

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Факультет інформатики
Кафедра мультимедійних систем

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: «**МАСШТАБУВАННЯ ТА РОЗМІЩЕННЯ ВІРТУАЛЬНИХ
3D-ОБ'ЄКТІВ У РЕАЛЬНОМУ ПРОСТОРИ ЗА ДОПОМОГОЮ AR**»

Керівник кваліфікаційної роботи

доцент Афонін А.О.

(підпис)

“ ____ ” _____ 2025 р.

Виконала студентка 4-го курсу
Освітньої програми «Комп'ютерні
науки», 122

Петренко Єлизавета Григорівна

“ ____ ” _____ 2025 р.

Київ – 2025

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Завідувач кафедри мультимедійних систем, проф., доцент, кандидат наук
Жежерун О. П.

_____ (підпис)

„___” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

Студентки 4-го курсу, факультету інформатики Петренко Єлизавети Григорівни

Тема: Масштабування та розміщення віртуальних 3D об'єктів в реальному просторі за допомогою AR

Зміст дипломної роботи:

Вступ

Розділ 1. Аналіз використання доповненої реальності у різних сферах сучасності

Розділ 2. Огляд засобів створення та імплементації технології AR

Розділ 3. Реалізація прототипу онлайн-каталогу з AR-переглядом меблів

Висновки

Список літератури

Додатки

Дата видачі „___” _____ 2025 р.

Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Календарний план виконання роботи:

№ п/п	Назва етапу дипломної роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу	14.10.2024	
2.	Складання плану кваліфікаційної роботи та узгодження з науковим керівником	30.10.2024	
3.	Ознайомлення з джерелами за темою роботи	05.11.2024	
4.	Огляд технічної літератури за темою роботи	14.12.2024	
5.	Пошук середовищ та платформ для розробки доповненої реальності	25.01.2025	
6.	Дослідження інструментів для інтеграції доповненої реальності на вебсторінку	20.03.2025	
7.	Розробка та тестування практичної частини	07.04.2025	
8.	Створення слайдів та написання доповіді	22.05.2025	
9.	Попередній захист	26.05.2025	
10.	Корегування роботи за результатами попереднього захисту	28.05.2025	
11.	Остаточне оформлення пояснювальної роботи та слайдів	31.05.2025	
12.	Захист кваліфікаційної роботи	06.06.2025	

Студентка Петренко Є.Г.

Керівник Афонін А.О.

“ _____ ”

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. Аналіз використання доповненої реальності у різних сферах сучасності.....	7
1.1 Сучасний стан проблеми та обґрунтування теми	7
1.2 Переваги використання технології AR	8
1.3 Обмеження масштабування та розміщення віртуальних 3D-об’єктів у реальному просторі	9
1.4 Розгляд прикладів застосування технології доповненої реальності у різних галузях	10
РОЗДІЛ 2. Огляд засобів створення та імплементації технології AR	16
2.1 Джерела 3D-моделей для доповненої реальності	16
2.1.1 Створення власних моделей.....	17
2.1.2 Готові бібліотеки моделей.....	20
2.1.3 Платформи з вбудованими моделями	221
2.2 Порівняння платформ та середовищ розробки AR.....	22
2.2.1 ARKit	22
2.2.2 ARCore.....	23
2.2.3 Unity з AR Foundation	24
2.2.4 Unreal Engine з AR Plugin	25
2.2.5 WebAR платформи.....	26
2.3 Технології взаємодії користувача з AR-середовищем	29
2.3.1 Основні технології трекінгу та позиціонування в AR.....	29
2.3.2 Інструменти та фреймворки для інтерактивності.....	31
РОЗДІЛ 3. Реалізація прототипу онлайн-каталогу з AR-переглядом меблів	34
3.1 Розробка проекту доповненої реальності	34
3.2 Інтеграція доповненої реальності на вебсторінку	37
3.3 Розширення функціональних можливостей	38
Висновки	41
Список літератури.....	42
Додаток А	44

ВСТУП

З розвитком технологій у сучасному світі, доповнена реальність все більше застосовується як допоміжний інструмент у різних сферах діяльності. Вона розв'язує низку різних задач, зокрема допомагає візуалізувати віртуальні 3D-об'єкти у режимі реального часу. Завдяки технології AR користувач має змогу отримувати миттєвий зворотній зв'язок через візуальну проекцію у просторі шляхом голосових команд, жестів або ж масштабування, переміщення, зміни розмірів або параметрів об'єкту. Важливим є також забезпечення інтуїтивної інтерактивності та контекстної адаптації аби забезпечити комфортну взаємодію користувача з віртуальним контентом.

Водночас, ефективне впровадження доповненої реальності потребує ретельного дослідження способів їх масштабування, розміщення та технічної бази, що забезпечує реалістичне відображення та інтеграцію віртуального контенту з реальним середовищем.

Метою даної роботи є розробка проєкту доповненої реальності, що містить певний набір 3D-моделей, а також імплементація AR-моделі на вебсторінку з каталогом меблів з можливістю передивлятися, розміщувати та масштабувати моделі меблів за допомогою технології доповненої реальності.

Об'єкт дослідження є процес розміщення та масштабування 3D-моделі у реальному просторі за допомогою технології AR.

У роботі будуть використовуватися чимало методів дослідження. Аналіз та узагальнення джерел буде одним з основних. Цей метод допоможе вибрати найбільш ефективні технології і платформи для реалізації. Також, у процесі розробки проєкту застосовується метод моделювання та тестування для покращення взаємодії користувача з практичним застосунком. Для підвищення якості та оптимізації процесу створення AR-контенту буде використано також порівняльний аналіз.

Дослідження складається з трьох розділів.

У першому розділі розглядається загальна проблематика використання доповненої реальності, були визначені сильні сторони та слабкі аспекти у роботі з нею. Також, у цьому розділі були розглянуті та порівняні приклади використання технології AR у застосунках популярних сервісів та онлайн-платформ для шопінгу.

У другому розділі було детально проаналізовано сучасні підходи до використання технологій доповненої реальності для інтеграції 3D-об'єктів у реальний простір. Також, було розглянуто та оцінено у порівнянні різні платформи для додавання, публікації та перегляду AR-моделей у застосунках.

Наприкінці, у третьому розділі, за допомогою вже визначеного аналізу інструментів та необхідних сервісів, було розроблено вебзастосунок з імплементацією 3D-об'єктів завдяки AR-технологіям. Також, було реалізовано певне розширення функціональних можливостей роботи проєкту щодо розміщення та масштабування правильної моделі у просторі.

Отже, доповнена реальність стає ключовою технологією сучасності, що трансформує способи взаємодії людей із навколишнім світом. У той же час, розвиток точних методів масштабування та розміщення 3D-об'єктів є фундаментальним для якісної інтеграції AR у різні сфери життя. Поглиблене дослідження цих методів розкриє комплексну картину про використання доповненої реальності, а також про її вплив у сьогоденні.

РОЗДІЛ 1. Аналіз використання доповненої реальності у різних сферах сучасності

1.1 Сучасний стан проблеми та обґрунтування теми

На сьогоднішній день доповнена реальність все більше й більше набуває актуальності завдяки її можливостям представлення віртуальних 3D-об'єктів у реальному просторі. Спираючись на статистику тенденцій, станом на 2024й рік вартість світового ринку AR сягнула 83,65 мільярдів доларів США. Окрім цього, очікується зростання з 2025й по 20230й роки на 37,9% у середньому за кожен рік (рисунок 1.1.1). Такі оцінки та планована динаміка явно свідчать про перспективність та значний потенціал використання доповненої реальності у різних галузях та сферах діяльності. Очевидно, що прогрес у комп'ютерній графіці, інтеграції зі штучним інтелектом та співпраці з іншим програмним забезпеченням грають визначну роль у розвитку та просуванні технології AR у найближчі роки.

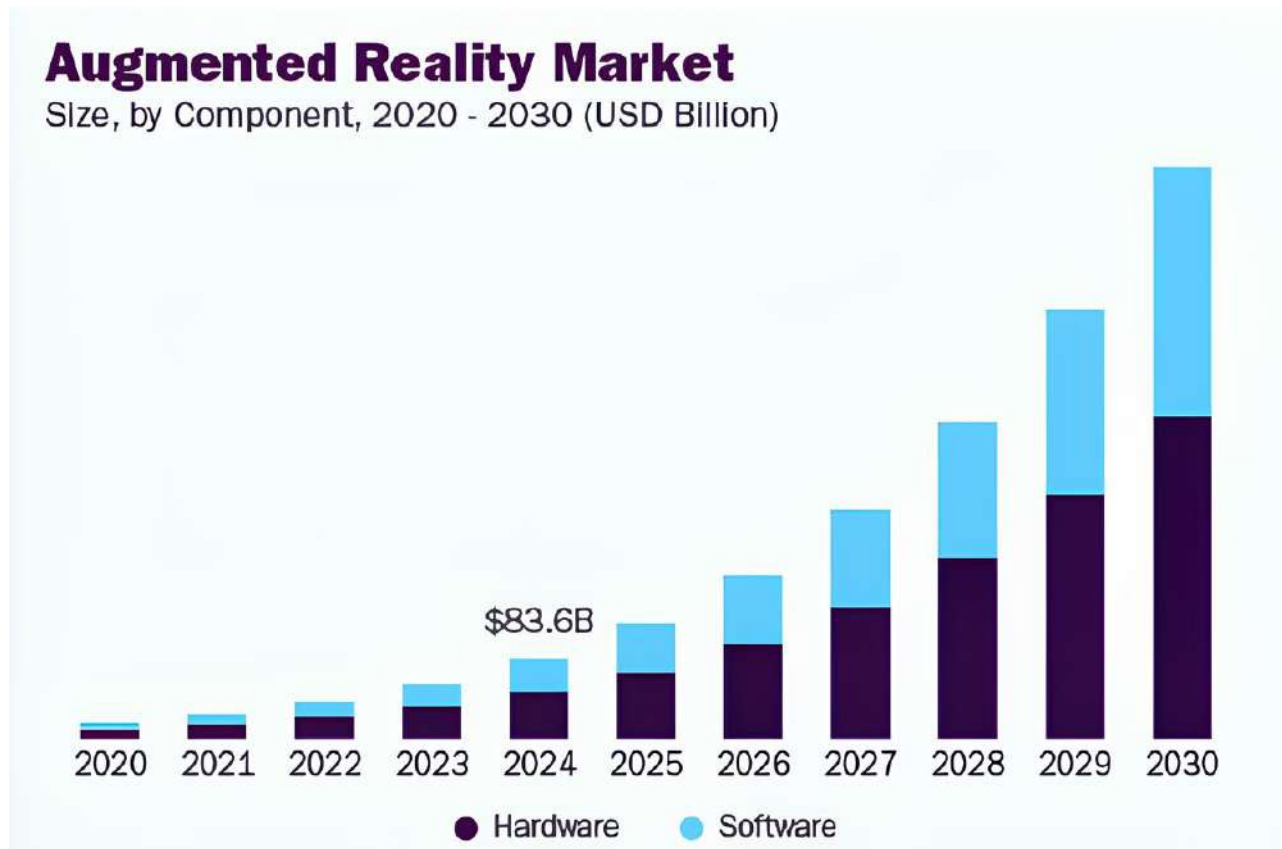


Рисунок 1.1.1 - Статистика оцінки ринку доповненої реальності

Безупинно зростає кількість компаній, які залучають до роботи технології AR. Це обумовлено удосконаленням залученості клієнта, покращенням обслуговування та користувальницького досвіду.

Однак, не зважаючи на зростання ринку, користувачі все ж можуть зіткнутися з проблемою точності масштабування та розміщення 3D-об'єктів у просторі. Наприклад, в залежності від умов тестування та виду трекінгу точність позиціонування може варіюватися від сантиметрів до кількох метрів. Такі нюанси можуть створювати певні труднощі щодо точності застосування доповненої реальності, що може створювати серйозні ризики.

Безперечно, розробка та впровадження інновації та нових технологій, таких як SLAM, LiDAR, SceneAR та інших, покращують якість масштабування та розміщення об'єктів. Наприклад, використання сучасних алгоритмів трекінгу забезпечують достатньо надійну роботу AR у контексті коректного позиціонування з похибкою до 3 мм.

