точність відображення моделей у реальному середовищі.

Інтерфейс застосунку створено з урахуванням простоти використання та інтуїтивного управління. На головному екрані розміщено елементи керування такі як: кнопки, слайдери, списки; що дозволяють користувачу обирати 3D-моделі, налаштовувати їх розташування та взаємодіяти з ними у реальному просторі.

На рис 4.1 зображена початкова сторінка застосунку, яку користувач бачить кожний раз при запуску системи. Користувач має можливість натиснути на кнопку, щоб продовжити роботу з застосунком.

Welcome to AR App

Tap to continue

Continue

Рисунок 4.1 – Початкова сторінка застосунку

При натисканні на кнопку продовжити, користувач потрапляє на сторінку зі списком моделей до огляду (рис. 4.2). Користувач може вибрати одну з доступних моделей. Натискання на картку перенаправляє користувача на сторінку з попереднім переглядом моделі.

Choose a model to preview

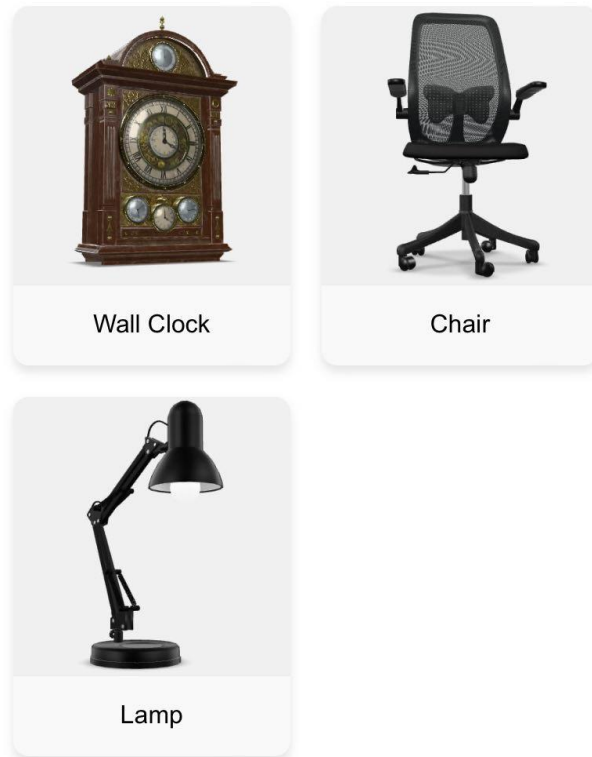


Рисунок 4.2 – Інтерфейс сторінки зі списком моделей до огляду

На рис. 4.3 зображена сторінка з попереднім переглядом моделі та її коротким описом. При натисканні на кнопку повернутись, можна потрапити назад на сторінку з вибором моделей.

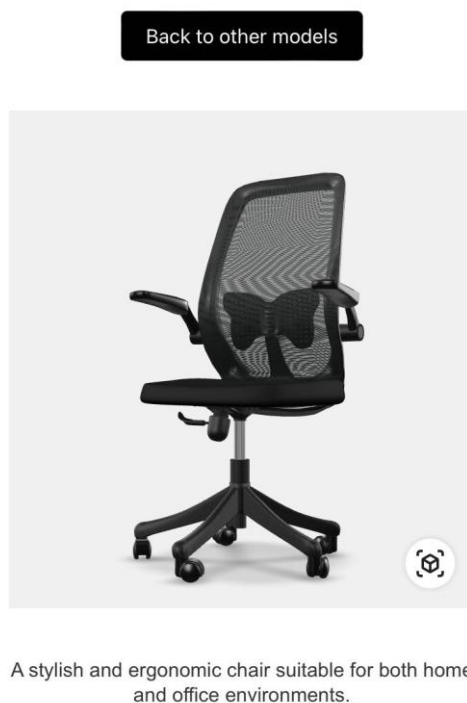


Рисунок 4.3 – Сторінка попереднього перегляду моделі

На рис. 4.4 зображена модель лампи після того, як користувач натискає на кнопку перегляду об'єкта у доповненій реальності.



Рисунок 4.4 – Відображення 3D-моделі лампи у доповненій реальності

Додавання AR-об'єктів у простір відбувається шляхом сканування поверхонь за допомогою камери пристрою. Користувач обирає необхідний об'єкт зі списку доступних моделей і розміщує його у реальному просторі, де він інтегрується у фізичне середовище з дотриманням пропорцій і орієнтації.

На рис. 4.5 зображена модель настінного годинника відображеного на стіні в реальному приміщенні.



Рисунок 4.5 – Відображення годинника у доповненій реальності

Користувач має можливість взаємодіяти з AR-сценою за допомогою сенсорного введення. Можна змінювати розмір, положення та орієнтацію моделей, переміщуючи їх у межах обраної сцени. Це забезпечує високий рівень гнучкості у процесі моделювання та дозволяє користувачу швидко адаптувати сцену до своїх потреб.

Результати тестування показали стабільну роботу AR-застосунку навіть при високому навантаженні, а затримки у відображенні моделей були мінімальними. Інтерактивні функції працювали коректно, а користувацький інтерфейс залишався інтуїтивно зрозумілим.

Реалізований AR-додаток продемонстрував високий рівень ефективності, стабільності та функціональності. Він відповідає заявленим вимогам та має значний потенціал для подальшого вдосконалення, що дозволить інтегрувати його у масштабні системи управління розумним будинком.

