

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ**

**КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ**

Кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему Навігація з використанням технологій доповненої реальності та
GPS

Виконав: студент 2 курсу, групи 8.1210-2іпз
спеціальності 121 Інженерія програмного
забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного
забезпечення

(код і назва освітньої програми)

С. О. Воропаєв

(ініціали та прізвище)

Керівник Н.П. Полякова
доцент, канд.техн.наук, Н.П. Полякова

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

Рецензент директор ТОВ «Альтер Віжен Груп»

В.С. Тряпичко

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

Запоріжжя
2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ

Кафедра _____ програмного забезпечення автоматизованих систем
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 121 Інженерія програмного забезпечення _____
(код та назва)
Освітня програма _____ Інженерія програмного забезпечення _____
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ *Вербицький* В.Г. Вербицький
“ 01 ” вересня _____ 2021 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Воропаєву Сергію Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Навігація з використанням технологій доповненої реальності та GPS

керівник роботи Полякова Наталія Петрівна, доцент, канд.техн.наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від “30” червня 2021 року № 974-с

2. Строк подання студентом кваліфікаційної роботи 30.11.2021

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми навігації з використанням технологій доповненої реальності та GPS;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
15 слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.09.2021**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-10.09.21	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.09-12.09.2021	виконано
3	Аналіз існуючих методів рішення	13.09-17.09.21	виконано
4	Дослідження області доповненої реальності	18.09-24.09.21	виконано
5	Узгодження подальших дій з науковим керівником	25.09-26.09.21	виконано
6	Аналіз теоретичних відомостей	27.09-15.10.21	виконано
7	Проектування прототипу застосунку	15.10-23.10.21	виконано
8	Узгодження застосунку з науковим керівником	23.10-24.10.21	виконано
9	Реалізація функціоналу застосунку	25.10-14.11.21	виконано
10	Представлення отриманих результатів науковому керівнику і узгодження плану подальшого дослідження	15.11-16.11.21	виконано
11	Реалізація функціоналу гри	16.11-23.11.21	виконано
12	Проведення аналізу можливостей розробленого застосунку	24.11-26.11.21	виконано
13	Оформлення звіту	27.11-30.11.21	виконано

Студент  С.О. Воропаєв
(підпис) (прізвище та ініціали)Керівник роботи  Н.П. Полякова
(підпис) (прізвище та ініціали)**Нормоконтроль пройдено**Нормоконтролер  І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок: 90

Рисунків: 39

Таблиць: 7

Джерел: 32

Воропаєв С.О Навігація з використанням технологій доповненої реальності та GPS.

Кваліфікаційна робота для здобуття ступеня вищої освіти магістра за спеціальністю 121 — Інженерія програмного забезпечення, науковий керівник Н.П. Полякова. Інженерний навчально-науковий інститут Запорізького національного університету, 2021.

Мета кваліфікаційної роботи полягає у вивченні методів побудови системи доповненої реальності, аналізу і порівнянні існуючих засобів забезпечення стійкості віртуального досвіду між сесіями, а також доведення ефективності використання доповненої реальності в навігації на прикладі створеного застосунок. Застосунок використовує данні геолокації для прокладання маршруту на вулиці, та данні збереженої сесії для маршрутів в приміщенні.

Досліджено методи і засоби для створення систем доповненої реальності на мобільних пристроях, а також доведено доречність поєднання технологій доповненої реальності та GPS. Спроектовано та реалізовано застосунок для мобільних пристроїв під управлінням операційної системи iOS за допомогою ігрового движку Unity3D та бібліотеки AR Foundation.

Ключові слова: *доповнена реальність, GPS, навігація, ARFoundation, ARKit, локалізація і картографування, стійкий AR.*

SUMMARY

Pages: 90

Figures: 39

Tables: 7

Sources: 32

Voropaiev S.O. Navigation with use of augmented reality and GPS.

Qualification work for higher master's degree in specialty 121 — Software Engineering, supervisor N.P. Poliakova. Engineering, educational and scientific institute of ZNU. 2021.

The aim of the qualification work is to study the methods of creation of augmented reality systems, analysis and comparison of existing means to keep persistence of virtual experience between sessions, and proving effectiveness of augmented reality use in navigation by creating a navigation application.

Methods and tools for augmented reality systems creation on mobile devices are studied and relevance of the combination of augmented reality and GPS technologies is proved. Designed and implemented and application for mobile devices, running on iOS operation system, using the Unity3D game engine and the AR Foundation library.

Keywords: augmented reality, GPS, navigation, ARFoundation, ARKit, localization and mapping, persistent AR.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПОБУДОВИ СИСТЕМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ	15
1.1 Загальні відомості про технологію доповненої реальності	15
1.2 Класифікація типів доповненої реальності	17
1.2.1 Доповнена реальність на основі проєкції	17
1.2.2 Доповнена реальність на основі маркерів	17
1.2.3 Доповнена реальність на безмаркерній основі	19
1.2.4 Доповнена реальність на основі накладання.....	20
1.3 Прикладне використання доповненої реальності	21
1.3.1 Використання доповненої реальності в медицині	21
1.3.2 Використання доповненої реальності в сфері розваг та ігор	22
1.3.3 Використання доповненої реальності в сфері освіти	23
1.4 Проблема навігації з використанням доповненої реальності.....	24
1.4.1 Навігація з доповненою реальністю для автомобілей.....	25
1.4.2 Навігація з доповненою реальністю для пішоходів	26
1.4.3 Навігація з доповненою реальністю в приміщенні.....	27
1.5 Аналіз існуючих систем доповненої реальності.....	28
1.5.1 Google Maps	28
1.5.2 Apple Maps	29
1.5.3 Результати аналізу існуючих систем	31
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАСОБІВ ПОБУДОВИ СТІЙКОЇ СИСТЕМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ З ВИКОРИСТАННЯМ GPS	32
2.1 Методи комп'ютерного зору в технології доповненої реальності.....	32
2.1.1 Накопичення помилки	33
2.1.2 Представлення відносності	35
2.1.3 Приближення максимальної імовірності.....	36
2.1.4 Розширений фільтр Калмана	37

2.1.5 Порівняння коваріаційної та інформаційної матриць.....	38
2.1.6 Роль SLAM в стійкій доповненій реальності.....	40
2.2 Методи збереження інформації між сесіями застосунку.....	40
2.2.1 Стійка доповнена реальність в ARCore.....	41
2.2.2 Стійка доповнена реальність в ARKit.....	42
2.2.3 Порівняння стійкої доповненої реальності в ARCore та ARKit.....	43
2.3 Методи перетворення GPS координат в системі доповненої реальності.....	44
РОЗДІЛ 3 ПРОЕКТ ПРОГРАМНОЇ СИСТЕМИ НАВІГАЦІЇ	3
ВИКОРИСТАННЯМ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА GPS	48
3.1 Архітектура системи.....	48
3.2 Засоби реалізації.....	50
3.2.1 Ігровий движок Unity.....	51
3.2.2 Середовище розробки Visual Studio.....	53
3.2.3 Мова програмування C# та фреймворк .NET.....	53
3.2.4 AR Foundation.....	54
3.2.5 AR Kit.....	55
3.2.6 Xcode.....	57
3.3 Система клієнт-сервер.....	58
3.4 Модулі та алгоритми.....	58
3.4.1 Модуль створення шляху навігації.....	58
3.4.2 Модуль відображення GPS шляху.....	64
3.4.3 Модуль відображення шляху доповненої реальності.....	67
3.4.4 Модуль взаємодії з віддаленим сервером.....	69
3.5 Проект інтерфейсу та візуалізації маршруту.....	72
3.5 Вимоги до апаратного забезпечення.....	76
3.6 Опис функціональних можливостей.....	77
РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ	78
НАВІГАЦІЇ	78

4.1 Аналіз швидкості та точності побудови маршрутів навігації з використанням GPS даних	78
4.2 Аналіз швидкості та точності побудови маршрутів з використанням доповненої реальності	79
4.3 Аналіз швидкості та точності відновлення маршрутів в стійкій доповненій реальності	81
4.4 Порівняння якісних характеристик GPS та AR маршрутів	82
ВИСНОВКИ	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87

ВСТУП

Актуальність теми

З технологічним прогресом, зв'язок повсякденного життя людини з віртуальним світом стає сильнішим кожного дня. Соціальні мережі, необмежений доступ до інформації та віртуальні розваги вже стали невід'ємною частиною життя багатьох людей. Але вся ця інформація залишається на моніторах наших девайсів. Саме технологія доповненої реальності (augmented reality, AR) надає можливість відобразити віртуальний контент в фізичному світі.

В наш час одним з найбільш перспективних напрямків ІТ-розробок є саме доповнена реальність. Данна технологія — це відносно новий спосіб представлення інформації. Доповнена реальність відкриває новий рівень сприйняття інформації людиною, роблячи взаємодію з даними програми набагато наочніше та простіше. Наочність, в свою чергу, досягається шляхом накладання віртуальних об'єктів на існуючі в режимі реального часу.

Сучасний споживач розпещений різноманітністю форматів реклами і послуг, тому його увагу все складніше привернути до конкретного продукту. Згідно зі статистикою, що проводиться німецькою компанією Statista, близько 75% рекламних компаній не окупаються і спрямовані на підтримку так званого «рекламного фону», який не запам'ятовується цільовою аудиторією і не сприяє збільшенню продажів. Дана ситуація змушує шукати нові шляхи просування і рекламування різних продуктів, в тому числі і ґрунтуючись на сучасних технологіях.

За останні роки AR стала інноваційним і ефективним інструментом маркетингу, що змінює саму суть споживання. Використовуючи AR, ми можемо не тільки відображати інформацію про продукт, але і залучати споживачів до інтерактивної гри, тим самим збільшуючи інтерес. Розробки доповненої реальності в рекламі використовуються у всіх сферах. Функціями доповненої реальності забезпечуються вітрини, примірочні, інтерактивні кіоски, промо-

стенди. Рекламні кампанії, проведені з використанням технологій доповненої реальності, формують стійке позитивне враження від товару або послуги. Виходячи з результатів ще одного дослідження проведеного компанією Statista, прибуток від AR-додатків в 2019 році склав приблизно 3.3 млрд. доларів. Передбачений прибуток на 2022 рік складає 15.5 млрд. доларів. П'ятикратний приріст протягом трьох років дозволяє впевнено сказати, що ринок використання AR технологій тільки починає свій ріст. Таких даних статистики вдалося досягти завдяки одній з головних відмінностей застосунків з використанням AR — можливість не просто уявити споживачеві товар або послугу, а й відразу залучити його до процесу їх використання, тим самим досягаючи «вау-ефекту».