Отже, постійне удосконалення технології доповненої реальності та прогнози щодо її застосування свідчать про великий потенціал цієї технології у найближчі роки. Без сумніву можна стверджувати, що технологія AR відіграє важливу роль у спрощенні роботи компаній, впливає на зростання продажів та залученість клієнтів.

1.2 Переваги використання технології AR

AR-технології являються одними з інноваційних технологічних досягнень сьогодення. Поєднуючи реальність з цифровою взаємодією вони дозволяють відтворювати точність розміщення та інтеграцію у реальний простір. Сучасні системи базуються на методі SLAM, що означає одночасну локалізацію та картографію. За допомогою цього методу з'являється можливість визначити своє місцезнаходження у режимі реального часу при моделюванні мапи навколишнього середовища. За рахунок цієї методології, а також, обробки сенсорних даних пристроєм забезпечується точність визначення місцезнаходження та орієнтування у просторі. Завдяки цьому, 3D-об'єкти у

доповненій реальності мають можливість досить точно відповідати розмірам та положенням у межах виміру.

Потрібно також відзначити, що використання AR-технології у проєктах значно знижує витратність на створення фізичного продукту або моделі. Завдяки можливості розміщення моделі прямо на робочому майданчику, виробники продукту можуть оглянути та внести корективи у функціональність або у певні характеристики ще до початку виготовлення товару. Таке динамічне вдосконалення виробів значно пришвидшує час розробки та роботи над проєктами.

Водночас, технології AR суттєво покращують залученість клієнтів та стимулюють використання продукту завдяки безпосередній взаємодії користувача з інтерактивним та персоналізованим контентом. Сприяння вдосконалення користувальницького досвіду та підвищення рівню лояльності аудиторії відбувається якраз через взаємодію з віртуальними об'єктами у реальному світі.

На додаток, використання AR-технологій підвищують якість навчання за допомогою інтерактивності та візуалізації складних процесів у роботі зі складним обладнанням, технічними деталями, медичними процедурами та іншим.

Окрім того, такі технології мають підтримку більшості платформ та девайсів, легко інтегруються у вебзастосунки та мобільні додатки, що робить AR більш доступним інструментом без додаткових інвестицій та вкладень.

Отже, переваги доповненої реальності охоплюють різні сфери – від маркетингу до фінансів. Ефективне використання можливостей новітньої системи сприяє поліпшенню інтерактивного досвіду та взаємодії з продуктом.

1.3 Обмеження масштабування та розміщення віртуальних 3D-об'єктів у реальному просторі

Не зважаючи на чималу кількість переваг технології AR, існують також і слабкі сторони цієї системи. Масштабування та розміщення 3D-об'єктів у реальному вимірі може бути деформовано за рахунок певної кількості факторів,

що, на жаль, не гарантує точність функціонування. Такі аспекти тісно пов'язані зі швидкістю переміщення моделі, наявністю значних коливань рельєфу та рівнів у просторі. Також, дуже часто точне відображення ускладнюється через низьку контрастність, погане освітлення або відсутність чітких орієнтирів у межах навколишнього середовища, що ускладнює відслідковування координат відображеного об'єкту.

До того ж, розміщення віртуальних об'єктів може бути обмежено через певні технічні характеристики старих моделей обладнання відтворення AR - технології, а саме камери та сенсори. Зокрема слабка обробка зображень та недостатня якість камер негативно впливають на точність розміщення та масштабування 3D-об'єкту.

Отож, хоча технологія AR має численні переваги у застосуванні та впровадженні у проєкти, вона також має певні недоліки, що впливають на низьку ефективність і точність відображення.

1.4 Розгляд прикладів застосування технології доповненої реальності у різних галузях

Розміщення та масштабування об'єктів у просторі знайшли широке застосування у чималій кількості галузей, що підтверджує їх цінність та ефективність у використанні. Таким чином, багато брендів запровадили цю технологію до себе в експлуатацію задля комерційного просування та клієнтської лояльності.

Так, наприклад, за останні роки популярним напрямленням застосування AR-технологій стали інтерактивні віртуальні примірочні у магазинах одягу, взуття та аксесуарів. Ця технологія виявилась особливо доцільною для того, аби побачити як на потенціальному покупці будуть виглядати певні елементи одягу або прикраси. Звичайна камера смартфона дає можливість зробити покупку менш втомливою та прискорити процес примірки.

Для прикладу можна привести функцію "Nike Fit" бренду Nike, що активно застосовує технологію доповненої реальності у своєму мобільному застосунку.

Якщо користувач хоче зробити заміри стопи, йому потрібно перейти на сторінку продукту та натиснути на посилання «виміряти стопу» поруч з меню вибору розміру. Далі застосунок запропонує користувачу встати біля стіни та направити камеру девайсу на стопи. Наступним кроком відбувається розпізнавання навколишнього простору та сканування стопи. У межах хвилини застосунок запропонує користувачу його ідеальний розмір взуття.

Згодом Nike розробили ще одну функцію під назвою “Nike By You”, за допомогою якої з’являється можливість інтерактивно підбирати дизайн, колір та матеріал взуття (рисунок 1.4.1). Такий підхід до примірки також значно посприяв збереженню бренду мільйонів доларів через значні скорочення онлайн-повернень на 28%.

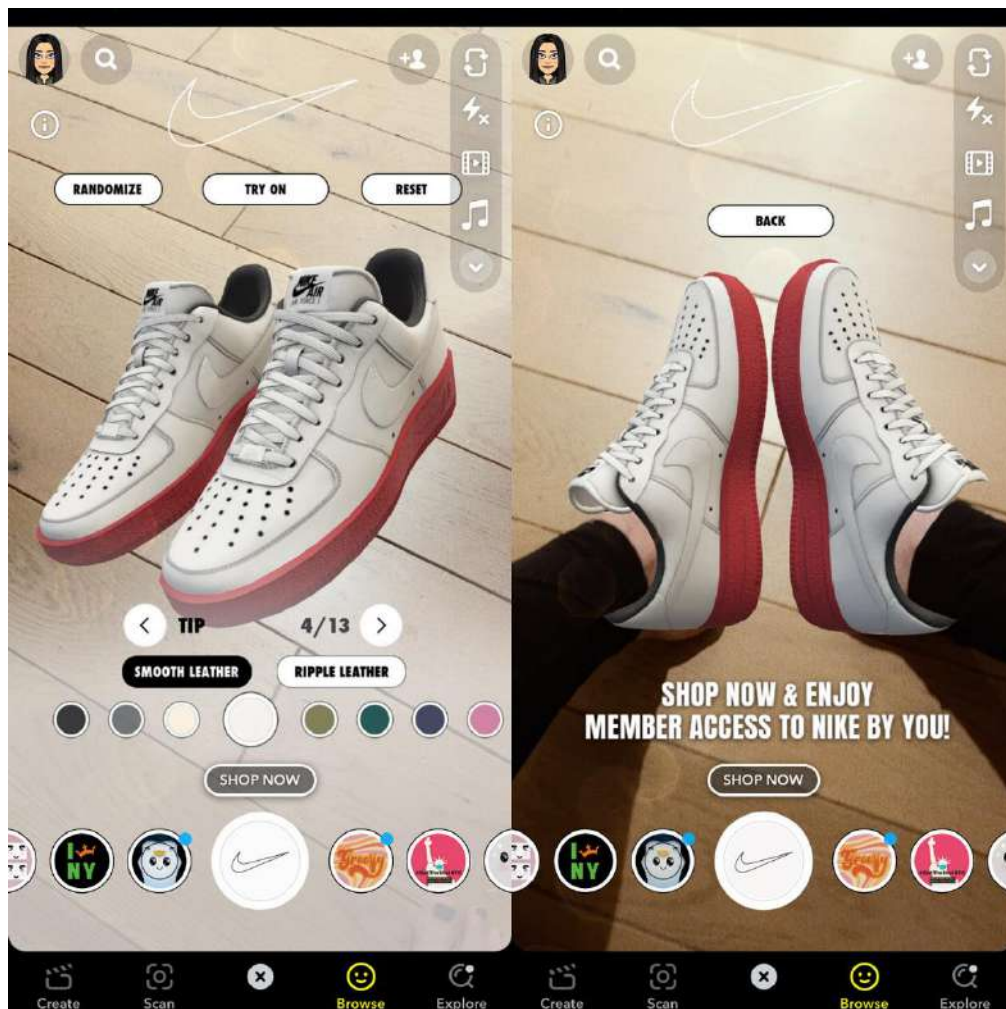


Рисунок 1.4.1 - Приклад примірки взуття за допомогою Nike By You

Ще одним прикладом може слугувати бренд окулярів Warby Parker. На вебсайті компанії можна знайти посилання на скачування її мобільного застосунку для Apple або Android аби протестувати функцію AR на собі (рисунок 1.4.2). Після переходу встановлення застосунку у потенційного покупця з'являється можливість обрати тип окулярів, переглянути усі доступні варіанти як зазвичай, а також додається функція “Virtual Try-On”. Тут користувач може обрати форму лінзи та колір.

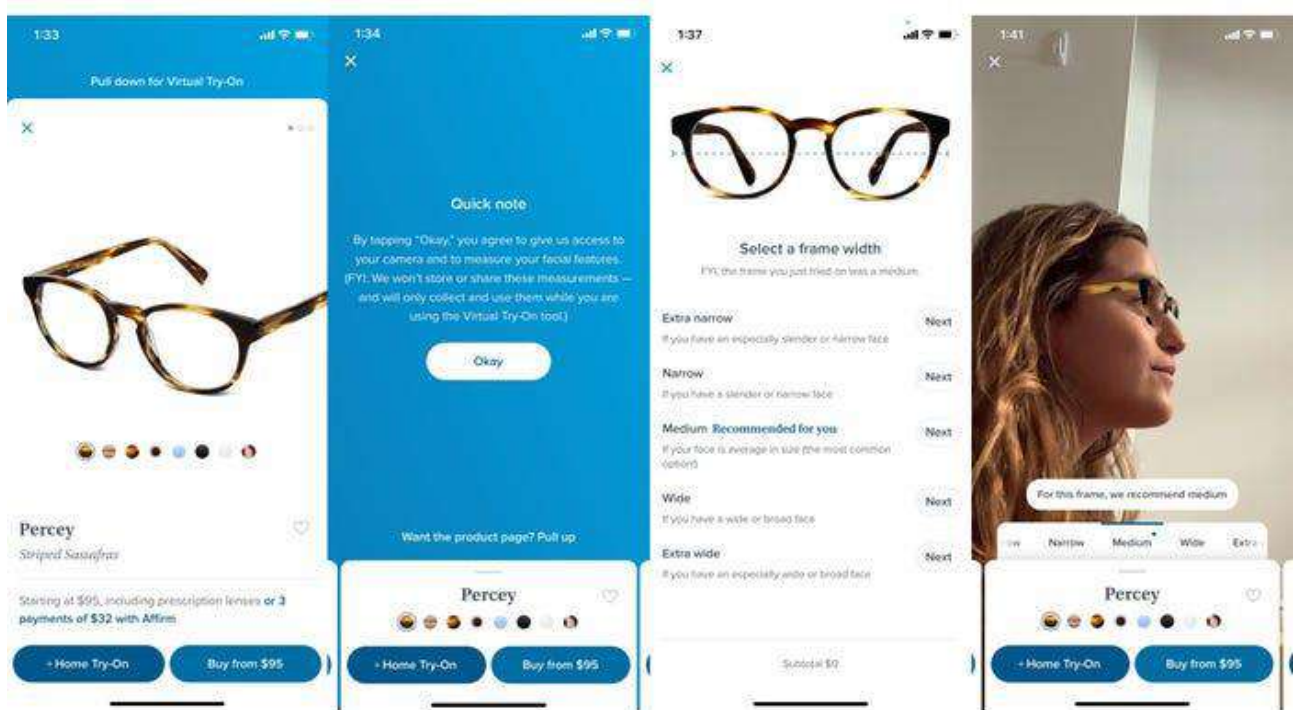


Рисунок 1.4.2 – Інтерфейс застосунку Warby Parker з можливістю AR-примірки

Усім знайомо, що онлайн шопінг добре відомий своїм головним недоліком - невідповідність очікуваного товару з реальним. Часто картинка в мережі Інтернет не сходиться з реальним товаром, який отримує клієнт після замовлення. Тому, такий спосіб інтерактивної примірки служить для вирішення або, принаймні, зменшення цієї проблеми.