ВИСНОВКИ

Кваліфікаційна робота присвячена дослідженню методів моделювання приміщень розумного будинку з використанням технологій доповненої реальності та практичній реалізації AR-застосунку для візуалізації інтерактивних 3D-моделей у реальному середовищі. Використання AR у контексті систем розумного будинку відкриває широкі можливості для підвищення ефективності, зручності та інноваційності процесів проєктування, управління та експлуатації житлових приміщень.

У роботі було проведено аналіз концепції розумного будинку, визначено ключові компоненти таких систем, їхні переваги та недоліки. Розглянуто сучасні підходи до впровадження технологій доповненої реальності та їх застосування у сфері автоматизації житлових приміщень. Окрему увагу приділено аналізу відомих інструментів для створення AR-застосунків, зокрема Google Model Viewer, Unity та WebXR API з AR.js, проведено їх порівняльний аналіз з метою визначення найбільш раціонального для розробки та специфіку їхнього застосування у контексті моделювання приміщень розумного будинку.

Важливим етапом у роботі є аналіз методів моделювання, зокрема, за допомогою маркерів та немаркерних способів моделювання приміщень, а також використання технологій, що базуються на алгоритмах SLAM та хмарах точок. Було проведено детальний аналіз алгоритмів, принципів їхньої роботи та особливостей застосування у реальних умовах.

На основі проведеного аналізу визначено, що використання Google Model Viewer є найбільш оптимальним підходом для створення AR-застосунків завдяки його доступності, простоті інтеграції та підтримці сучасних форматів 3D-моделей. Використання алгоритмів SLAM та хмар точок забезпечує високу точність моделювання і ефективність у реальних умовах.

У практичній частині роботи було розроблено AR-застосунок для моделювання приміщень у концепції розумного будинку. Вибір технологій проєктування було обґрунтовано з урахуванням продуктивності, стабільності, кросплатформенності та доступності інструментів. Програмна реалізація AR-застосунку виконана з використанням мови програмування JavaScript. Основу архітектури застосунку склали бібліотека React, модуль Google Model Viewer та платформа для розробки застосунків Node.js. Було розроблено та описано алгоритм роботи застосунку, створено блок-схему, що демонструє послідовність основних етапів його функціонування. AR-застосунок дозволяє інтерактивно взаємодіяти з 3D-моделями у середовищі доповненої реальності, забезпечуючи можливість вибору моделей, їх масштабування, обертання та інтеграцію у фізичний простір.

У процесі тестування застосунку було підтверджено його стабільність, коректність відображення AR-сцен та відповідність технічним вимогам. Результати тестування показали ефективність обраних методів та технологій, зручність у розробці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Haidon C., Using Augmented Reality and Ontologies to Co-design Assistive Technologies in Smart Homes / C. Haidon, H. K. Ngankam, S. Giroux, H. Pigot. // 25th International Conference on Intelligent User Interfaces Companion, London, Great Britain – 2020. – P. 1-6 – DOI: 10.1145/3379336
2. Воронін Р.А. Моделювання інтер'єра розумного будинку з використанням інструментальних засобів доповненої реальності / Р.А. Воронін // Радіоелектроніка та молодь у XXI столітті: матеріали 28-го Міжнар. молодіж. форуму. 16-18 квітня 2024р. – Харків: ХНУРЕ. 2024 – Т.5. – С. 7-8.
3. Nagao K., Building-Scale Virtual Reality: Another Way to Extend Real World / K. Nagao, M. Yang, X. Cao, Y. Miyakawa. // 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) – 2019 – P. 205–211 – DOI: 10.1109/MIPR.2019.00044
4. Marques B., Configuration and Use of Pervasive Augmented Reality Interfaces in a Smart Home Context: A Prototype / B. Marques, P. Dias, J. Alves, E. Fonseca, B. Santos. // Advances in Intelligent Systems and Computing? Springer, Cham – Vol 1026. – 2019. P.96-102, DOI: 10.1007/978-3-030-27928-8_16
5. Ma Y., Interactive Home Product Design Method Using Augmented Reality Technology / Y. Ma, G. Chen, H. Gao. // 2023 4th International Conference on Machine Learning and Computer Application, Hong Kong, China – 2023. – P. 115-119, DOI: 10.1145/3650215.3650305
6. Deshmukh P., 2D to 3D Floor plan Modeling using Image Processing and Augmented Reality / P. Deshmukh, S. Kulkarni, D. Samuel, J. Mishra, L. Sankpal, C. Kulkarni, P. Kulkarni. // 2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies

(REEDCON), Boston, USA – 2023 – P. 682–687, DOI: 10.1109/REEDCON57544.2023.10151165

7. WebXR Device API [Электронный ресурс]. – Режим доступа <https://www.w3.org/TR/webxr/> (дата звернения: 01.06.2024).

8. React Official Documentation [Электронный ресурс]. – Режим доступа: <https://reactjs.org/> (дата звернения: 15.06.2024).

9. Unity Official Documentation [Электронный ресурс]. – Режим доступа: <https://docs.unity.com/> (дата звернения: 10.09.2024).

10. Google Model Viewer Docs for Developers [Электронный ресурс]. – Режим доступа: <https://modelviewer.dev/> (дата звернения: 21.10.2024).

11. MDN Web Docs [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/> (дата звернения: 10.11.2024).

12. Three.js Documentation [Электронный ресурс]. – Режим доступа: <https://threejs.org/docs/> (дата звернения: 18.11.2024)