Все більше провідних брендів використовують доповнену реальність, щоб залучити аудиторію споживачів. Доповнена реальність дозволяє поглянути на звичні всім речі по-новому, а також пропонує нові продукти брендів в новому світлі, як би «оживляючи» їх. Це дає можливість взаємодіяти зі споживачами або ж потенційними покупцями зовсім інакше. Такий вид рекламного продукту може стати цікавим для аудиторії за рахунок його інтерактивності. Уявіть собі, що ви заходите в продуктової магазин, і потрібні вам товари підсвічуються для вас на полицях (це бачите тільки ви). Або що ви можете, сидячи вдома, побачити, як будуть виглядати у вас нові меблі з ІКЕА. Або приміряти черевики ще до походу в магазин. Або вивчити курорт, куди ви збираєтеся відправитися під час відпустки, лежачи у ванній. Можливості безмежні. Інтерактив сам по собі привабливіше, ніж статичні об'єкти, тому реклама або ж інші продукти з використанням технологій доповненої реальності мають всі шанси стати популярними і успішними.

Але популярність доповненої реальності не обмежується сферою маркетингу, але також має своє використання в навігації. Огляд споживчих технологій роздрібної торгівлі в 2019 році показав, що 61 відсоток покупців шукають спосіб зменшення часу проведеного в магазинах. Саме AR навігація в

приміщені дозволяє користувачам знайти потрібний їм товар за лічені секунди, замість того, щоб витратити цінний час на пошук консультантів.

Приймаючи в розрахунок кількість існуючих, та швидкість створення нових торгівельних центрів, зберегти в пам'яті місцезнаходження усіх магазинів стає майже неможливо. Але з використання переваг сучасних технологій, таких як AR та GPS, знайти детальний шлях до цікавого вам місця не викликає проблем.

Тож актуальність даної теми полягає в тому, що програмний продукт який використовує інтерактивність технології доповненої реальності і вирішує проблему пошуку шляху, може значно вплинути на зацікавленість відвідувачів магазинів.

Мета і завдання дослідження

Мета дослідження полягає в аналізі методів створення доповненої реальності та порівнянні засобів, що дозволяють зберігати та відтворювати положення віртуальних об'єктів між сесіями застосунку, а також створення мобільного застосунку для навігації, як підтвердження ефективності об'єднання технологій доповненої реальності та GPS.

Об'єкт дослідження

Об'єктом дослідження є побудований шлях навігації в просторі доповненої реальності.

Предмет дослідження

Предметом дослідження є процес побудови шляхів навігації, що базуються на мапі навколишнього середовища, чи на збережених геоданих, а також їх сумісне використання.

Методи дослідження

Для виконання представлених завдань та цілей використовуються такі методи дослідження:

— Аналіз методів одночасної локалізації і картографування навколишнього середовища.

- Порівняння методів збереження мапи навколишнього середовища між сесіями застосунку.
- Аналіз алгоритмів проекції геоданих в просторі доповненої реальності.

Наукова новизна одержаних результатів

Одержані результати демонструють практичність використання передових методів доповненої реальності в повсякденному житті людей, а саме для наглядної візуалізації шляхів маршруту або точок інтересу в невідомій користувачу місцевості. Проаналізувавши отримані дані можливе об'єднання доповненої реальності з іншими технологіями, що оперують лише цифровими значеннями.

Практичне значення отриманих результатів

Все більше провідних компаній використовують доповнену реальність, щоб залучити аудиторію споживачів. Доповнена реальність дозволяє поглянути на звичні всім речі по-новому. Це дає можливість взаємодіяти зі споживачами або ж потенційними покупцями зовсім інакше. Інтерактив сам по собі привабливіше, ніж статичні об'єкти, тому навігація, реклама або ж інші продукти з використанням технологій доповненої реальності мають всі шанси стати популярними і успішними.

Апробація результатів

Результати дослідження були представлені на XIII науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених Запорізького національного університету «Молода наука-2021» [31], а також на XXV науково-технічній конференції студентів, аспірантів, магістрантів і викладачів Інженерного навчально-наукового інституту Запорізького національного університету [32].

Глосарій

AR Foundation — кросплатформний фреймворк для Unity, що дозволяє створювати застосунки з використанням технології доповненої реальності для смартфонів під контролем операційних систем Android та iOS.

Віртуальна реальність (англ. Virtual reality) — різновид реальності в формі тотожності матеріального й ідеального, що створюється та існує завдяки іншій реальності. У вужчому розумінні — ілюзія дійсності, створювана за допомогою комп'ютерних систем, які забезпечують зорові, звукові та інші відчуття.

Бібліотека (від англ. Library) — збірка об'єктів чи підпрограм для вирішення близьких за тематикою задач. У залежності від мови програмування бібліотеки містять об'єктні модулі чи сирцевий код та дані, допоміжні для задіяння та інтеграції нових можливостей в програмні рішення.

Доповнена реальність (англ. Augmented reality, AR) — термін, що позначає всі проекти, спрямовані на доповнення реальності будь-якими віртуальними елементами. Доповнена реальність — складова частина змішаної реальності (англ. Mixed reality), в яку також входить «доповнена віртуальність» (коли реальні об'єкти інтегруються у віртуальне середовище).

Комп'ютерний зір — теорія та технологія створення машин, які можуть проводити виявлення, стеження та класифікацію об'єктів.

Кросплатформність — характеристика фреймворку чи бібліотеки, що дозволяє створювати застосунки для декількох операційних систем, використовуючи один набір інструментів.

Одометрія — це спосіб оцінки руху за допомоги даних, отриманих із сенсорів руху.

Поза — структура, що описує положення і обертання об'єкту в просторі.

Сериалізація — процес переводу структури даних до послідовності байтів.

Система глобального позиціювання (з англ. Global positioning system, GPS) — супутникова система навігації, що дозволяє виміряти відстань, час і місцеположення на поверхні Землі майже за будь яких умов. Положення об'єкта обчислюється завдяки використанню розміщеному на ньому GPS-приймача.

Фреймворк (англ. Framework, каркас, платформа, структура, інфраструктура) — інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проєкту та написання коду.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПОБУДОВИ СИСТЕМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

1.1 Загальні відомості про технологію доповненої реальності

Доповнена реальність (від *англ. Augmented Reality — AR*) — це інтерактивний досвід з реальним навколишнім середовищем, де фізичні об'єкти доповнені перцептивною інформацією, згенерованою комп'ютером [1]. Доповнена реальність є варіацією віртуальної реальності, але замість повної заміни навколишнього середовища виконує функцію доповнення [2].

Мілгарм П. запропонував, так званий, *континуум реальність-віртуальність* один з кінців якого є повністю реальний світ, а інший — повністю віртуальний [3]. Візуальне представлення концепту зображено на рисунку 1. В рамках цього континууму змішана реальність може бути визначена як поєднання фізичних та згенерованих об'єктів. Окрім цього, змішана реальність ґрунтується на двох основних ідеях: доповнена реальність та доповнена віртуальність. Доповнена реальність є комбінацією, в якій переважають реальні об'єкти, в той час як доповнена віртуальність складається з переважно віртуальних.

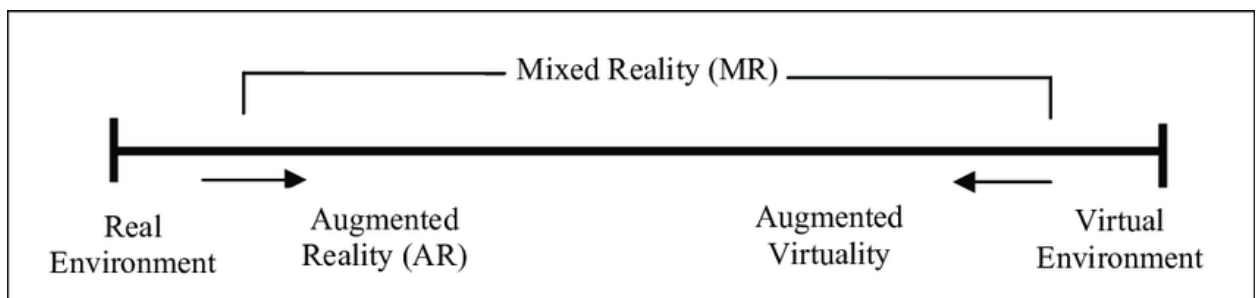


Рис. 1 Континуум реальність-віртуальність

Доповнена реальність націлена на полегшення життя користувача за допомогою відображення контенту на фізичних речах, тим самим покращуючи перцептивне сприйняття інформації.

Азума Р. також описав можливість використання доповненої реальності для видалення реальних об'єктів з навколишнього середовища [4]. Така методика отримала назву «зменшеної» (від англ. *Diminished*) реальності. «Стирання» досягається шляхом накладання віртуального контенту, що співпадає з заднім фоном, на об'єкт.

Перша згадка про доповнену реальність датується 1950 роком, коли Мортон Хейліг, кінематографіст, вирішив зробити кіно інтерактивним, надаючи можливість глядачам відчувати стрічку усіма органами чуття. В 1962 році Хейліг побудував прототип його ідеї, який мав назву «Sensorama» [5]. В 1968 Айвен Сазерланд створив першу систему доповненої реальності, яка використовує прозорий оптичний дисплей, що одягається на голову [6], дисплей зображено на рисунку 2. В 1975 році виходить «KARMA» — перша наукова дослідницька робота на тему прототипу системи доповненої реальності. В 1997 році Рональд Азума написав перший огляд доповненої реальності, що і дав їй визначення, яким ми користуємось і в сучасності. Перша гра з доповненою реальністю була розроблена Брюсом Томасом в 2000 році. В 2005 році видавництво «Horizon Reports» передбачило, що технологія доповненої реальності займе нові висоти на протязі наступних 4-5 років [7]. І справді, з розвитком мобільних пристроїв та з появою смартфонів, в наш час майже кожен має в кишені пристрій, що дозволяє змінити реальність.

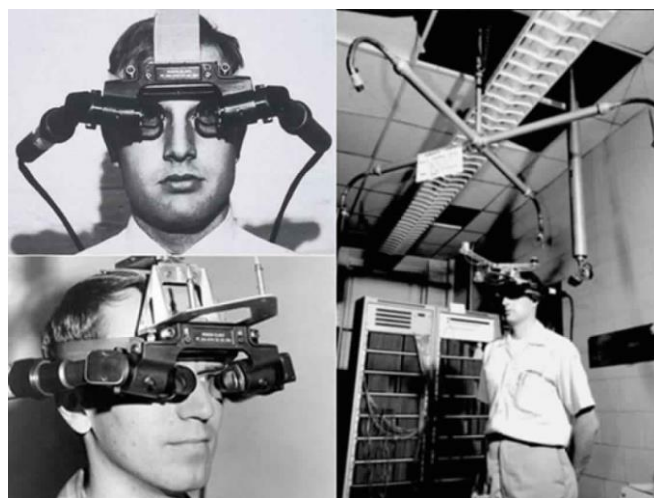


Рис. 2 Дисплей Айвена Сазерланда

1.2 Класифікація типів доповненої реальності

Доповнена реальність поділяється на 5 типів, кожен має свої переваги, недоліки та сфери використання.