Наступним популярним напрямком застосування AR-технологій є віртуальний перегляд 3D-моделі меблів. У цьому напрямленні головними

представниками посідають крупні компанії меблів, такі як ІКЕА, Wayfair, Room & Board та інші, хоча з часом, все більше й більше, доповнена реальність використовується середніми та малими видами магазинів.

До прикладу, ІКЕА оголосила про запуск мобільного застосунку ІКЕА Place від ARKit, що дозволяє користувачам переглядати віртуальні меблі у реальному просторі завдяки підтримці AR-технології (рисунки 1.4.3). Усі товари на платформі також представлені у тривимірному форматі та відповідають заявленим розмірам. Масштабування меблів та інших товарів відбувається з точністю 98%, що користувач навіть має можливість детально розглянути матеріал та текстуру певної одиниці товару. Підвищує залученість клієнтів також і можливість поділитися зображеннями та відео дизайну зі знайомими.

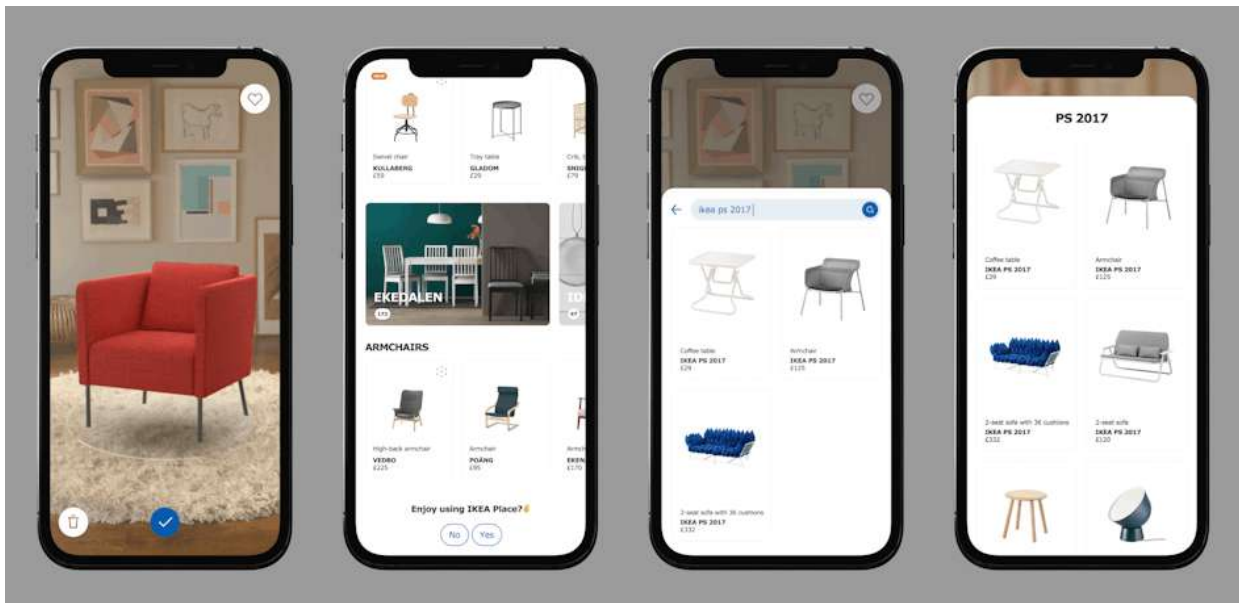


Рисунок 1.2.3 – Тестування застосунку ІКЕА Place з AR-технологією

Ще одним напрямком використання доповненої реальності – освітні застосунки. Взаємодія з 3D-об'єктами у цій галузі надає краще розуміння студентам природних явищ, історії, медицини та інших сфер навчання. Одним з таких навчальних застосунків є SkyView (рисунки 1.4.4). Це астрономічна платформа для вивчення та розуміння небесних тіл та сузір'їв, що використовує

AR. SkyView за допомогою датчиків гіроскопу та акселерометру надає можливість через камеру смартфона або іншого пристрою надавати інформацію про зірки та планети на небі у тому місцезнаходженні та реальному часі, де перебуває користувач та куди наводить камеру під час використання застосунку. Також, для кращого вивчення небесних тіл користувач може отримати детальну інформацію про той чи інший об'єкт, натискаючи прямо на нього. Такий інтерактивний досвід користувача тільки підкріплює його інтерес до науки та вивчення навколишньої природи.

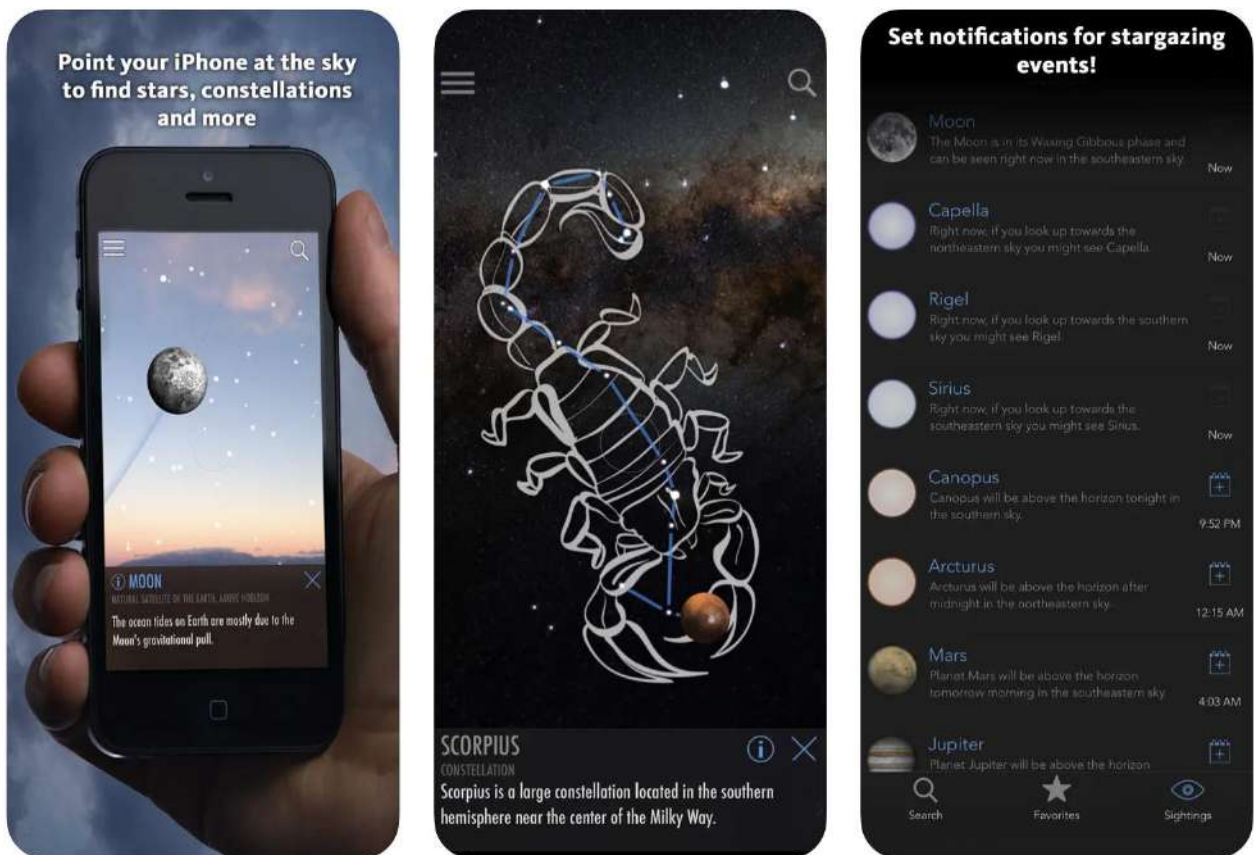


Рисунок 1.4.4 - Приклад тестування астрономічного застосунку SkyView

Окрім того, доповнена реальність широко відома також в ігровій сфері на прикладі знаменитої AR-гри – Pokémon GO (рисунок 1.4.5). Щоб виконувати завдання та ловити покемонів, користувачі використовують камеру свого пристрою, де у режимі реального часу з'являються віртуальні покемони. Гра

надає можливість безпосередньо взаємодіяти з ними та іншими гравцями, а також брати участь у змаганнях на виділених майданчиках.



Рисунок 1.4.5 - Скріншот гри Pokémon GO

Враховуючи усе вище зазначене, можна зробити висновок, що доповнена реальність використовується у численній кількості напрямків діяльності та виконує багато важливих місій. До того ж, імплементація AR-технологій у бізнес значно покращує задоволеність клієнта та підвищує дохід компанії.

РОЗДІЛ 2. Огляд засобів створення та імплементації технології AR

Для початку, для того, щоб інтегрувати AR-технологію у програмне забезпечення, потрібно мати тривимірну модель необхідного розширення для такого проєкту. Існує декілька форматів, що підтримують AR-технологію. Кожен з них призначений для різних цілей та застосувань.

Розглянемо основні формати:

1. `.gltf / .glb` – ці розширення описують матеріали, текстури, анімації, враховуючи скелетну, камери та сцени. Також, вони підходять для інтеграції прямо в браузер, кросплатформенних додатків і підтримують відкритий стандарт, а також компактну версію.
2. `.usdz` – цей архівований формат може містити моделі, текстури, анімації та матеріали. Таке розширення працює прямо в браузері iPhone. Також, `.usdz` початково був розроблений для iOS, тому не підтримується на Android. З недоліків, при використанні можуть виникати труднощі з експортом у цей формат.
3. `.fbx` – розширення, що підтримується такими популярними платформами для розробок 3D-моделей як Unity, Unreal Engine, Blender, Maya та інші. `.fbx` може містити анімацію, матеріали, камери та лайтінг. Він широко використовується в ігровій розробці, а також при розробці великих сцен зі складними моделями та анімаціями.
4. `.obj` – це застаріле розширення, що не підтримує анімації. Воно вважається незручним для транспортування та реалізації AR, проте часто використовується для простих сцен та конвертації у інші формати.

2.1 Джерела 3D-моделей для доповненої реальності

Отож, існує багато способів отримання 3D-моделей. Один з таких видів – самостійне створення моделі у відповідних програмах тривимірного моделювання або ж за допомогою 3D-сканування.

2.1.1 Створення власних моделей

Головними перевагами самостійного створення 3D-об'єкту є те, що модель користувача може бути унікальною завдяки використанню великої кількості інструментів на панелях програми. Водночас, процес розробки моделі може зайняти певний час та ресурси. Також, може постраждати якість результату залежачи від рівня навичок користувача та його майстерності використання програми.

Отже, одною з найпоширеніших програм з 3D-моделювання є Blender. Ця безкоштовна платформа об'єднує у собі полігональне та скульптурне моделювання, текстурування, рендеринг, анімацію, створення візуальних ефектів та багато інших інструментів для розробки необхідної моделі. Очевидно, що Blender притягує AR-розробників своєю гнучкістю та доступністю. Він також підтримує необхідні формати для доповненої реальності, а саме .glb/.gltf та .fbx.

Наступним найпоширенішим програмним забезпеченням для 3D-моделювання є Autodesk Maya. Він забезпечує полігональне та NURBS моделювання, анімацію, симуляцію та кастомізацію. Завдяки високій якості текстурування та освітлення, а також, інтеграції з ігровими рушіями Unity та Unreal Engine, Autodesk Maya можна вважати гідним конкурентом у своїй ніші.

Наступний інструмент, що не поступається попереднім, є Autodesk 3ds Max. Ця програма зосереджена більше на архітектурній та інтер'єрній візуалізації, ігровій індустрії і, звичайно ж, розробці AR/VR-контенту. За допомогою 3ds Max є можливість полігонального моделювання, текстурування, інтеграції з популярними ігровими рушіями, а також багатofункціонального рендерингу.