1.2.1 Доповнена реальність на основі проекції

Доповнена реальність на основі проекції, виходячи з назви, використовує проекцію для відображення цифрових зображень на поверхню або об'єкт [8]. Прикладом проекційної доповненої реальності є віртуальна клавіатура, що проектується на офісний стіл. Такий тип доповненої реальності прибирає необхідність в дисплеї чи гарнітурі, але є досить обмеженим в інтерактивності. Приклад проекційної реальності зображено на рисунку 3.

Головною перевагою такого типу доповненої реальності є можливість прийняти участь в аугментації великій кількості людей одночасно, в той час як інші види доповненої реальності переважно спроектовані для одного користувача.

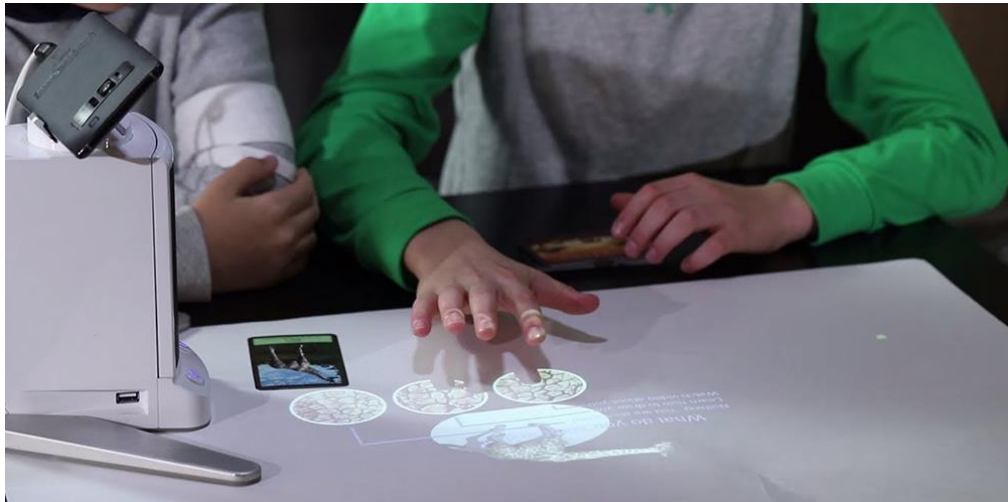


Рис. 3 Приклад проекційної доповненої реальності

1.2.2 Доповнена реальність на основі маркерів

Доповнена реальність на основі маркерів базується на алгоритмах розпізнавання зображень — маркерів [9]. Маркер використовуються як точка

розташування віртуального контенту. Переважно, маркери мають чорно-білу структуру, але використання кольорів також можливе якщо контраст між ними буде достатнім для розпізнавання камерою пристрою. Найпростіші маркери доповненої реальності можуть складатися з однієї чи декількох базових геометричних фігур чорного кольору, що розташовані на білову фоні. Приклад такого маркеру наведений на рисунку 4. Більш складні маркери можуть приймати вигляд довільного малюнку.

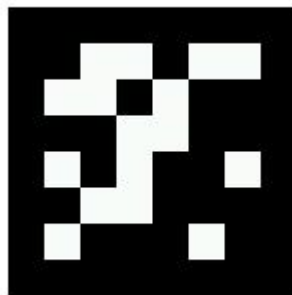


Рис. 4 Маркер доповненої реальності

Основною перевагою доповненої реальності на основі маркерів є найбільш надійна та точна візуалізація і відстеження віртуального контенту. Окрім цього, для налаштування такої системи користувачу потрібно докласти мінімальні зусилля — навести камеру на роздрукований малюнок.

Серед недоліків такого підходу слід зазначити обмеженість масштабування інтерактивного досвіду. Приклад використання маркерної доповненої реальності зображено на рисунку 5.



Рис. 5 Доповнена реальність з використанням маркерів

1.2.3 Доповнена реальність на безмаркерній основі

Доповнена реальність на безмаркерній основі є наймолодшим типом доповненої реальності. З розвитком технологій розробники почали шукати спосіб обійти обмеження в масштабуванні фізичних маркерів, що і привело до створення нового класу аугментації.

Перші версії безмаркерної доповненої реальності спиралися на дані з акселерометру, GPS та цифрового компасу для визначення положення та орієнтації пристрою в просторі. Але таке рішення страждало від неточності в усіх визначених параметрах, тому у використання було прийнято систему аналізу зображень **SLAM** (simultaneous localization and mapping) — технологія одночасної локалізації та мапування [10].

SLAM базується на скануванні навколишнього середовища та створенні мап, що використовуються для розташування цифрового контенту. Завдяки цьому віртуальні об'єкти, що були розташовані в реальному світі, залишаються на своїх місцях, навіть коли вони зникають з поля зору користувача при переміщенні.

Безмаркерна доповнена реальність може бути використана майже на будь-якій невідомій місцевості. Будування мапи та підтримання цілісності віртуального контенту в реальному часі виводить доповнену реальність на

новий рівень імерсивності, що і робить цей тип аугментації самим популярним.

Але безмаркерна доповнена реальність, як і будь-яка динамічна система, має свої недоліки. Головною проблемою динамічної побудови мапи навколишнього середовища є погане освітлення та рефлексивні поверхні. При недостатньому освітленні кількість характеристичних точок для побудови мапи значно зменшується, а рефлексивні поверхні змінюють зовнішній вигляд в залежності від куту спостереження, що приводить до нестабільності характеристичних точок та порушення цілісності мапи. Приклад використання безмаркерної доповненої реальності приведено на рисунку 6.

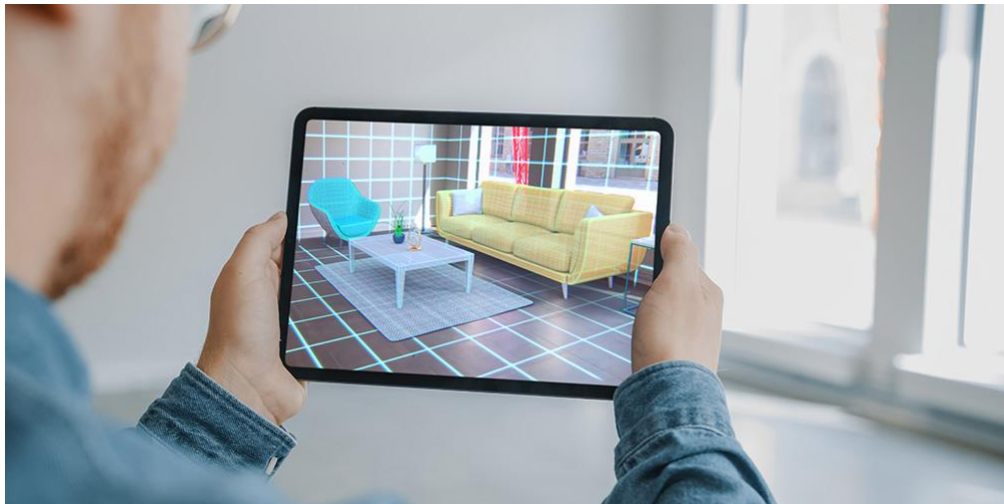


Рис. 6 Приміряння меблів в кімнаті за допомоги безмаркерної доповненої реальності.

1.2.4 Доповнена реальність на основі накладання

Доповнена реальність на основі накладання базується на методах комп'ютерного бачення для розпізнання об'єктів, та накладання на них цифрового зображення [11]. Цей тип аугментації переважно використовується в медицині, для накладання рентгенівського знімку на тіло людини, як зображено на рисунку 7.

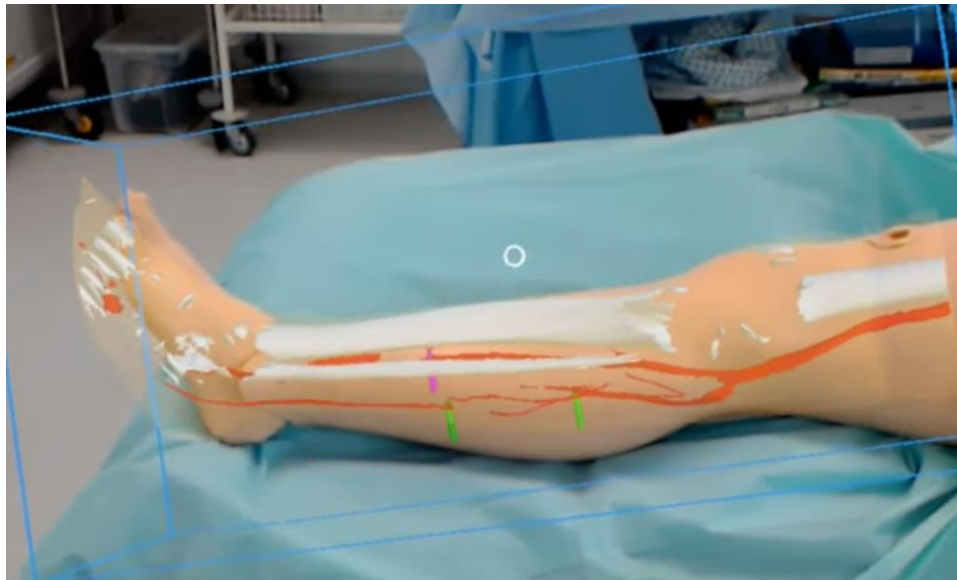


Рис. 7 Застосування накладної доповненої реальності в медицині

1.3 Прикладне використання доповненої реальності

Доповнена реальність дозволяє взаємодіяти з реальним та віртуальним світом одночасно. Це є прикладом механічно доповненого інтелекту, що означає використання комп'ютеру як інструменту для спрощення задач людини [12]. Саме тому ця технологія знайшла своє застосування в багатьох сферах людського життя.

1.3.1 Використання доповненої реальності в медицині

Доповнена реальність широко використовується в сфері охорони здоров'я із-за потреби в візуалізації медичної інформації на пацієнта не рухаючи його [12]. Доповнена реальність може бути використана під час хірургічних операцій або для проведення операцій дистанційно, без фізичної наявності хірурга біля пацієнта [13]. Приклади застосунків, що використовуються в медицині:

- EyeDecide — медичний застосунок, що використовує камеру для симуляції впливу захворювань та медикаментів на зір людини.

— AccuVein — застосунок, що використовує ручну камеру для проєкції вен на шкіру людини. Використання застосунку зображено на рисунку 8.



Рис 8 Медичний застосунок AccuVein

1.3.2 Використання доповненої реальності в сфері розваг та ігор

Доповнену реальність можна вважати революціонером індустрії розваг та ігор, розширюючи горизонти візуалізації та інтерактивності контенту. Доповнена реальність може, навіть, бути використана на телебаченні. Спортивні телеканали використовують накладання контенту [14], як зображено на рисунку 9.

Ігри також в повній мірі використовують технологію аугментації. Наприклад, популярний на цей час «Pokémon Go» дозволяє геймерам пограти зі своїми улюбленими персонажами з мультфільму.



Рис. 9 Доповнена реальність на телебаченні

1.3.3 Використання доповненої реальності в сфері освіти

Доповнена реальність в сфері освіти, мабуть, найбільш вдале використання технології в повсякденному житті людини. Доповнена реальність допомагає учням шкільного віку візуалізувати складні просторові зв'язки та абстрактні концепти [12]. Ця технологія дозволяє учням приймати участь у феноменах, які не можливі в реальному світі. Наприклад, такі концепти як магнітні поля, що не помітні оку, можуть бути візуалізовані доповненою реальністю. Приклади застосунків з доповненою реальністю для навчання:

- AugThat — цей застосунок допомагає покращити процес навчання [15] за допомоги 3D уроків.
- Elements 4D — цей застосунок націлений на вивчення хімії. Він дозволяє учням побачити як різні елементи взаємодіють між собою [15]. Використання застосунку продемонстровано на рисунку 10.
- Zookazam — це застосунок, що зображує тварин у 3D моделях, для їх подальшого вивчення [15].



Рис. 10 *Elements 4D* — застосунок для вивчення хімії

1.4 Проблема навігації з використанням доповненої реальності

В наш час самими популярними системами навігації на ринку, є застосунки для смартфонів або незалежні системи, що використовують домашню супутникову систему. Такі системи навігації, зазвичай, відображають карту з накладеним навігаційним методом та інформацією про положення користувача. Окрім цього, для дотримання заданого шляху застосунок надає інструкції у вигляді візуальних або аудіо підказок. Хоча існуючі навігаційні застосунки і вирішують проблему навігації, але вони не відповідають усім інтересам користувача. Типова навігаційна система надає інформацію в обмеженій формі, що зазвичай представлена у вигляді стрілок, які вказують напрямок руху, або зображенням карти та маршруту з висоти пташиного польоту.

Навігаційні системи для авто представлені дисплеєм, що розташований на панелі приладів. Така система потребує зміни уваги водія з дороги на дисплей, що може призвести до аварійної ситуації. Цього відволікання можна уникнути якщо ввести технологію доповненої реальності в процедуру навігації.

Як правило навігація з використанням доповненої реальності працює за таким алгоритмом [16]:

1. Отримати зображення навколишнього середовища користувача.
2. Отримати дані про локацію користувача. Зазвичай ця інформація надається системою GPS.
3. Згенерувати візуальну інформацію на основі навколишнього середовища та географічних даних.
4. Зв'язати віртуальний контент з геоданими, та побудувати таким чином доповнену реальність.

Візуальне представлення схеми зображено на рисунку 11.

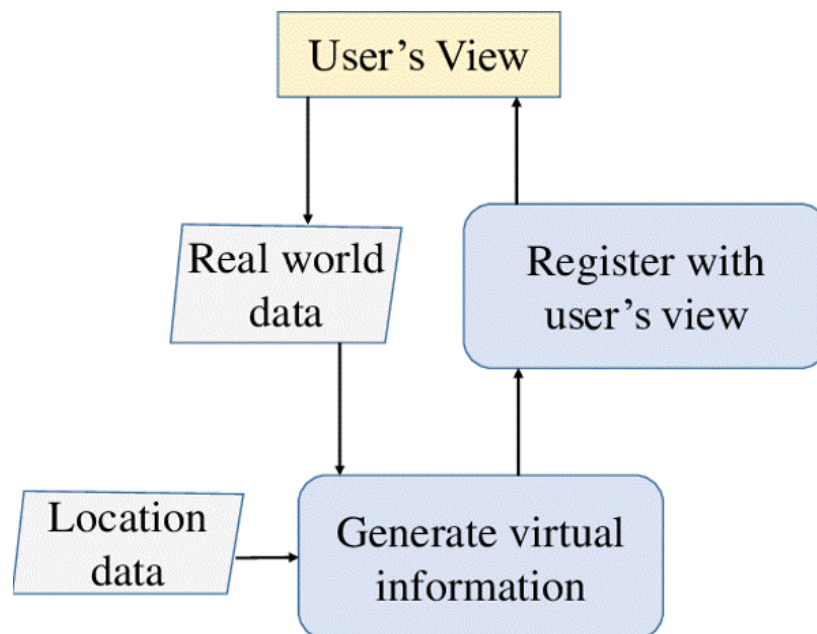


Рис. 11 Блок-схема навігації з використанням доповненої реальності

1.4.1 Навігація з доповненою реальністю для автомобілей

Головною причиною використовувати доповнену реальність в автонавігації є видалення потреби водію змінювати увагу з дороги для отримання інформації про маршрут. Ця ціль досягається шляхом інтегрування операційного обладнання в навколишнє середовище користувача, тобто в сам транспортний засіб. Така система має певні вимоги до автомобіля, а саме: засіб

проекції на лобове скло, GPS та інші пристрої відстеження положення і камера.

Програмний застосунок використовує зображення з камери, порівнює його зі збереженими картами вулиць та створює 3D картину доріг. Ця 3D картина анутується дорожніми позначками та проектується у вигляді доповненої реальності на лобове скло авто, приклад такої проекції зображено на рисунку 12. Також слід зазначити, що дані геолокації можуть бути отримані не тільки з GPS, а і з систем відстеження положення авто, що використовує рух коліс.



Рис. 12 Авто-навігація з використанням доповненої реальності

1.4.2 Навігація з доповненою реальністю для пішоходів

Навігаційні системи створюються не тільки для автомобілей, а і для пішоходів також. Існує ряд застосунків, які працюють як в режимі авто-навігації, так і в режимі навігації для пішоходів. Основна мета доповненої реальності в навігації для авто і пішоходів має схожий характер.

Переважна більшість навігаційних застосунків для пішоходів розроблена для сучасних смартфонів. Програма використовує сенсори орієнтації для відображення контенту доповненої реальності з максимальною ефективністю, для збереження енергії мобільного пристрою. Дані про положення отримуються з GPS сенсорів. Якщо GPS тимчасово недоступний, то місцепо-

ложення може бути розраховано використовуючи інформацію з сенсорів орієнтації та акселерометру. Візуальні підказки накладаються на зображення з камери смартфона, разом з віртуальними об'єктами доповненої реальності. Приклад такого застосунку продемонстровано на рисунку 13.

Досить часто, для зменшення навантаження на мобільний пристрій, важкі розрахунки (рендеринг графіки, побудова маршрутів) переносяться на віддалений сервер, в той час як смартфон виступає у ролі клієнту.

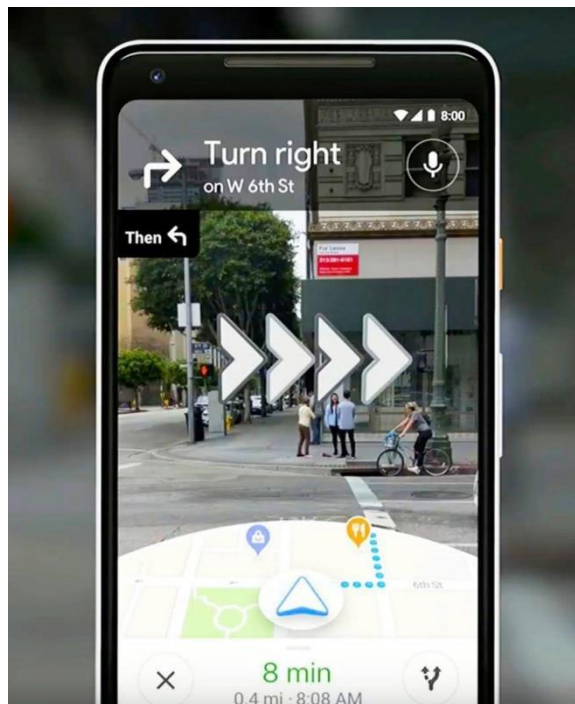


Рис. 13 Google Maps AR Navigation

1.4.3 Навігація з доповненою реальністю в приміщенні

Доповнена реальність також може бути використана для навігації в приміщенні. Цей спосіб навігації відрізняється від усіх приведених раніше, адже відомо, що точність даних GPS значно погіршується в закритому просторі. В таких випадках замість GPS використовуються інші технології для визначення місцеположення користувача, а саме: WI-FI, Bluetooth або інші бездротові рішення.

Деякі системи використовують фізичні маркери для визначення початкового положення. В таких системах місцезнаходження користувача не об-

новлюється з часом, а визначається за допомоги інформаційних точок (точок інтересу), що роздруковані на підлозі. Приклад застосунку для навігації в приміщенні наведено на рисунку 14.

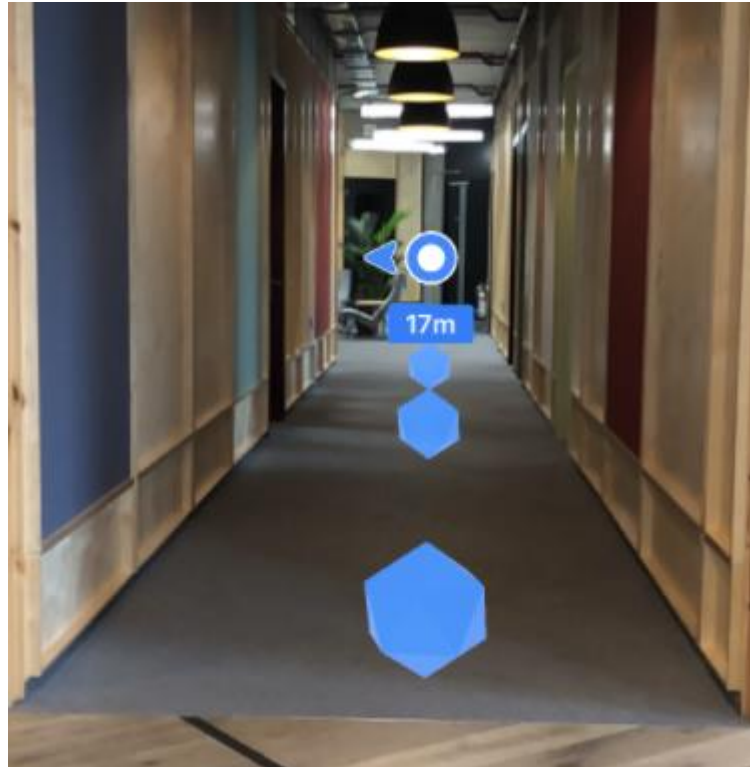


Рис. 14 Навігація в приміщенні з використанням доповненої реальності

1.5 Аналіз існуючих систем доповненої реальності

На сьогоднішній день існує велика кількість навігаційних застосунків з використання технології доповненої реальності як для водіїв авто, так і для пішоходів. Також існують рішення для навігації в приміщенні, але жоден із доступних застосунків не поєднує ці два типи навігації.

1.5.1 Google Maps

Самим популярним навігаційним застосунком для смартфонів під управлінням операційної системи Android є Google Maps.

Google Maps використовує традиційні методи навігації з GPS та картами місцевості. З 2018 року має функцію навігації з доповненою реальністю для пішоходів в місті.

До переваг Google Maps можна віднести: зручний інтерфейс, сумісність з переважною більшістю пристроїв під контролем операційної системи Android та можливість використовувати доповнену реальність в будь-якому місті.

Із недоліків слід виділити: відносну неточність положення віртуального контенту та потреба в постійному інтернет з'єднанні. Роботу застосунку продемонстровано на рисунку 15.