Звичайно ж, існує ще велика кількість інших платформ для якісного 3D-моделювання для подальшого використання у доповненій реальності. Проте, на основі вже розглянутих програм можна зробити висновок, що при виборі програмного забезпечення, з метою подальшої розробки AR-проєкту, варто

звертати увагу на функціональні можливості, текстурування, види моделювання та можливі розширення для збереження моделей.

У випадку, якщо користувач хоче спростити процес створення 3D-об'єкту або немає необхідних навичок моделювання та обізнаності у програмах, він може скористатись іншими способами створення тривимірних моделей реальних об'єктів і сцен.

Гарним прикладом таких способів може слугувати фотометрія. Її метод полягає у тому, що користувач робить серію кадрів об'єкту з різних ракурсів, потім ці кадри обробляються спеціальною програмою та визначаючи просторові координати кожної точки. Таким чином, створюється реалістична та високоякісна 3D-модель бажаного об'єкту, яку у подальшому можна імпортувати до AR-проєкту. Фотометрія являється оптимальним методом створення 3D-моделі, бо має багато переваг, спрямованих на відносно швидку та маловитратну за ресурсами роботу. Звичайно, якість результату та необхідний час обробки зображень напряму залежать від якості зроблених знімків та розміру об'єкту відповідно, проте для реалізації цього методу не потрібне ніяке додаткове обладнання або додаткові грошові витрати.

Якщо трохи детальніше розглянути спеціальні програми для фотометрії, то можна виокремити дві основні та найпопулярніші – Polycam та RealityScan.

Polycam – мобільний застосунок, що забезпечує експорт у необхідних для AR-проєкту розширеннях - .obj, .glb, .usdz, доступний інтерфейс для користувачів-початківців, підтримку LiDAR для вищої точності моделей, а також швидкість роботи. Вже за мить потрібна модель буде створена, яку одразу можна переглянути та експортувати. І все це доступно лише завдяки простій камері телефону (*рисунок 2.1.1.1*).

Водночас, RealityScan є ще одним, не менш потужним, інструментом для 3D-моделювання також у вигляді мобільного застосунку (*рисунок 2.1.1.2*). Функціональність цієї платформи здебільшого дуже схожа на попередню, просте основні його переваги – безкоштовна інтеграція з Unreal Engine та

створення більш точних моделей з високою деталізацією. Втім, RealityScan доступний лише для операційної системи iOS, що обмежує його доступність.

Отже, фотометрія є доступним шляхом для створення достатньо деталізованого 3D-об'єкту для майбутньої інтеграції у доповнену реальність.

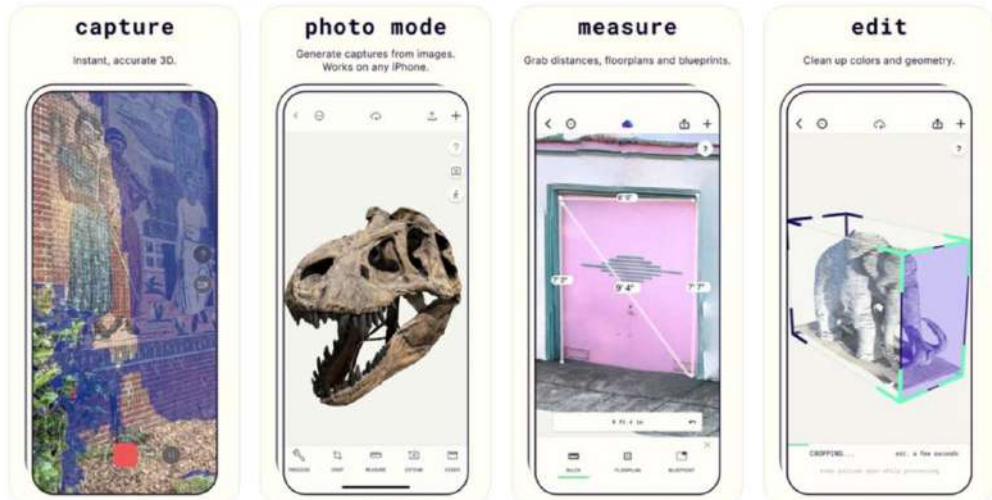


Рисунок 2.1.1.1 - Тестування застосунку Polycam

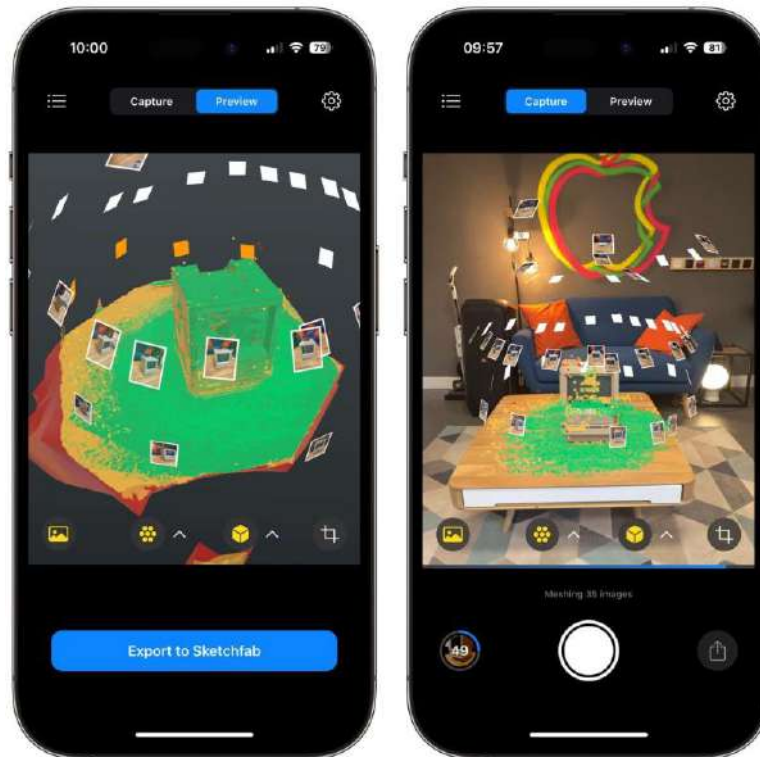


Рисунок 2.1.1.2 – Тестування застосунку RealityScan

Підсумовуючи, можна зауважити, що існує багато різноманітних джерел для моделювання тривимірних об'єктів для подальшого імпорту у проєкт доповненої реальності. Тим не менш, основні з них – 3D-моделювання власноруч та фотометрія. Для першого способу існує безліч програм, в залежності від типу бажаної моделі, її деталізації, текстурування та інших характеристик. Для другого способу необхідно використовувати спеціальні програми для отримання очікуваного результату, що залежить від якості зроблених знімків. Окрім цього, при виборі програмного забезпечення також важливими є реально визначити технічні навички користувача і підібрати зручний метод саме для нього.

2.1.2 Готові бібліотеки моделей

Ще одним способом отримання 3D-моделі є просте знаходження потрібної моделі у готовій бібліотеці. Такі бібліотеки значно спрощують процес та дозволяють зосередитись конкретно на етапі розробки ар-проєкту. Найпопулярнішими платформами, де можна знайти або продати 3D-моделі являються Sketchfab, TurboSquid та CGTrader. Ці бібліотеки надають різноманітні варіанти моделей, в тому числі і для AR-проєктів, бо підтримують різний формат.

Sketchfab – одна з таких платформ, яка підтримує формати .glb, .gltf, .obj, .fbx для подальшого імпорту в AR. У бібліотеці є платні та безкоштовні моделі, що можна переглянути у 3D-форматі миттєво, бо платформа використовує технологію WebGL. Також, черговою перевагою Sketchfab є можливість покупки моделей з анімаціями, що можна легко інтегрувати у проєкт та спростити процес розробки.

TurboSquid – ще одна з найбільших бібліотек для торгівлі 3D-моделями з великим вибором, що підтримує потрібні формати для AR-технології. До того ж, ця платформа підтримує перевірку файлів задля зменшення випадків помилок у проєктах.

CGTrader – також являється прикладом доступної та гнучкої бібліотеки для покупки та продажу 3D-моделей. Функціональність особливо не відрізняється від вже вищезазначених, проте головна особливість – оптимізовані моделі, зокрема для мобільних девайсів.

Отже, вибір бібліотеки для користувача буде залежати від вимог до якості моделей, фінансових можливостей та від специфікації AR-проєкту.

2.1.3 Платформи з вбудованими моделями

Звичайно, існує безліч інших джерел, де можна придбати потрібну 3D-модель. Також, існують платформи для розробки доповненої реальності, що містять вже вбудовані моделі. Цей варіант створення 3D-моделі є дуже економним та значно спрощує роботу.

Одна з таких платформ - Reality Composer. Вона має зручний інтерфейс та надає доступ до готових 3D-моделей та шаблонів анімації. До того ж, вона легко інтегрується з ARKit та підтримує необхідні розширення для доповненої реальності.

Наступний інструмент орієнтований вже більше на веб-дизайнерів та маркетологів – Adobe Aero. Ця платформа має велику бібліотеку 3D-моделей, без значних зусиль інтегрується з іншими продуктами Adobe та підтримує кросплатформність для операційних систем iOS та Android.

До того ж, можна знайти багато альтернатив, окрім вже запропонованих методів, такі як: замовлення моделей у 3D-художників, тестування усіляких AI-інструментів для конвертування 2д зображення у тривимірну модель, пошук open source платформ з відкритими репозиторіями. Такі способи можуть бути значно дешевшими або ж зовсім безкоштовними, дозволяючи автоматизувати процес розробки. Однак, вони потребують додаткових перевірок на валідність файлів та у частих випадках обмежують деталізацію моделей.

Таким чином, можна зробити висновок, що для отримання 3D-моделі для розробки проєкту доповненої реальності можна використовувати величезну кількість ресурсів. Якщо є потреба створити унікальну деталізовану модель за індивідуальними параметрами, тоді варто використовувати програми для 3D-моделювання. Якщо ж потрібно просканувати існуючий об'єкт та створити з нього точну тривимірну цифрову модель, тоді краще використовувати фотометрію. У разі коротких термінів роботи та менш важливій якості результату, можна використовувати платформи вже з готовими вбудованими моделями.

2.2 Порівняння платформ та середовищ розробки AR

Коли користувач вже має необхідні 3D-моделі потрібного розширення, він має зупинитися на якомусь з програмних забезпечень для створення проєкту доповненої реальності. Звичайно, є багато альтернатив, тому постає задача вибору найбільш оптимального варіанту.

Тож, розглянемо наступні платформи для розробки AR-проєктів: ARKit, ARCore, Unity з застосуванням AR Foundation, Unreal Engine з застосуванням AR Plugin та платформи WebAR.

2.2.1 ARKit

ARKit – це платформа, розроблена компанією Apple, що дозволяє розробляти доповнену реальність на девайсах операційної системи iOS (*рисунок 2.2.1.1*). Це програмне забезпечення має багато переваг у використанні. Одним з них є алгоритм SLAM, що допомагає зробити розміщення об'єктів точнішим без потреби у фізичних маркерах. Окрім цього, пристрої Apple, починаючи з iPhone 12 Pro, підтримують технологію LiDAR сканування, тому ARKit інтегрує отримані дані зі скану, що значно покращує оптимізацію проєкту. Взаємодія з LiDAR безпосередньо впливає на якість розміщення об'єктів у просторі та покращує відстеження глибини сцени за будь-яких умов освітлення. Також, ARKit дозволяє розпізнавати 3D-об'єкти та

образи. За допомогою цього стає можливим активація AR-контенту при скануванні певних зображень. Ще однією перевагою ARKit можна визначити можливість відстеження обличчя та захоплення руху користувача. Ці функції значно вдосконалюють інтерактивний досвід користувача, так само, як і підтримка створення спільних AR-сцен, де може відбуватися взаємодія між декількома користувачами. До популярних прикладів застосування ARKit можна віднести Pokemon GO, IKEA Place, Measure App та інші.