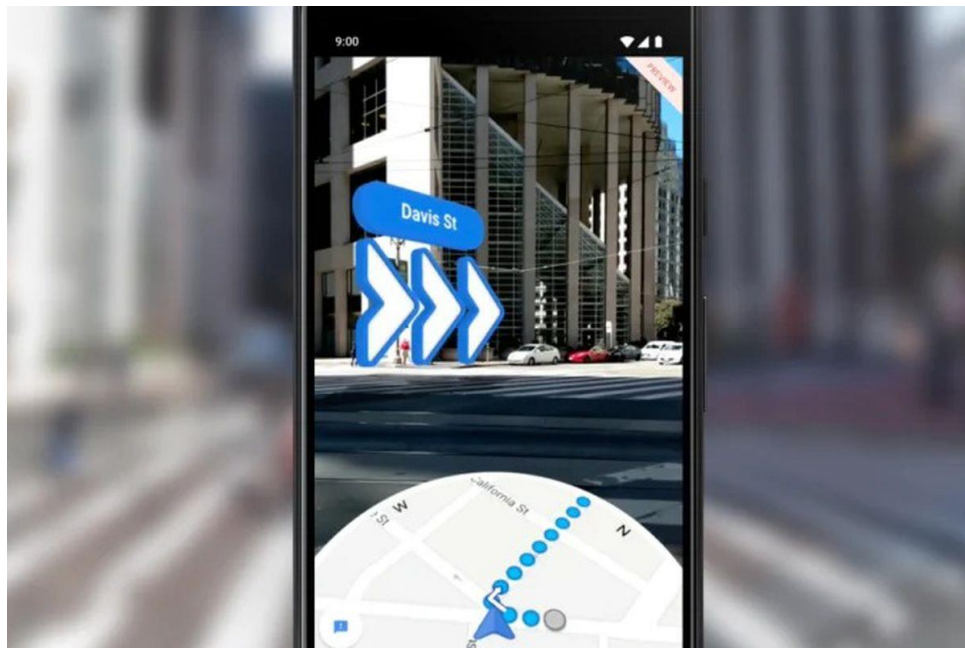


Рис. 15 Google Maps, AR режим

1.5.2 Apple Maps

Стандартним рішенням проблеми навігації на пристроях під контролем операційної системи iOS є застосунок Apple Maps. Як і Google Maps, Apple Maps використовує традиційні методи побудови навігаційних шляхів, як було описано раніше. Функція навігації з використанням доповненої реальності була реалізовано відносно нещодавно, на початку 2021го року.

До переваг Apple Maps можна віднести: високу точність положення віртуального контенту та відносну енергоефективність.

Серед недоліків слід виділити: потреба в новому програмному забезпеченні (iOS 15), новому апаратному забезпеченні (чіп A12) та постійному інтернет з'єднанні. Також AR навігації можлива лише в декількох великих містах. Роботу застосунку продемонстровано на рисунку 16.



Рис. 16 Apple Maps, AR режим

1.5.3 Результати аналізу існуючих систем

Усі популярні застосунки навігації схожі між собою та використовують стандартні методи побудови маршрутів. Навігація в приміщенні взагалі не реалізована в більшості застосунків, так як для встановлення місцеположення користувача використовуються лише дані з GPS.

Все це робить актуальним розробку застосунку для одночасної навігації на вулиці та в приміщенні.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАСОБІВ ПОБУДОВИ СТІЙКОЇ СИСТЕМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ З ВИКОРИСТАННЯМ GPS

2.1 Методи комп'ютерного зору в технології доповненої реальності

В основі усіх безмаркерних систем доповненої реальності використовується метод одночасної локалізації та мапування — **SLAM**.

SLAM (від *англ.* Simultaneous Localization and Mapping) — це алгоритм одночасної побудови і оновлення мапи невідомого оточення з одночасним відстежуванням місцеположення під час руху.

SLAM може розглядатися як теоретична проблема приближення значень. Параметрами, що приближуються є ρ позицій робота в різні моменти часу та положення n орієнтирів. Вимірювання отримується з одометрії та спостережень орієнтирів. Спостереження орієнтира надає його позицію у відношенні до положення робота у певний момент часу. Одометрія, в свою чергу, визначає відносне положення робота між двома послідовними точками часу. Алгоритм припускає, що дана апіорна модель невизначеності кожного вимірювання.

Важливою властивістю алгоритму є те, що кожне вимірювання включає лише два об'єкти, кожен з яких має 2 або 3 параметри. Таким чином, цілком природно розглядати всю систему як граф, в якому пози робота та позиції орієнтирів виступають у ролі вершин, а виміри утворюють ребра, як зображено на рисунку 17.

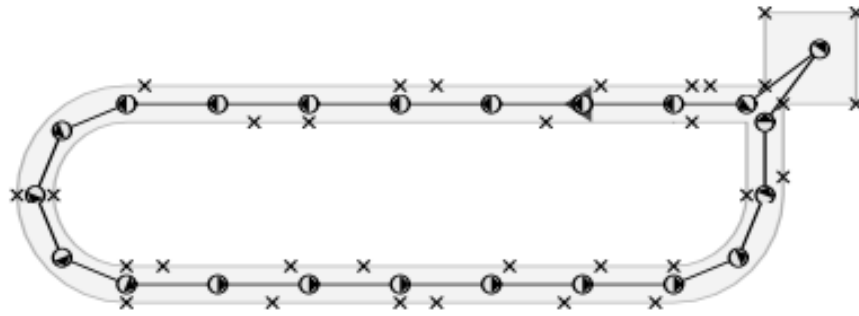


Рис. 17 SLAM система у вигляді графу

Для кращого розуміння структури проблеми можна провести фізичну аналогію. Уявіть, що граф це укріплена металева структура в якій болти виступають у ролі вершин, а еластичні металеві пруті, що їх з'єднують, у якості ребр. Нехай жорсткість прутів відповідає за точність вимірювань. Болти можуть рухатися відносно один одного. Незначні зрушення обумовленні невизначеними відносинами, а сильні зміщення обумовленні точно відомими відносинами. Одометричну послідовність можна розглядати як довгий тонкий металевий шпон який з'єднується з прутами — спостереженням орієнтирів [17].

2.1.1 Накопичення помилки

Якщо робот рухається через відоме середовище, тобто за допомогою апріорної карти, невизначеність пози робота може бути збережено на низькому рівні, оскільки кожне спостереження за орієнтиром зменшує невизначеність до невизначеності орієнтира плюс невизначеність спостереження.

Однак, якщо робот рухається через невідому місцевість, невизначеність його пози може стати необмежено великою, тому що одометрична похибка накопичується з часом. Приклад графу з одометричною похибкою зображено на рисунку 18. Невизначеність може бути зменшена шляхом об'єднання одометрії з декількома вимірами нових точок орієнтації, як зображено на рисунку 19. Для більшості сенсорів це приводить до набагато ліпших результатів, ніж використання лише однієї одометрії [18]. Тим не менш, оцінка положен-

ня робота після подорожі на велику відстань все ще піддається накопиченню помилки. Із-за обмеженого діапазону сенсорів позиція отримується з ланцюгу, що об'єднує лише декілька відносин зв'язних точок орієнтації.

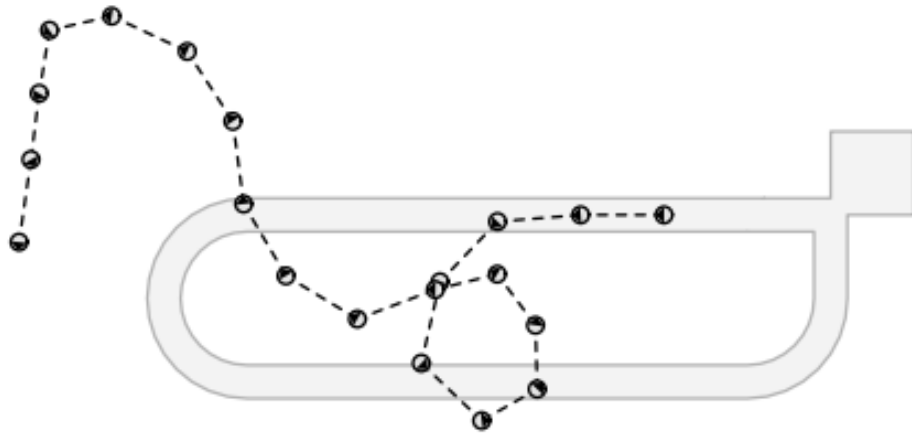


Рис. 18. Накопичення одометричної похибки



Рис. 19 Об'єднання одометрії з вимірами нових точок орієнтації

Для застосунків, що працюють на вулиці, ця проблема може бути зглажена за допомоги компаса [19], однак який, як відомо, погано працює в будівлях, які містять в своїй основі велику кількість сталі.

Той факт, що помилки можуть накопичуватися до досить високих значень, відрізняє SLAM від багатьох інших методів приближення значень та породжує проблеми, які описуються в розділах 2.1.2 та 2.1.4.

2.1.2 Представлення відносності

Домінуючим аспектом алгоритму SLAM є необхідність побудувати модель — «Точність відношень, незважаючи на невизначеність позицій», така модель має назву «представлення відносності». Наприклад, поза кімнати буде досить невизначеною, а її форма — дуже точна.

Якщо робот рухається через раніше невідомий регіон і спостерігає послідовність орієнтирів, точність взаємного розташування залежить лише від похибки зчитуванням орієнтирів роботом та одометричної похибки між цими вимірами. Тож найточніші відносини — це відносини між сусідніми точками орієнтації.

Однак, невизначеність абсолютної пози роботи перед знаходженням першого орієнтиру збільшує неточність абсолютного положення всіх орієнтирів, тобто виступає у ролі невідомої трансформації на усю сукупність спостережуваних точок орієнтації. Оскільки абсолютна поза робота піддається накопиченню помилок, як правило відносини між точками досить точні, в той час як абсолютні позиції можуть бути довільно невизначеними. На дуже великих мапах цей ефект може проявитися в різних масштабах: відносне положення деяких точок орієнтації в кімнаті краще відоме ніж положення кімнати в будівлі, а відношення кімнат до інших кімнат точніше ніж абсолютне положення будівлі.

Таким чином, система SLAM повинна бути спроможною відобразити визначеність відносин між орієнтирами, незважаючи на велику невизначеність їх абсолютного положення.

Хоча теоретично і можливо досягти абсолютної точності, повторюючи всі вимірювання досить часто (з точно визначеною початковою позою робота), але це загалом не є, ані практичним, ані необхідним, оскільки більшість

застосунків може працювати виключно на відносних величинах. Наприклад, для навігації немає необхідності точно обчислювати траєкторію від початку до кінця в якійсь глобальній системі координат. В результаті планування шляху скоріше буде створено послідовність точок шляху. Розташування кожної точки маршруту буде відомим відносно до навколишніх орієнтирів, тоді робот, знатиме його власну позу щодо цих орієнтирів, зможе орієнтуватися від однієї маршрутної точки до іншої.

2.1.3 Приближення максимальної імовірності

Якщо припустити незалежні похибки вимірювання Гауса з апріорно відомою коваріацією і не зважати на вимоги до швидкості обчислювання, SLAM може бути повністю вирішеним за допомогою модифікації нелінійної моделі найменших квадратів. Це виконується шляхом знаходження мінімуму \hat{x} з квадратичної функції помилки $Q(x)$:

$$q_i(x) = \frac{1}{2} (y_i - f_i(x))^T C_i^{-1} (y_i - f_i(x)) \quad (1)$$

$$Q(x) = \sum_i q_i(x), \quad \hat{x} := \arg \min_x Q(x); \quad (2)$$

$$C_\alpha := \{x \mid Q(x) - Q(\hat{x}) \leq \alpha\}.$$

Позиції точок орієнтації та різні пози роботи формують параметричний вектор x_i, y_i , що є i -им виміром. C_i — це коваріація, а $f_i(x)$ є відповідним рівнянням вимірювання, тобто значенням, яке має мати вимірювання, якщо позиції робота та орієнтирів мали би значення x . Незважаючи на те, що f_i нелінійна, мінімальний \hat{x} з Q є оцінкою максимальної ймовірності для екземпляра мапи і може бути знайденим за допомогою алгоритму Левенберга-Марквардта. Множина C_α , що охоплює \hat{x} , є областю впевненості, що визначає точність мапи, де значення α залежить від бажаного рівня визначеності.

Однак такий підхід не є практичним рішенням SLAM, оскільки він вимагає збереження всіх вимірювань і виконання ітерації з кількома системами

лінійних рівнянь, що вирішуються після кожного додавання нового виміру в систему. З n точок орієнтації та p поз робота, така система має $O((n + p)^3)$ час виконання. Однак такий підхід може бути корисним для порівняння ефективності інших рішень.

При лінеаризації рівнянь вимірювання f_i , Q стає квадратичним у параметричному векторі: $Q(x) = x^T A x + x^T b + \gamma$. Матриця A називається інформаційною матрицею. Перші значення матриці A відповідають точно відомим співвідношенням.

2.1.4 Розширений фільтр Калмана

Розширений фільтр Калмана є інструментом, який найчастіше застосовують до SLAM [20], з використанням тих самих рівнянь вимірювання, що й для оцінки максимальної правдоподібності. Розширений фільтр Калмана об'єднує всі вимірювання в коваріаційну матриці орієнтирів і фактичних поз робота, без необхідності в збереженні будь-яких вимірювань надалі.

Для лінійних моделей вимірювання фільтр Калмана виводить максимально можливе приближення. Для нелінійних рівнянь вимірювання, які з'являються в SLAM із-за можливості обертання робота, максимальність результату дотримується доти, доки рівняння можуть бути адекватно лінеаризовано. Якщо після кожного нового виміру всі вимірювання можна лінеаризувати за поточним приближенням, то результат буде еквівалентним нелінійному максимальному приближенню. З розширеним фільтром Калмана, однак, змінити точку лінеаризації після інтегрування вимірювання є неможливим, тому вимірювання зазвичай лінеаризуються з приближенням відразу після їх отримання.

Як наслідок, точка лінеаризації може бути значно неправильною під час руху через невідому область, оскільки при цьому ще накопичується одометрична похибка. На практиці помилка орієнтації робота може легко перевищити 45° , що робить лінеаризацію \sin та \cos марною.

У порівнянні з методом приближення максимальної ймовірності, розширений фільтр Калмана приводить до гірших результатів: стартові і кінцеві орієнтири мапи не співпадають, розмір кімнати значно збільшується. Такі результати обумовлені тим, що розширений фільтр Калмана повинен був би неявно переміщати та повертати кімнату щоб привести її до стійкого стану. Замість цього фільтр виконує це шляхом повертання, лінеаризуючи кут до 0:

$$R(\varphi) = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix}, \quad (3)$$

$$R(\varphi) \xrightarrow{\varphi=0} \begin{pmatrix} 1 & -\varphi \\ \varphi & 1 \end{pmatrix} = \sqrt{1 + \varphi^2} \cdot R(\arctan \varphi). \quad (4)$$

В результаті кімната має більший розмір ніж в реальності та має недостатній кут обертання.

2.1.5 Порівняння коваріаційної та інформаційної матриць

Розширений фільтр Калмана та метод найменших квадратів стають еквівалентними, тобто приводять до однакового приближення, в тому випадку, коли використовуються вимірювальні рівняння, що лінеаризуються в момент вимірювання. На рисунку 20 зображено відношення між коваріаційною матрицею C розширеного фільтру Калмана та інформаційною матрицею A лінеаризованого методу найменших квадратів.

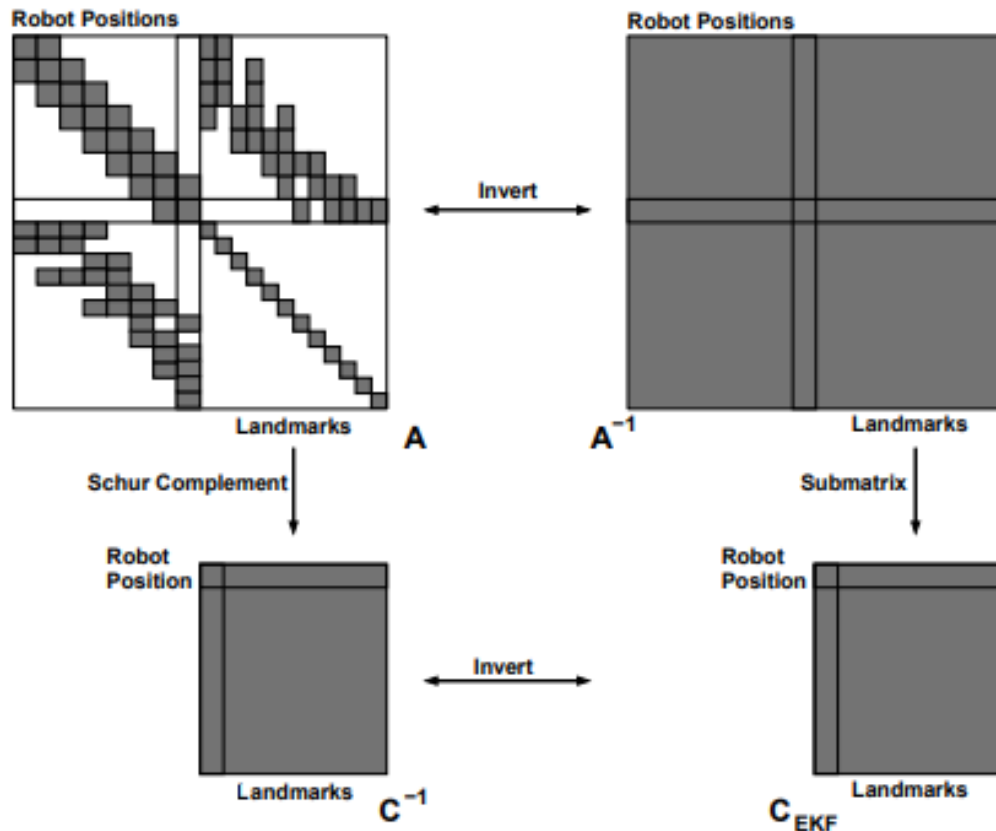


Рис. 20 Зв'язок коваріаційної та інформаційної матриць

Дві стохастично незалежні інформаційні матриці можуть бути об'єднані звичайним додаванням. Причиною цьому є те, що інформаційна матриця да набір вимірювань є сумою інформаційних матриць для кожного вимірювання. Інтегрування двох незалежних коваріаційних матриць виконується шляхом обчислення: $(C_1^{-1} + C_2^{-1})^{-1}$. Дійсно, розширений фільтр Калмана можна розглядати як кмітливий спосіб додати інформаційну матрицю, що відповідає одному вимірюванню, до зворотної коваріаційної матриці, при цьому уникаючи інвертування коваріаційної матриці шляхом використання формули Вудбері для оновлення матриць [21].

Основна відмінність інформаційної матриці від коваріаційної полягає в представленні непрямих зав'язків. Припустимо, що робот знаходиться в позі P1, спостерігаючи за орієнтиром L1 і рухається до P2, спостерігаючи L2. Вимірювання безпосередньо визначають відносини P1-L1, P1-P2, P2-L2, опосередковано складаючи відношення L1-L2. Коваріаційна матриця явно зберігає

це відношення у недиагональних записах, що відповідають L1-L2, а інформаційна матриця це відношення не зберігає.

Таким чином матриця A є розрідженою, має лише ненульові недиагональні записи для тих пар параметрів, які беруть участь у загальному вимірюванні. Обернена A^{-1} є коваріаційною матрицею для положень орієнтирів і усіх поз робота. A^{-1} представляє всі непрямі відносини явно і, таким чином, не є розрідженою. Видалення рядків і стовпців, що відповідають старим позам робота створює коваріаційну матрицю C розширеного фільтра Калмана. Обернена матриця C^{-1} є інформаційною матрицею усіх орієнтирів і фактичної пози робота. Однак, обернена матриця не є підматрицею A , так як видалення усіх поз робота з A потребує обчислення їх неявного впливу на відносини між іншими параметрами за допомоги доповнення Шура, що, в свою чергу, руйнує розрідженість матриці.

2.1.6 Роль SLAM в стійкій доповненій реальності

Саме із-за архітектури SLAM, яка використовує пози камери та позиції точок орієнтації, стало можливим створювати стійкий досвід доповненої реальності зберігаючи та відновлюючи мапу навколишнього середовища. Також, для відновлення усієї мапи достатньо ідентифікувати лише декілька точок з правильними зв'язками.

2.2 Методи збереження інформації між сесіями застосунку

В умовах обмеженості обчислюваної спроможності мобільних пристроїв в фреймворки доповненої реальності було введено поняття *анкеру*. **Анкер** — це віртуальна точка в мапі навколишнього середовища, яка має свою позу та зв'язки з орієнтирами. На підтримку та оновлення пози анкеру виділяються додаткові обчислення. Зазвичай анкери використовують для покращення точності положення віртуального контенту в доповненій реальності.

Головними фреймворками для розробки мобільних застосунків з доповненою реальністю на цей час є: ARCore для Android та ARKit для iOS пристроїв.

2.2.1 Стійка доповнена реальність в ARCore

З версією 1.20 в фреймворк ARCore було додано можливість зберігати анкери на віддаленому сервері Google, цей функціонал отримав назву — Cloud Anchors (*від англ. Хмарові Анкери*).

Cloud Anchors надає змогу зберігати користувацькі анкери на сервері з тривалістю життя в 365 днів з базовим доступом [22]. Для відновлення анкери застосунк-клієнт відправляє дані на сервер, а у разі вдалого розпізнавання сервер відповідає інформацією про позицію анкери та його унікальний ідентифікатор.