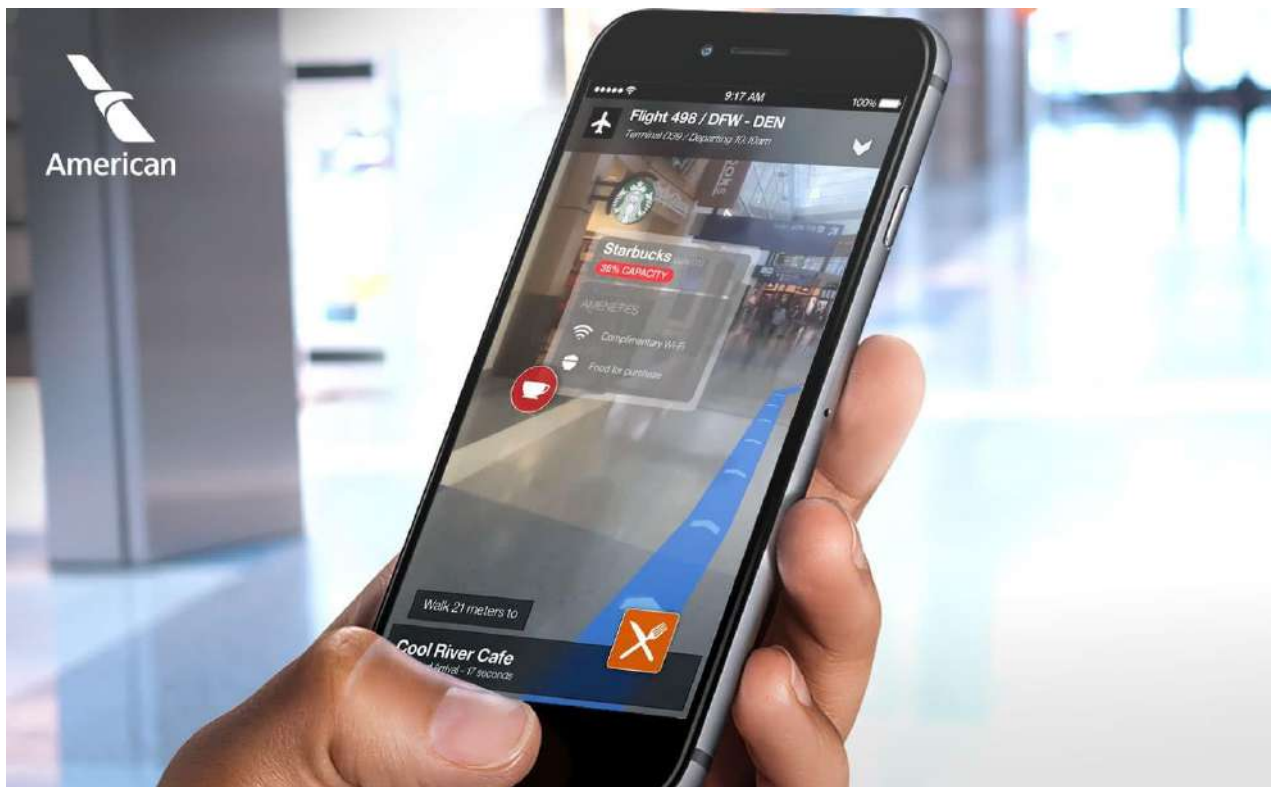


Рисунок 2.2.1.1 – Приклад застосування ARKit компанією American Airlines

2.2.2 ARCore

ARCore – це ще одна платформа для створення доповненої реальності (рисунки 2.2.2.1). Вона розроблена компанією Google для пристроїв операційної системи Android. Як і попередня платформа, ARCore забезпечує роботу технології SLAM, розпізнавання образів, поверхонь та площин. Також, це програмне забезпечення адаптує відображення об'єктів, аналізуючи навколишнє освітлення. Окрім цього, в ARCore є створення «анкерів» - фіксованих точок у просторі для прив'язки віртуальних об'єктів та інтеграція з

різними іншими програмами. До прикладу, платформа взаємодіє з такими рушіями, як Unreal Engine, Unity та інтегрується з сервісами Google Cloud та AI. Представниками застосування ARCore можуть бути Google Measure, Just a Line та Houzz View in My Room.



Рисунок 2.2.2.1 - Приклад застосунку ARCore та Android Studio

2.2.3 Unity з AR Foundation

Unity – один з найпопулярніших рушіїв для розробки ігрових застосунків, з можливістю створення AR та VR-проектів. Програми, розроблені за допомогою Unity, підтримуються на різних платформах та операційних системах. Цей редактор надає потужні інструменти створення сцен з 3D-графікою, анімаціями, реалістичною фізикою та інтерактивністю.

AR Foundation – це фреймворк від Unity, що дозволяє створювати кросплатформову доповнену реальність. Цей набір скриптів слугує надбудовою, яка підтримується і через ARKit, і через ARCore. Це забезпечує необхідні умови для роботи на iOS, так само, як і на Android, без необхідності писати окремий код під кожен платформу, надаючи уніфікований API (рисунки 2.2.3.1).

Саме поєднання Unity та AR Foundation дає можливість розробити якісний продукт доповненої реальності з гарними анімаціями та інтерактивністю, який буде підтримуватися на багатьох пристроях та операційних системах. З особливостей можна виділити інструменти для тестування та симуляції, створення складних сцен з гнучкою кастомізацією. Водночас, робота з Unity та AR Foundation може бути складною для новачків, адже потрібні поглиблені знання у програмі та програмування на C#. Прикладом використання Unity та AR Foundation можуть слугувати JigSpace, Wonderscope або оновлена версія Pokemon GO.



Рисунок 2.2.3.1 – Тестування застосунку на Unity та AR Foundation

2.2.4 Unreal Engine з AR Plugin

Unreal Engine – наступний варіант для розробників, яким потрібна реалізація AR або VR-застосунку з високою якістю та складною структурою. Unreal Engine теж являється одним з найпотужніших рушіїв для створення кросплатформних програм з доповненою реальністю за допомогою власного фреймворку AR Plugin. Особливістю цього програмного забезпечення є висока складність проєктів з деталізованими і реалістичними об'єктами та анімаціями. Також, використання технології візуального програмування BluePrints дозволяє створювати логіку додактів без необхідності написання коду за допомогою

візуальних блоків. Зазвичай, тандем Unreal Engine з AR Plugin використовують для реалізації архітектурної візуалізації, безлічі ігрових проєктів з додавання технологій AR та VR та іншого. Як приклад можна навести NASA Mars 2030 AR Project та Real Estate AR Visualization. Очевидно, що такий масштаб має жорсткіші вимоги до апаратного забезпечення, тому для швидкої та стабільної роботи кінцевого застосунку необхідна якісна оптимізація.

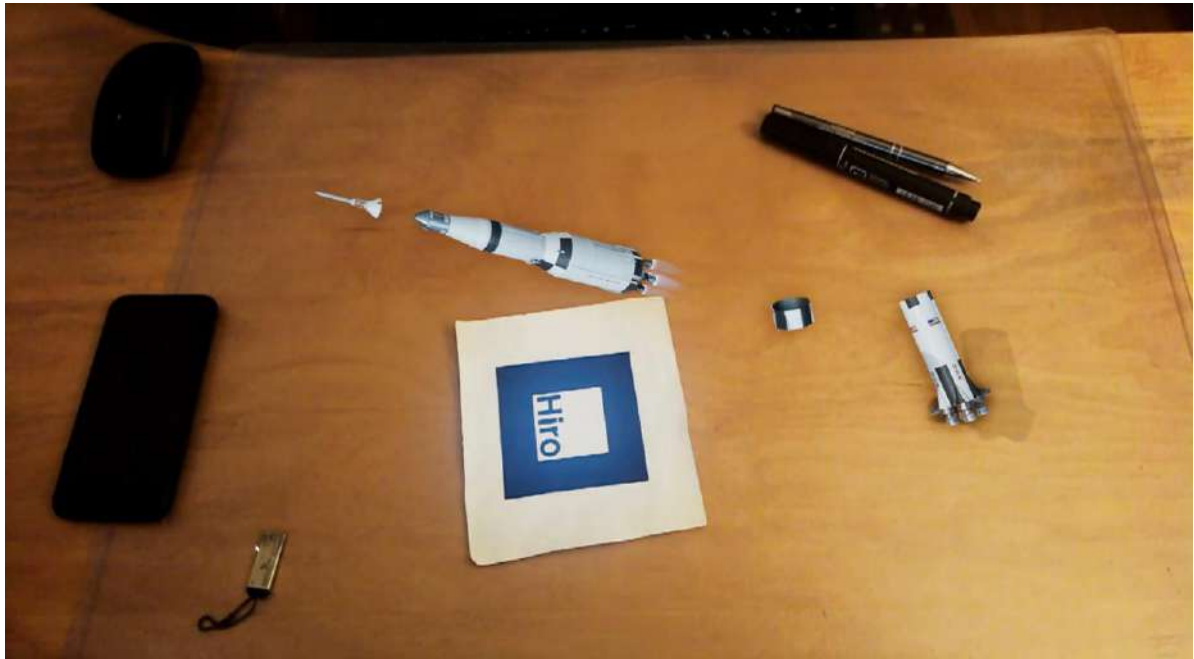


Рисунок 2.2.4.1 – Скрін роботи застосунку на Unreal Engine з AR Plugin

2.2.5 WebAR платформи

З доповненою реальністю можна взаємодіяти безпосередньо у браузері вашого пристрою без завантаження окремих застосунків. Це стає можливим за допомогою технології WebAR. Головна особливість платформ, що підтримують WebAR, – це отримання AR-досвіду, перейшовши за QR-кодом. Такий спосіб дозволяє підтримувати роботу кінцевого застосунку на більшості смартфонів та планшетів, що є дуже зручним для користувача. Яскравим представником WebAR являється 8th Wall.

8th Wall – платформа, де можна створювати сцени доповненої реальності з різними анімаціями. Така платформа використовує .glb та .gltf розширення

для 3D-моделей, підтримує маркерний, безмаркерний та геолокаційний трекінг і легко інтегрується з WebGL та Three.js. На жаль, чим складніше сцени, тим вони працюють повільніше, бо 8th Wall запускається з браузеру, що задає певне навантаження. Прикладами застосування цієї платформи можуть бути: Brand Campaigns або ж Snapchat WebAR.

При використанні 8th Wall приємно тішить безліч відео-уроків на YouTube, а також на офіційному сайті платформи. Деталізована документація та приклади проєктів від простого до складного рівню спонукають користувачів до вивчення нових технологій та поглиблення знань у сфері 3D.



Рисунок 2.2.5.1 – Скрін роботи програми, розробленої на 8th Wall

Враховуючи усі технічні та функціональні аспекти тестування програм для створення AR-проєкту, порівняємо платформи для розробки доповненої реальності:

Середовище	Операційна Система	Сумісність платформ	Формат моделей	Необхідні знання мов для роботи	Переваги	Недоліки
ARKit	iOS	iPhone, iPad	.usdz, .reality, .obj, .fbx	Swift	LiDAR, широкий функціонал, реалістичність та інтерактивність	Обмежена кількість користувачів, підтримка лише на iOS
ARCore	Android	Усі смартфони та планшети на Android	.fbx, .obj, .glb, .gltf	Java/Kotlin	Доступність, інтеграція та підтримка Google, багато сумісних пристроїв	Складність оптимізації, різноманітність апаратного забезпечення
Unity + AR Foundation	iOS, Android	Кросплатформність	.fbx, .obj, .glb, .gltf	C#	Різноманітний вибір інструментів, велика спільнота, ARKit та ARCore	Важкі за розміром додатки, необхідність оптимізації
Unreal Engine + AR Plugin	iOS, Android	Кросплатформність	.fbx, .obj	C++	Якісна графіка, потужні інструменти анімації та фізики, Blueprints	Труднощі в опанування, ресурсоемність, важкі за розміром додатки
8th Wall	Браузер (iOS/Android)	Кросбраузерність	.glb, .gltf	JavaScript	Відсутність потреби встановлення додатку, швидкий браузерний доступ, широка сумісність	Залежність від браузера та якості Інтернету, обмежена продуктивність

Порівняння проводилось за такими критеріями, як операційна система, сумісність платформ, формат підтримуваних моделей та інструментів, переваги, недоліки та необхідні знання мов програмування. З урахуванням результатів порівняння стає зрозуміло, що кожне з п'яти розглянутих середовищ розробки доповненої реальності має свої унікальні особливості використання. ARKit більше підходить для розробки реалістичних та інтерактивних сцен для пристроїв iPad та iPhone. ARCore – це більше про доступність, широку аудиторію, інтеграцію та підтримку Google сервісів. Використання Unity з AR Foundation краще підходить для складних сцен з кросплатформністю та гнучкою розробкою. У той же час, Unreal Engine з AR Plugin – про потужний функціонал та високу деталізацію. На останок, 8th Wall відрізняється доступністю та застосуванням через браузер.