Також, Cloud Anchor має тип анкерів які оновлюються в реальному часі. Такі анкери використовуються для створення багатокористувацького досвіду доповненої реальності та мають більш обмежену тривалість життя. Використання динамічних анкерів в ARCore продемонстровано на рисунку 21.

Cloud Anchors можливо використовувати як на Android, так і на iOS пристроях, але для iOS пристроїв потрібно створити проміжний сервер для інтерпретації анкерів.

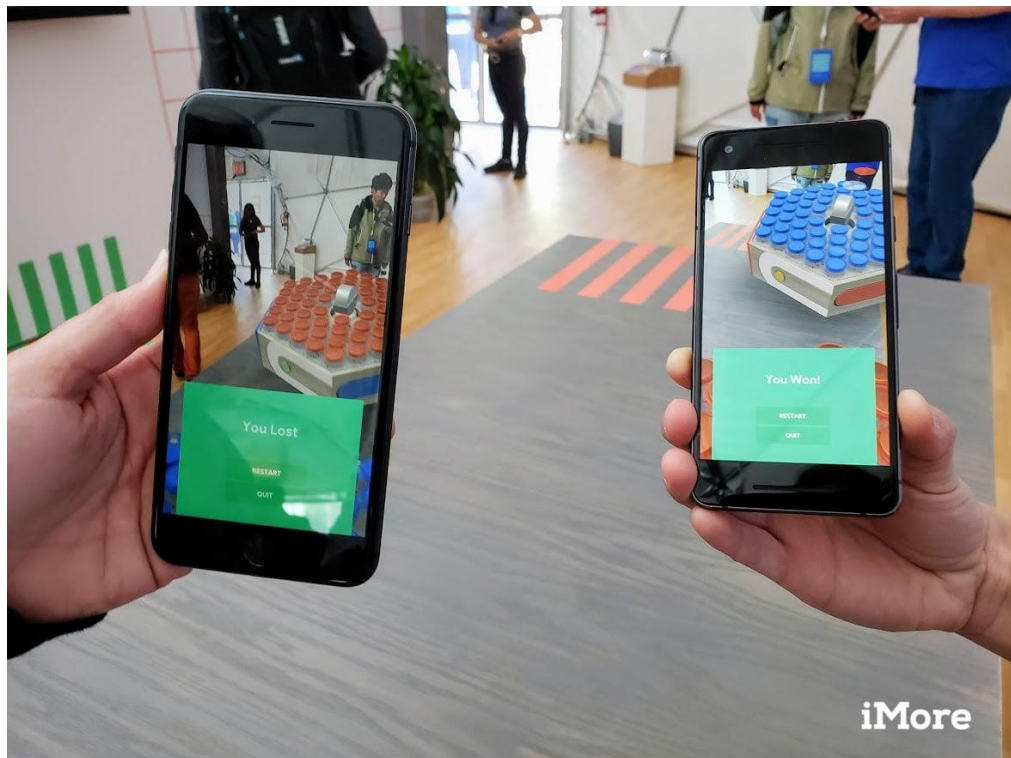


Рис. 21 Приклад використання Cloud Anchors в ARCore

2.2.2 Стійка доповнена реальність в ARKit

Можливість створювати стійку доповнену реальність в ARKit з'явилась значно раніше ніж в ARCore та використовує інший спосіб збереження сесій. ARKit зберігає повністю всю мапу навколишнього середовища локально в файл. До збереженої мапи входять: анкери, точки орієнтації та площини. Для реконструкції мапи в новій сесії застосунку потрібно просканувати місцевість на якій існувала минула сесія. Після знаходження співпадаючих точок орієнтації відбувається повне відновлення мапи з усіма її складовими об'єктами.

Багатокористувацький досвід також можливо побудувати, але для цього потрібно використовувати особистий сервер для обміну мапами між девайсами [23]. Використання динамічних анкерів в ARKit продемонстровано на рисунку 22.

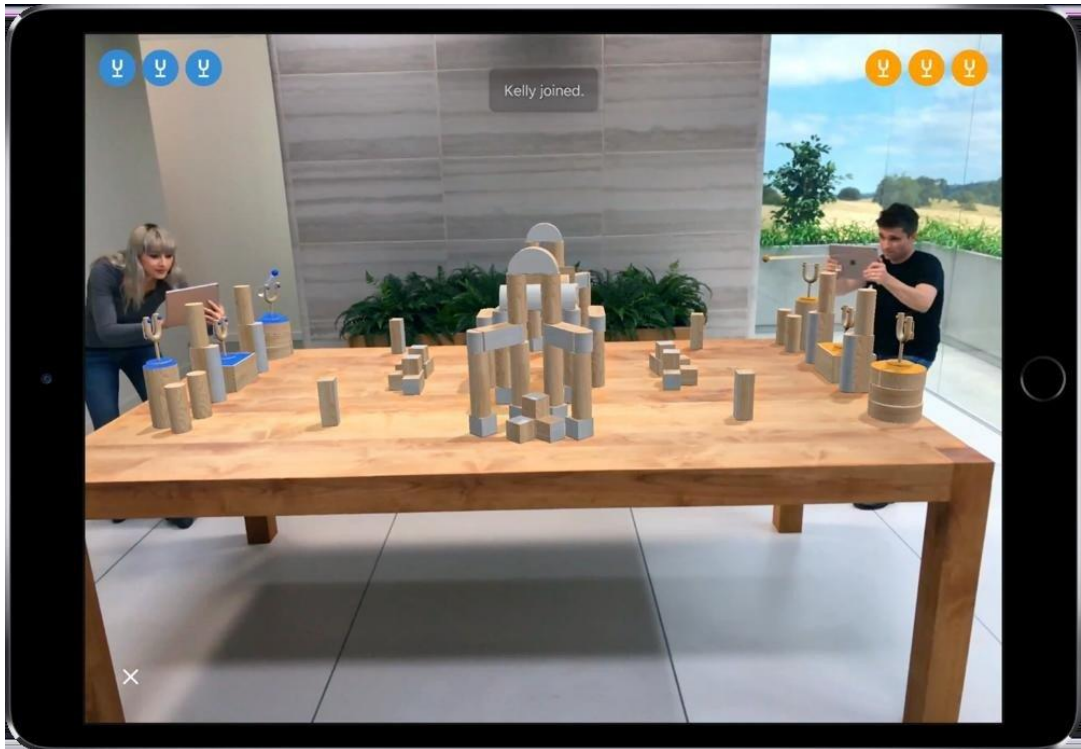


Рис. 22 Приклад використання динамічних анкерів в ARKit

2.2.3 Порівняння стійкої доповненої реальності в ARCore та ARKit

ARCore та ARKit мають можливість створення застосунків зі стійкою доповненою реальністю, але підхід до збереження інформації між сесіями у кожного фреймворка свій. Різницю в архітектурі систем стійкої доповненої реальності в ARCore та ARKit описано в таблиці 1.

Таблиця 1

Архітектура стійкої доповненої реальності у ARCore та ARKit

Характеристика	ARCore	ARKit
Спосіб збереження сесії	На віддаленому сервері	В локальному файлі
Спосіб відтворення сесії	На віддаленому сервері	Локально на пристрої
Об'єкти, що зберігаються	Анкери	Анкери, площини, точки орієнтації
Сумісні операційні системи	Android, iOS (з додатковим сервером)	iOS
Оновлення анкерів у реальному часі	Можливо	Можливо (з додатковим сервером)
Цінова політика	Безкоштовно з обмеженнями, або додаткове придбання кількості та часу життя анкерів	Безкоштовно
Доступ до інтернету	Необхідний	Не вимагається

2.3 Методи перетворення GPS координат в системі доповненої реальності

Для коректного відображення віртуального контенту в доповненій реальності по заданим координатам GPS потрібно провести перетворення значень довготи та широти з полярної системи координат в $x - z$ координати сесії доповненої реальності [24].

Перенесення GPS-координат точки A на відповідний їй вектор положення $(x, 0, z)$ в систему координат доповненої реальності включає 4 змінні: відстань $|\overrightarrow{OA}|$ між A та користувачьким O (початком координатної системи

сесії); кут між вектором \overrightarrow{OA} та північним полюсом — ϑ ; кут між початком координатної системи сесії та північним полюсом — θ ; поточний вектор положення віртуальної камери — (x_0, y_0, z_0) .

Щоб полегшити перенесення припускається, що пристрій направлений на Північний полюс і його поточний вектор положення дорівнює $(0,0,0)$ [28], як зображено на рисунку 23.

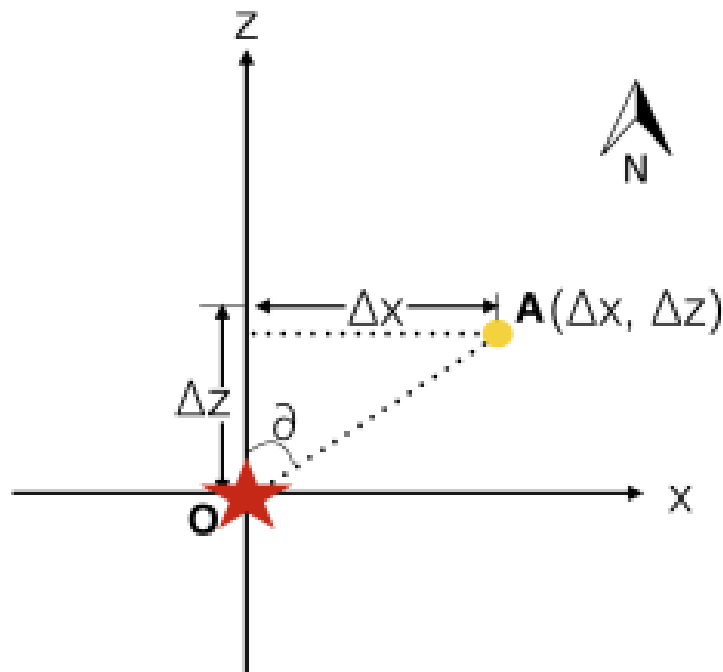


Рис. 23 Відносне положення віртуальної камери базуючись на GPS-координатах

Таким чином, зміщення точки A до віртуальної камери по осі x та осі z можна представити у вигляді:

$$\Delta x = |\overrightarrow{OA}| \times \sin(\theta), \quad (5)$$

$$\Delta z = |\overrightarrow{OA}| \times \cos(\theta). \quad (6)$$

Далі, без припущення, що пристрій направлено на північ, виходячи з x, z і куту між початком координатної системи сесії та північного полюсу —