Отож, провівши аналіз усіх платформ з підтримкою AR-технології, можна зробити висновок, що 8th Wall – найдоступніша для дослідження платформа, яка буде детально протестована у практичній частині.

2.3 Технології взаємодії користувача з AR-середовищем

Після визначення середовища розробки, а також, специфіки та мети кінцевого продукту, постає важливим питання вибору технологій, з якими користувач буде взаємодіяти. До таких технологій відносяться: трекінг, позиціонування, різні фреймворки, контекстна адаптація, інтерактивна взаємодія та інше. У цьому розділі буде детально розглянуті основні аспекти, їхні сильні сторони та недоліки.

2.3.1 Основні технології трекінгу та позиціонування в AR

Взагалі, трекінг та позиціонування - це визначальні технології у проєкті доповненої реальності, оскільки вони дозволяються коректно розміщувати віртуальні 3D-об'єкти. Трекінг та позиціонування відповідають за розташування та орієнтацію користувача відносно навколишнього простору.

Існує декілька базових підходів до визначення позиціонування. Кожен з них застосовується залежно від мети проєкту, а також, від технічних можливостей пристрою. Отож, розглянемо основні з них.

2.3.1.1 Маркерний трекінг

Маркерний трекінг – технологія, що використовує спеціальні «маркери» або «мітки» для коректного визначення позиції, орієнтації або ж для прив'язки об'єктів у сцені. Коли камера пристрою направляється на маркер, вона сканує його розмір та положення. Після цього об'єкти, які прив'язані до цього маркеру, відображаються у віртуальному просторі у правильній пропорції відносно цього маркеру. Основними перевагами такої технології є висока точність позиціонування прив'язаних об'єктів. Також маркерний трекінг зовсім не складний у реалізації та підходить для тестування на простих пристроях, оскільки займає невеликий об'єм обчислюваних ресурсів. Водночас, використання такої технології обмежує AR-досвід в рамках маркеру та

погіршує відстеження розташування користувача при поганому освітленні. Не зважаючи на недоліки, маркерний трекінг часто впроваджується в освітні AR-застосунки, різні культурні заходи, наприклад, музеї та виставки, а також, у деякі рекламні кампанії.

2.3.2.2 Безмаркерний трекінг

Наступний вид трекінгу – безмаркерний, або SLAM. Ця технологія дозволяє визначати розташування користувача та побудувати мапу навколишнього простору вже без маркерів. Оточення аналізується за допомогою камер та датчиків руху. Коли випробовується AR-проект, що задіює безмаркерний трекінг, користувач може вільно переміщатись у просторі без усіляких обмежень. Таке відслідковування позиціонування пристрою надає розробникам свободу у масштабуванні сцени. Однак, якість правильного відображення об'єкту у просторі напряду залежить від якості камер та сенсорів пристрою, що обумовлює вищу вимогливість до апаратних ресурсів. Зазвичай, безмаркерний трекінг використовують у різних застосунках доповненої реальності, де AR-досвід відбувається без прив'язок до певного місцезнаходження. Гарним прикладом застосування такого трекінгу можуть бути ігри з відкритим віртуальним світом та інтерактивністю у ньому.

2.3.3.3 GPS та геолокаційний трекінг

Використовуючи супутникову навігацію, така технологія дозволяє визначати положення користувача. Якщо супутниковий сигнал недостатній, GPS навігація також може доповнюватися використанням інших сенсорів, таких як Bluetooth, Wi-Fi та інші. Такий вид трекінгу ідеально підходить для масштабних відкритих віртуальних просторів, бо не потребує ні маркерів, ні високих вимог до камери пристрою. До слабких сторін можна віднести похибку до декількох метрів в залежності від якості сигналу супутника. Найчастіше за все геолокаційний трекінг використовується у туристичних екскурсіях, маршрутах та навігаціях по містах з інтеграцією доповненої реальності.

Підсумовуючи, можна зауважити, що вибір правильного трекінгу для коректного відображення об'єктів у віртуальному просторі безпосередньо залежить від типу кінцевого проєкту та його призначення. Для використання технології доповненої реальності на одній локації, без значного переміщення у середовищі краще застосовувати маркерний трекінг. Якщо користувачу потрібно вільно переміщатися в обмеженому просторі або ж на відкритій місцевості, тоді у якості технології відстеження позиції краще використовувати безмаркерний або ж геолокаційний трекінг відповідно.

2.3.2 Інструменти та фреймворки для інтерактивності

Доповнена реальність це не тільки про відображення 3D-об'єктів у віртуальному середовищі, вона також включає у себе певну взаємодію з користувачем. Така інтерактивність може відрізнятися в залежності від специфікації проєкту, реалізуючись за допомогою різних технологій. Тому у цьому підрозділі будуть розглянуті основні технології взаємодії з користувачем.

2.3.2.1 Інструменти для розпізнавання жестів у AR

Розпізнавання жестів – це набір інструментів, що конвертують рухи тіла користувача у певні команди для управління AR-застосунком. Принцип роботи полягає у розпізнаванні заздалегідь запрограмованих жестів або рухів користувача. Коли камера захоплює картинку руху тіла, алгоритм машинного навчання аналізує знаходження, наприклад, суглобів пальців та виділяє ключові точки. Після розпізнавання руху дані передаються у застосунок, і відбувається команда взаємодії з віртуальним об'єктом. Частіше за все, такий інтерактив стосується обертання, масштабування або ж переміщення об'єкту у просторі. Яскравим прикладом таких інструментів може бути ManoMotion, Google MediaPipe Hands, Ultraleap та інші інструменти для розпізнавання жестів рук користувача.

2.3.2.2 Голосове керування в AR-застосунках

Голосове керування дозволяє керувати функціями за допомогою голосу. Існують декілька ключових аспектів, що забезпечують роботу голосового керування. Один з них – сервіси speech-to-text, що аналізують голос користувача та перетворюють його у відповідні команди. Також, важливою є і інтеграція з голосовими асистентами, наприклад Siri або Alexa, для обробки команд. Ще один важливим етапом роботи голосового керування є динамічна зміна поведінки AR -застосунку в залежності від команди користувача. Головними перевагами керуванням голосом є зручна взаємодія зі сценою та меню, а також підтримка користувачів з обмеженими фізичними можливостями. Разом з тим, розробляючи проєкт доповненої реальності з голосовим керуванням може виникати проблема підтримки різних мов та діалектів. Також, необхідна висока якість звуку та мінімізація зовнішніх шумів. Прикладами платформ з голосовим керуванням можуть слугувати Wonderscope та Magic Leap One.

2.3.2.3 Контекстна адаптація інтерфейсу

Контекстна адаптація – являється достатньо важливим підходом у взаємодії користувача з AR-застосунком, оскільки динамічна зміна зовнішнього вигляду, в залежності від контексту використання, сприяє зменшенню візуального шуму. Це робить інтерфейс значно природнішим та зручнішим. Контекстна адаптація можлива завдяки використанню певних сенсорів пристрою для збору даних, фреймворкам для створення адаптивних UI-елементів та програмній логіці, що відстежує контекстні параметри для подальшої зміни елементів UI. Взагалі, адаптивний інтерфейс застосунку доповненої реальності полягає у динамічній зміні компонентів, спираючись, в основному, на навколишнє освітлення, положення та відстань користувача до певної локації або об'єкту. Прикладом контекстної адаптації може бути зміна кольору та контрасту елементів, розмір кнопок та розташування панелей,

затемнення або підсвічування кнопок, переміщення об'єктів задля зручного перегляду більш важливих компонентів сцени, відображення підказок та інше.

Отже, інструменти для інтерактивності у доповненій реальності сприяють поліпшенню якості взаємодії і загального враження від використання AR-застосунку.

РОЗДІЛ 3. Реалізація прототипу онлайн-каталогу з AR-переглядом меблів

У цьому розділі буде представлений покроковий процес створення проєкту за допомогою AR-технологій, а також його імплементація у вебсередовище. У якості способу трекінгу моделі та її позиціонування буде реалізований безмаркерний трекінг. Також, задля швидшої роботи з меншою витратністю ресурсів буде проведено розширення деякого функціоналу.

Для початку, за допомогою платформ Sketchfab та gltf Editor була підготовлена певна кількість 3D-моделей для майбутнього використання – пошук та редагування відповідно.

3.1 Розробка проєкту доповненої реальності

Наступний крок – це розробка проєкту, використовуючи технологію WebAR та фреймворк A-Frame на платформі 8h Wall. Для реалізації практичної частини було обране саме це середовище, бо головною його перевагою є отримання AR-досвіду прямо у браузері, без встановлення зайвих додатків. Аби переглянути модель, можна просто перейти за QR-кодом. Розробка самого проєкту відбувається також у браузері, що є дуже зручним.

Отож, спочатку визначаються необхідні компоненти та плагіни сцени – обробка жестів, можливість взаємодіяти через контролери, підтримка WebAR, повідомлення про помилки та інше (рисунок 3.1.1).

```
<a-scene
  button-controller
  xrextras-gesture-detector
  xrextras-almost-there
  xrextras-loading="loadBackgroundColor: #007AFF; cameraBackgroundColor: #5AC8FA; loadImage: #myCustomImage; loadAnimation: pulse"
  xrextras-runtime-error
  renderer="colorManagement: true;"
  xrweb>
```

Рисунок 3.1.1 - Визначення необхідних компонентів

Наступним кроком, використовуючи базові A-Frame контейнери, визначаються основні внутрішні елементи сцени – завантаження та відображення моделей, камери, освітлення та кнопок (рисунок 3.1.2).

```

<a-camera
  id="camera"
  position="0 0 0"
  raycaster="objects: .draggable"
  cursor="fuse: false; rayOrigin: mouse;">
</a-camera>

<a-entity light="type: directional; intensity: 0.8; castShadow: true;" position="1 2 1" shadow></a-entity>

<a-light type="ambient" intensity="0.7"></a-light>

```

Рисунок 3.1.2 - Визначення внутрішніх елементів сцени

Для того, аби користувач міг взаємодіяти зі сценою, була прописана правильна робота кнопок інтерактивності з 3D-моделлю. До прикладу, функція `attachModelHandlers` відповідає за дію однієї з кнопок переміщення об'єкта (рисунок 3.1.3). Вона приймає на вхід параметр `model`, що має поле `position`. Саме за допомогою цього поля функція змінює позицію 3D-об'єкта збільшуючи або зменшуючи координати потрібних осей.

```

function attachModelHandlers(model) {
  document.querySelector('#forward-button')?.addEventListener('click', () => {
    const position = model.getAttribute('position')
    model.setAttribute('position', {x: position.x, y: position.y, z: position.z + 0.5})
  })
}

```

Рисунок 3.1.3 - Функція `attachModelHandlers()`

Функція, що відповідає за обертання 3D-об'єкту навколо себе, дещо відрізняється від попередньої. Для реалізації правильної роботи кнопки оберту використовується змінна, яка відповідає за стан об'єкту – чи він знаходиться у статичному положенні чи обертається (рисунок 3.1.4). Отож, якщо користувач натискає на кнопку оберту, то об'єкт обертається на 1 градус кожні 20 мс навколо себе.

```

let rotateInterval = null

const rotateButton = document.querySelector('#rotate-button')

rotateButton?.addEventListener('click', () => {
  if (rotateInterval) {
    // якщо вже обертається - зупинити
    clearInterval(rotateInterval)
    rotateInterval = null
  } else {
    // якщо не обертається - запустити
    rotateInterval = setInterval(() => {
      const rotation = model.getAttribute('rotation')
      model.setAttribute('rotation', {x: rotation.x, y: rotation.y + 1, z: rotation.z})
    }, 20)
  }
})

```

Рисунок 3.1.4 - Змінна, що відповідає за стан об'єкту

Ще однією перевагою платформи 8th Wall є вбудований симулятор, що динамічно відображає роботу проєкту. Тут можна обрати пристрій та навколишнє середовище за допомогою яких буде відображатися доповнена реальність. Звичайно ж, такий симулятор надає можливість одразу виправляти помилки при розробці та з легкістю тестувати застосунок. Окрім симулятора, також є можливість протестувати сцену зі свого пристрою, просто відсканувавши QR-код (рисунок 3.1.5).

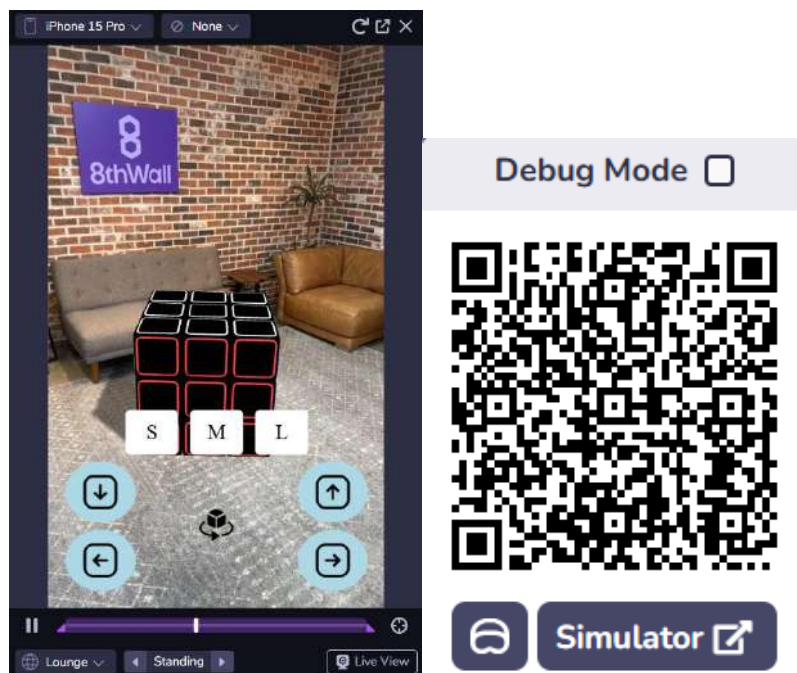


Рисунок 3.1.5 - Симулятор та відображення QR-коду

Отже, фреймворк A-Frame дозволяє розробникам визначати компоненти та елементи 3D-сцени, їх характеристики та налаштування. А використання браузерної платформи 8th Wall значно спрощує процес роботи, бо не вимагає додаткових завантажень або наявності інших програм та інструментів.

3.2 Інтеграція доповненої реальності на вебсторінку

Для того, аби додати проєкт доповненої реальності у вебсередовище, необхідно спочатку створити вебсайт. У його якості буде розроблений каталог меблів, де у деталях кожного предмету можна знайти QR-посилання для перегляду AR-моделі цього предмету. Технологічний стек сайту представляє собою використання Node.js та Express.js для створення серверних додатків та обробки запитів відповідно (рисунки 3.2.1).

```
const app : any | Express = express();

// Start the server
app.listen( port: 3000, hostname: () : void => {
  console.log('Server is running on port 3000');
});
```

Рисунок 3.2.1 - Запуск серверу

Також, для швидкого і спрощеного створення дизайну інтерфейсу був задіяний фреймворк Bootstrap 5 (рисунки 3.2.2). Він дозволяє класам, які використовуються в HTML, отримувати потрібні стилі та розташування.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
```

Рисунок 3.2.2 - Застосування Bootstrap

Для самого переходу до AR-сцени застосовується сканування QR-коду з телефону або планшета. Для того, аби додати QR-код до сайту використовується зовнішній вебсервіс qrserver.com. Після формування правильного URL на потрібну модель, код передає посилання на вебсервіс,

який у свою чергу повертає згенероване зображення . Таким чином, користувач може отримати відображення коректного QR-коду на вебсторінці (рисунки 3.2.3).

```
//Використання arUrl з furniture.json для створення правильного QR-коду
const qrCodeLink = `https://yelyzavetapetrenko.8thwall.app/camera/?model=${item.arUrl}`;
console.log('qrCodeLink:', qrCodeLink);

document.getElementById('qrCodeImage').src = `https://api.qrserver.com/v1/create-qr-code/?data=${encodeURIComponent(qrCodeLink)}&size=200x200`;
```

Рисунок 3.2.3 - Генерація QR-коду

У той же час, необхідні моделі проєкту зберігаються у форматі .json та мають такі поля, як “id”, “name”, “image”, “shortDescription”, “description”, “arUrl” (рисунки 3.2.4).

```
{
  "id": "fridge",
  "name": "Холодильник",
  "image": "fridge.jpg",
  "shortDescription": "Енергоефективне зберігання продуктів",
  "description": "Збережіть свіжість ваших продуктів на найвищому рівні з нашим сучасним холодильником – поєднанням стильного дизайну",
  "arUrl": "fridge"
},
```

Рисунок 3.2.4 - Скрін файлу furniture.json

3.3 Розширення функціональних можливостей

З метою зменшити витратність ресурсів та зробити проєкт більш універсальним, було реалізоване певне розширення функціоналу. Підґрунтям для нього стала проблема створення нового проєкту під кожну модель. Саме тому було використано властивість window.location.pathname об’єкта window.location, яка повертає шляхову частину URL поточної сторінки.

Його робота полягає в отриманні шляху після домену властивістю pathname задля відображення необхідної користувачу моделі. На практичному прикладі можна спостерігати, як window.location.pathname вилучає ідентифікатор URL для динамічного відображення контенту, перед цим розбивши шлях на масив (рисунки 3.2.5).

```
const itemId = window.location.pathname.split('/').pop(); // Витягаємо ідентифікатор елемента з URL
const item = furnitureData.find(f => f.id === itemId);
```

Рисунок 3.2.5 - Отримання ідентифікатора елемента

Далі у проєкті 8th Wall з поточного URL сторінки зчитується параметр після знаку «?» та співставляється до існуючого за допомогою switch (рисунок 3.2.6). Наступним кроком, атрибут gltf-model набуває значення, наприклад, #fridge, яке потім буде оголошено у сцені. Далі задаються початкові характеристики розташування об'єкту – масштаб, тінь та позиція. Згодом, новий елемент додається до <a-scene> та викликається функція обробників для керування рухами моделі.

```
// Встановлюємо модель мебелі в залежності від значення параметру
if (modelName) {
  switch (modelName) {
    case 'fridge':
      {
        const oldModel = document.querySelector('#model')
        if (oldModel) oldModel.parentNode.removeChild(oldModel)
        const newModel = document.createElement('a-entity')
        newModel.setAttribute('id', 'model')
        newModel.setAttribute('class', 'draggable')
        newModel.setAttribute('gltf-model', '#fridge')
        newModel.setAttribute('position', '0 0 -3')
        newModel.setAttribute('scale', '1 1 1')
        newModel.setAttribute('shadow', '')
        document.querySelector('a-scene').appendChild(newModel)
        attachModelHandlers(newModel)
      }
    break
  }
}
```

Рисунок 3.2.6 - Оператор switch

У результаті розробленої практичної частини роботи було створено прототип онлайн-каталогу меблів. Кожен товар можна переглянути окремо на його власній сторінці з можливістю візуалізації у реальному просторі за допомогою технології AR (рисунок 3.2.7). За серверну частину вебсайту відповідали Node.js та Express.js, за розробку проєкту доповненої реальності –

платформа 8th Wall, а для підготовки необхідних моделей було використано Sketchfab та gltf Editor.

Каталог меблів



Холодильник

Енергоефективне зберігання продуктів

[Детальніше](#)



Ванна

Комфорт і стиль для відпочинку

[Детальніше](#)



Комод

Практичне зберігання речей

[Детальніше](#)



Стул

Зручний для тривалого сидіння

[Детальніше](#)



Синтезатор

Ідеальний для музичного натхнення

[Детальніше](#)



Крісло

Комфорт і стиль для дому

[Детальніше](#)

[На головну](#)

Синтезатор



S M L

Показати в AR



Опис:

Відкрийте нові горизонти творчості з нашим сучасним синтезатором — ідеальним поєднанням потужності, функціональності та стильного дизайну. Легкий у використанні, він дозволить вам миттєво створювати унікальні звуки та професійні мелодії навіть без спеціальної підготовки. Завдяки широкому набору ефектів і налаштувань, цей синтезатор стане вашим незамінним партнером у студії, на сцені чи вдома. Спробуйте зараз і відчуйте свободу музичного самовираження!



Рисунок 3.2.7 - Реалізований проєкт

Висновки

Отже, у процесі виконання цієї роботи було досліджено засоби створення та імплементації технології AR. Були оглянуті основні джерела 3D-моделей, що у подальшому будуть імпортуватися у проєкт доповненої реальності. До того ж, було проведено огляд та порівняння платформ і середовищ для розробки AR-проєкту. Окрема увага приділилася вивченню технічної взаємодії користувача з середовищем AR: технології трекінгу та позиціонування, інструменти та фреймворки для інтерактивності. Також, було проведено розширення функціоналу задля швидшої роботи проєкту. У свою чергу, тестування локального прототипу показало, що AR-технології можуть успішно працювати у звичайних браузерях, а використання QR-кодів як механізму переходу до AR-візуалізації є досить зручним.

Провівши аналіз наявних інструментів та середовищ для розробки проєкту доповненої реальності, перевагу для реалізації практичної частини було надано саме тим, що найбільше відповідали тематиці та запитам кінцевого продукту. Використання технологій WebAR та A-Frame на платформі 8th Wall дозволило реалізувати масштабування і розміщення 3D-моделей меблів у реальному просторі, забезпечивши природність і правдоподібність візуалізації. А впровадження серверу qrserver.com та генерації QR-кодів як механізму переходу до AR-візуалізації показало свою зручність і простоту у тестуванні.

У подальшому цей проєкт можна покращити, розширивши каталог меблів та категорій. Також, можна було б покращити функцію кастомізації моделей у реальному часі - наприклад, зміна довжини, ширини або висоти об'єкту. До того ж, проєкт можна значно поліпшити, якщо додати можливість фотографувати сцену доповненої реальності, а також удосконалити UI/UX, запровадивши адаптивні підказки.

Список літератури

1. Jon Peddie (2023). Augmented Reality: Where We Will All Live.
<https://link.springer.com/book/10.1007/978-3-031-32581-6>
2. Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola Jr., Ivan Poupyrev (2011). "3D User Interfaces: Theory and Practice".
<https://ptgmedia.pearsoncmg.com/images/9780201758672/samplepages/0201758679.pdf>
3. Стаття про оцінку глобального ринку доповненої реальності та її останні розробки. [Електронний ресурс]
<https://www.grandviewresearch.com/industry-analysis/augmented-reality-market>
4. Стаття про потенціал та перспективи доповненої реальності в освіті. [Електронний ресурс]
<https://slejournal.springeropen.com/articles/10.1186/s40561-023-00288-0>
5. Стаття про використання WebAR. [Електронний ресурс]
<https://docs.byteplus.com/en/docs/byteplus-ar/docs-web-ar-product-overview>
6. Стаття про ринок, тенденції та аналіз доповненої реальності. [Електронний ресурс]
<https://smarttek.solutions/blog/augmented-reality-statistics/>
7. Стаття про поняття SLAM. [Електронний ресурс]
<https://www.mathworks.com/discovery/slam.html>
8. Документація 8th Wall. [Електронний ресурс]
<https://www.8thwall.com/docs/api/studio/>
9. Документація A-Frame. [Електронний ресурс]
<https://aframe.io/docs/1.7.0/introduction/models.html>
10. Браузерна платформа для розробки проєктів доповненої реальності – 8th Wall. [Електронний ресурс]
<https://www.8thwall.com/>
11. Платформа для редагування 3D-моделей. [Електронний ресурс]
<https://www.gltfeditor.com/>

12. <https://www.ikea.com/global/en/newsroom/innovation/ikea-launches-ikea-place-a-new-app-that-allows-people-to-virtually-place-furniture-in-their-home-170912/>

Додаток А

Код з платформи 8th Wall, що відповідає за створення компонентів у сцені:

```
<a-scene
  button-controller
  xrextras-gesture-detector
  xrextras-almost-there
  xrextras-loading="loadBackgroundColor: #007AFF; cameraBackgroundColor: #5AC8FA;
loadImage: #myCustomImage; loadAnimation: pulse"
  xrextras-runtime-error
  renderer="colorManagement: true;"
  xrweb>
  <a-assets>
    <!-- Завантаження моделей -->
    <a-asset-item id="fridge" src="assets/Furniture/Models/fridge.glb"></a-asset-
item>
    <a-asset-item id="chair" src="assets/Furniture/Models/chair.glb"></a-asset-
item>
    <a-asset-item id="cabinet" src="assets/Furniture/Models/cabinet.glb"></a-asset-
item>
    <a-asset-item id="bad" src="assets/Furniture/Models/bad.glb"></a-asset-item>
    <a-asset-item id="synthesizer"
src="assets/Furniture/Models/synthesizer.glb"></a-asset-item>
    <a-asset-item id="stuhl" src="assets/Furniture/Models/stuhl.glb"></a-asset-
item>
    <a-asset-item id="cube" src="assets/Furniture/Models/cube.glb"></a-asset-item>
  </a-assets>

  <!-- Камера -->
  <a-camera
    id="camera"
    position="0 0 0"
    raycaster="objects: .draggable"
    cursor="fuse: false; rayOrigin: mouse;">
  </a-camera>

  <a-entity light="type: directional; intensity: 0.8; castShadow: true;"
position="1 2 1" shadow></a-entity>

  <a-light type="ambient" intensity="0.7"></a-light>

  <!-- Відображення моделі -->
  <a-entity id="model" class="draggable" gltf-model="#cube" position="0 0 -3"
scale="1 1 1" shadow>
  </a-entity>

  <a-box
    position="0 -0.5 0"
    rotation="-90 0 0"
```

```

    height="100"
    width="100"
    material="color: #000; transparent: true; opacity: 0.5; shader: shadow"
    shadow>
  </a-box>
</a-scene>

<!-- Кнопки у 2D UI -->
<button class="ui-button ui-button-left" id="left-button">
  
</button>
<button class="ui-button ui-button-right" id="right-button">
  
</button>
<button class="ui-button ui-button-up" id="forward-button">
  
</button>
<button class="ui-button ui-button-down" id="backward-button">
  
</button>

<!-- Нові кнопки для зміни масштабу -->
<button class="ui-button ui-button-small" id="small-button">S</button>
<button class="ui-button ui-button-medium" id="medium-button">M</button>
<button class="ui-button ui-button-large" id="large-button">L</button>

<!-- Кнопка для обертю об'єкту навколо себе -->
<button class="ui-button ui-button-rotate" id="rotate-button">
  
</button>

```

Код, що відповідає за правильно роботу сцени, її компонентів та кнопок:

```

import './index.css'

AFRAME.registerComponent('button-controller', {
  init() {
    this.el.addEventListener('loaded', () => {
      const model = document.querySelector('#model')
      if (!model) {
        document.body.insertAdjacentHTML('beforeend', '<div
style="background:red;z-index:9999;position:fixed;top:60px;left:0;">NO
model</div>')
      }
      return
    })
  }
})

```

```
// Отримуємо параметр model із URL
const urlParams = new URLSearchParams(window.location.search)
const modelName = urlParams.get('model')

// Встановлюємо модель меблів в залежності від значення параметру
if (modelName) {
  switch (modelName) {
    case 'fridge':
      {
        const oldModel = document.querySelector('#model')
        if (oldModel) oldModel.parentNode.removeChild(oldModel)
        const newModel = document.createElement('a-entity')
        newModel.setAttribute('id', 'model')
        newModel.setAttribute('class', 'draggable')
        newModel.setAttribute('gltf-model', '#fridge')
        newModel.setAttribute('position', '0 0 -3')
        newModel.setAttribute('scale', '1 1 1')
        newModel.setAttribute('shadow', '')
        document.querySelector('a-scene').appendChild(newModel)
        attachModelHandlers(newModel)
      }
      break
    case 'bad':
      {
        const oldModel = document.querySelector('#model')
        if (oldModel) oldModel.parentNode.removeChild(oldModel)
        const newModel = document.createElement('a-entity')
        newModel.setAttribute('id', 'model')
        newModel.setAttribute('class', 'draggable')
        newModel.setAttribute('gltf-model', '#bad')
        newModel.setAttribute('position', '0 0 -3')
        newModel.setAttribute('scale', '1 1 1')
        newModel.setAttribute('shadow', '')
        document.querySelector('a-scene').appendChild(newModel)
        attachModelHandlers(newModel)
      }
      break
    case 'cabinet':
      {
        const oldModel = document.querySelector('#model')
        if (oldModel) oldModel.parentNode.removeChild(oldModel)
        const newModel = document.createElement('a-entity')
        newModel.setAttribute('id', 'model')
        newModel.setAttribute('class', 'draggable')
        newModel.setAttribute('gltf-model', '#cabinet')
        newModel.setAttribute('position', '0 0 -3')
        newModel.setAttribute('scale', '1 1 1')
        newModel.setAttribute('shadow', '')
        document.querySelector('a-scene').appendChild(newModel)
      }
      break
  }
}
```

```
    attachModelHandlers(newModel)
  }
  break
case 'stuhl':
  {
    const oldModel = document.querySelector('#model')
    if (oldModel) oldModel.parentNode.removeChild(oldModel)
    const newModel = document.createElement('a-entity')
    newModel.setAttribute('id', 'model')
    newModel.setAttribute('class', 'draggable')
    newModel.setAttribute('gltf-model', '#stuhl')
    newModel.setAttribute('position', '0 0 -3')
    newModel.setAttribute('scale', '1 1 1')
    newModel.setAttribute('shadow', '')
    document.querySelector('a-scene').appendChild(newModel)
    attachModelHandlers(newModel)
  }
  break
case 'synthesizer':
  {
    const oldModel = document.querySelector('#model')
    if (oldModel) oldModel.parentNode.removeChild(oldModel)
    const newModel = document.createElement('a-entity')
    newModel.setAttribute('id', 'model')
    newModel.setAttribute('class', 'draggable')
    newModel.setAttribute('gltf-model', '#synthesizer')
    newModel.setAttribute('position', '0 0 -3')
    newModel.setAttribute('scale', '1 1 1')
    newModel.setAttribute('shadow', '')
    document.querySelector('a-scene').appendChild(newModel)
    attachModelHandlers(newModel)
  }
  break
case 'chair':
  {
    const oldModel = document.querySelector('#model')
    if (oldModel) oldModel.parentNode.removeChild(oldModel)
    const newModel = document.createElement('a-entity')
    newModel.setAttribute('id', 'model')
    newModel.setAttribute('class', 'draggable')
    newModel.setAttribute('gltf-model', '#chair')
    newModel.setAttribute('position', '0 0 -3')
    newModel.setAttribute('scale', '1 1 1')
    newModel.setAttribute('shadow', '')
    document.querySelector('a-scene').appendChild(newModel)
    attachModelHandlers(newModel)
  }
  break
default:
  {
```

```

        const oldModel = document.querySelector('#model')
        if (oldModel) oldModel.parentNode.removeChild(oldModel)
        const newModel = document.createElement('a-entity')
        newModel.setAttribute('id', 'model')
        newModel.setAttribute('class', 'draggable')
        newModel.setAttribute('gltf-model', '#cube')
        newModel.setAttribute('position', '0 0 -3')
        newModel.setAttribute('scale', '1 1 1')
        newModel.setAttribute('shadow', '')
        document.querySelector('a-scene').appendChild(newModel)
        attachModelHandlers(newModel)
    }
    break
}
} else {
    const oldModel = document.querySelector('#model')
    if (oldModel) oldModel.parentNode.removeChild(oldModel)
    const newModel = document.createElement('a-entity')
    newModel.setAttribute('id', 'model')
    newModel.setAttribute('class', 'draggable')
    newModel.setAttribute('gltf-model', '#cube')
    newModel.setAttribute('position', '0 0 -3')
    newModel.setAttribute('scale', '1 1 1')
    newModel.setAttribute('shadow', '')
    document.querySelector('a-scene').appendChild(newModel)
    attachModelHandlers(newModel)
}
})
},
})

function attachModelHandlers(model) {
    document.querySelector('#forward-button')?.addEventListener('click', () => {
        const position = model.getAttribute('position')
        model.setAttribute('position', {x: position.x, y: position.y, z: position.z +
0.5})
    })
    document.querySelector('#backward-button')?.addEventListener('click', () => {
        const position = model.getAttribute('position')
        model.setAttribute('position', {x: position.x, y: position.y, z: position.z -
0.5})
    })
    document.querySelector('#left-button')?.addEventListener('click', () => {
        const position = model.getAttribute('position')
        model.setAttribute('position', {x: position.x - 0.5, y: position.y, z:
position.z})
    })
    document.querySelector('#right-button')?.addEventListener('click', () => {
        const position = model.getAttribute('position')

```

```

    model.setAttribute('position', {x: position.x + 0.5, y: position.y, z:
position.z})
  })
  document.querySelector('#small-button')?.addEventListener('click', () => {
    model.setAttribute('scale', {x: 1, y: 1, z: 1})
  })
  document.querySelector('#medium-button')?.addEventListener('click', () => {
    model.setAttribute('scale', {x: 1.2, y: 1.2, z: 1.2})
  })
  document.querySelector('#large-button')?.addEventListener('click', () => {
    model.setAttribute('scale', {x: 1.3, y: 1.3, z: 1.3})
  })
  let rotateInterval = null

  const rotateButton = document.querySelector('#rotate-button')

  rotateButton?.addEventListener('click', () => {
    if (rotateInterval) {
      // якщо вже обертається - зупинити
      clearInterval(rotateInterval)
      rotateInterval = null
    } else {
      // якщо не обертається - запустити
      rotateInterval = setInterval(() => {
        const rotation = model.getAttribute('rotation')
        model.setAttribute('rotation', {x: rotation.x, y: rotation.y + 1, z:
rotation.z})
      }, 20)
    }
  })
}

```

Код, що відповідає за відображення деталей продукту та генерацію необхідного QR-коду для перегляду моделі у доповненій реальності:

```

// Функція для отримання та відтворення деталей продукту з файлу JSON
async function renderItemDetails() {
  try {
    // Отримання даних з furniture.json
    const response = await fetch('/data/furniture.json');
    const furnitureData = await response.json();

    const itemId = window.location.pathname.split('/').pop(); // Витягаємо
ідентифікатор елемента з URL
    const item = furnitureData.find(f => f.id === itemId);

    // Перевіряємо, чи існує елемент, і відображаємо його деталі
    if (item) {
      document.getElementById('itemName').innerText = item.name;
      document.getElementById('itemImage').src = `/images/${item.image}`;
      document.getElementById('itemDescription').innerText =
item.description;
    }
  }
}

```

```
        //Використання arUrl з furniture.json для створення правильного QR-  
коду  
        const qrCodeLink =  
`https://yelyzavetapetrenko.8thwall.app/camera/?model=${item.arUrl}`;  
        console.log('qrCodeLink:', qrCodeLink);  
  
        document.getElementById('qrCodeImage').src =  
`https://api.qrserver.com/v1/create-qr-  
code/?data=${encodeURIComponent(qrCodeLink)}&size=200x200`;  
        } else {  
            document.getElementById('itemName').innerText = 'Product not found';  
            document.getElementById('itemDescription').innerText = '';  
        }  
    } catch (error) {  
        console.error('Error loading furniture data:', error);  
    }  
}  
  
window.onload = renderItemDetails;
```