θ , обчислюється відносні координати точки A до камери в координатній системі сесії — $(\acute{x}, 0, \acute{z})$, з тригонометричного відношення зображеного на рисунку 24. На цій діаграмі вісь z чорної системи координат вказує на північ (як було припущено раніше), тоді як вісь \acute{Z} червоної координатної системи вказує на реальне обертання пристрою. Кут між двома осями z є кутом між вектором направлення пристрою та північним полюсом θ . З цієї діаграми легко зробити висновок, що трикутник ΔOCD і трикутник ΔCAB — подібні трикутники. Відповідно:

$$\frac{OC}{AC} = \frac{CD}{CB} = \frac{OD}{AB}.$$

Крім того:

$$AB = \Delta z;$$

$$OD = \acute{x};$$

$$CD = \acute{x} \times \cos(\theta);$$

$$OC = \frac{\acute{x}}{\cos(\theta)};$$

$$AC = \acute{z} - CD;$$

$$CB = \Delta x - OC.$$

Звідки отримуємо:

$$\acute{x} = \cos(\theta) \times (\Delta x - z \times \tan(\theta)); \quad (7)$$

$$\acute{z} = \acute{x} \times \tan(\theta) + \frac{\Delta z}{\cos(\theta)}. \quad (8)$$

Нарешті, без припущення, що координати віртуальної камери дорівнюють $(0,0,0)$, отримуємо вектор реального положення A в координатах доповненої реальності (x, y, z) шляхом додавання вектору положення камери (x_0, y_0, z_0) до $(\acute{x}, 0, \acute{z})$:

$$(x, y, z) = (x_0, y_0, z_0) + (\acute{x}, 0, \acute{z}). \quad (9)$$

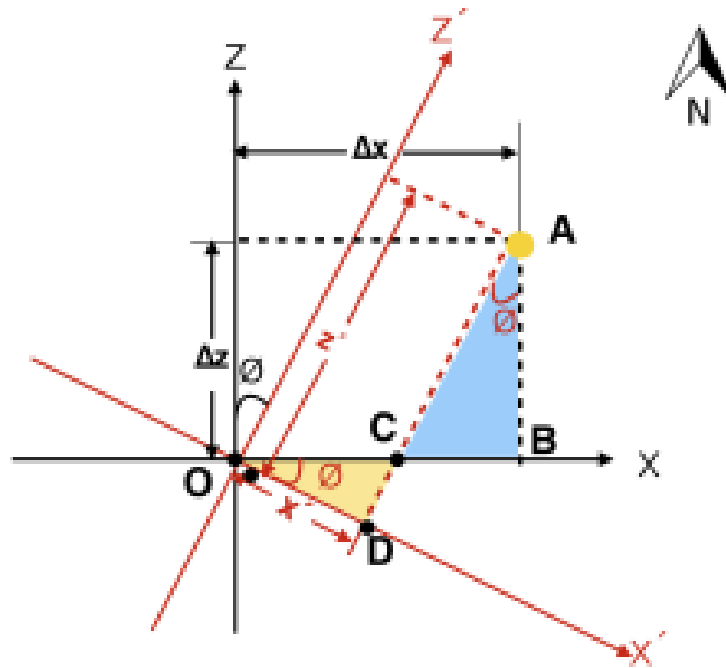


Рис. 24 Перетворення координат з локальної координатної системи камери до глобальної координатної системи

Після того, як всі координати GPS перетворені в вектори положення в системі координат сесії, координати віртуальної камери автоматично оновлюються шляхом візуального відстеження, просторове співвідношення між всіма об'єктами інтересу і користувачем також можна оновити з рухом пристрою.

Для досягнення найкращої точності перетворення координат повторюється кожного разу, коли користувач потрапляє до однієї з точок інтересу [29]. Кожне повторення проводиться на основі поточного GPS положення користувача, поточного вектора положення віртуальної камери в координатній системі сесії — (x_n, y_n, z_n) і початкового кута між пристроєм та північним полюсом.

Клас “PathCreator” відповідає за створення шляху навігації: розташування точок навігації, збереження GPS та AR координат навігаційних точок, серіалізації та десеріалізації файлу шляху.

Клас “PathManager” контролює відображення обраного шляху: візуалізує точки навігації, відображає інформацію про маршрут.

Клас “GPSEncoder” відповідає за приведення GPS координат до координатної системи сесії доповненої реальності.

Клас “GPSToARPathManager” виступає у ролі контролера для переходу між GPS та AR навігаціями.

Клас “MapRemoteManager” відповідає збереження та завантаження файлів шляху та мапи навколишнього середовища.

Застосунок написаний на мові програмування C# з використанням об’єктно-орієнтованого, компонентно-орієнтованого та реактивного програмування.

Компонентно-орієнтоване програмування

Використання компонентно-орієнтованого програмування обумовлено архітектурою системи скриптування движку Unity. Компонентно-орієнтоване програмування — це техніка розробки програмних додатків шляхом поєднання вже існуючих і нових компонентів, майже так само, як автомобілі будуються з інших компонентів. Компоненти програмного забезпечення — це автономні пакети функціональних можливостей, що описують самі себе та містять визначення типів, які відображають як поведінку, так і дані.

C# підтримує компонентно-орієнтоване програмування за допомогою концепцій властивостей, методів, подій та атрибутів (або метаданих), дозволяючи автономні й самоописувані компоненти функціональності, які називаються збірками.

Движок Unity побудований на компонентних відносинах між об’єктами, що дозволяє створювати гнучкі кросплатформні системи.

Реактивне програмування

Реактивне програмування описує парадигму проектування, яка покладається на логіку асинхронного програмування для обробки оновлень у реальному часі статичного вмісту. Реактивне програмування забезпечує ефективний засіб використання автоматизованих потоків даних для обробки оновлень даних вмісту щоразу, коли користувач робить запит.

Потоки даних, що використовуються в реактивному програмуванні є зв'язаними колекціями цифрових сигналів, створених на постійній або майже безперервній основі. Ці потоки даних надсилаються з джерела - наприклад, датчика руху, датчика температури або бази даних інвентарю продукції - у відповідь на тригер.

Реактивне програмування створює програмне забезпечення, яке реагує на події, а не вимагає дій від користувачів. Подія — це просто сигнал про те, що щось сталося. Загально визнано, що події є сигналами «реального часу», тобто вони генеруються одночасно з умовою, про яку вони сигналізують, і вони також повинні оброблятися в режимі реального часу. Ці події найкраще візуалізувати як «потоки», які можуть протікати через кілька елементів обробки, зупинятися й оброблятися на цьому шляху, або розгалужуватися та генерувати паралельну обробку.

Підхід реактивного програмування було використано при розробці класу “ExploreARWebAPI”, що відповідає за взаємодію з віддаленим сервером.

3.2 Засоби реалізації

Під час розробки програмної системи, найоптимальнішими засобами реалізації було обрано:

- Ігровий движок Unity.
- Середовище розробки Visual Studio 2019.
- Мова програмування C# та фреймворк .NET Framework.

- Кросплатформний фреймворк-обгортка для використання функціоналу доповненої реальності ARFoundation.
- Фреймворк доповненої реальності для iOS пристроїв ARKit
- Середовище розробки Xcode.

3.2.1 Ігровий движок Unity

На даний момент Unity є найпопулярнішим ігровим движком, який використовують розробники по всьому світу — і не дарма. Він має потужний візуальний інтерфейс для створення ігор, кросплатформну розробку та активну спільноту. Unity — це кросплатформний ігровий движок, який в основному використовується для розробки відеоігор і моделювання для ПК, консолей, мобільних пристроїв і веб-сайтів. Він розроблений Unity Technologies і був першим оголошений лише для OS X на Всесвітній конференції розробників Apple у 2005 році, з тих пір движок був поширений майже на кожній доступній платформі.

До списку деяких з численних функцій Unity з технічної точки зору входять:

- Створення та знищення ігрових об'єктів.
- Доступ до компонентів.
- Події для ігрових об'єктів.
- Робота з векторними змінними та змінними часу.
- Події, орієнтовані на фізику.
- Сопрограми (Coroutines).

Також Unity має фреймворки для створення застосунків віртуальної та доповненої реальності.

Система координат

Важливим аспектом створення застосунків з використанням доповненої реальності є координатна система движка. Система координат в Unity визначається трьома осями: X, Y та Z, де вісь Y вказує вгору. В системі Unity координати поділяються на глобальні та локальні.

Локальні координати визначаються в координатній системі батьківського 3D об'єкту, що дозволяє рухати дочірні об'єкти відносно батьківського.

Глобальні координати — це положення об'єкту відносно всіх інших об'єктів на сцені та початку координатної системи. Усі об'єкти мають локальні та глобальні координати.

Одиниця виміру в координатній системі Unity приблизно дорівнює одному метру фізичного світу, що є дуже практичним для підтримки правильності пропорцій та положень віртуальних об'єктів в аугментованому доповненою реальністю фізичного простору. Відношення координатних систем в доповненій реальності зображено на рисунку 26.

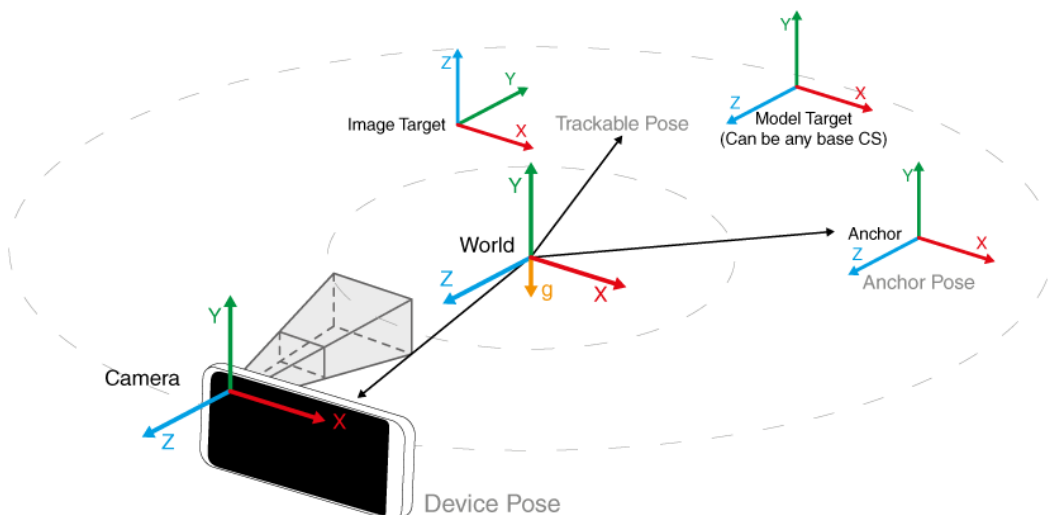


Рис. 26 Координатні системи Unity в доповненій реальності

3.2.2 Середовище розробки Visual Studio

Основним середовищем розробки застосунків Unity є Visual Studio. Це обумовлено тим, що написання коду на мові С# та фреймворку .NET краще всього інтегровано саме в цю IDE. Також в Visual Studio можливо інсталиювати плагін Unity, що дозволяє швидко писати стандартні методи Unity, такі як *Start*, *Update*, *OnEnable* тощо, завдяки тому, що середовище розробки автоматично доповнює назву методів та заповнює його параметри.

Також Visual Studio дозволяє зручно відлагоджувати написані скрипти, ставлячи точки зупинки виконання та простежувати помилки за стеком викликів.

3.2.3 Мова програмування С# та фреймворк .NET

С# — сучасна об'єктно-орієнтована мова програмування загального призначення, розроблена компанією Microsoft і схвалена Європейською Асоціацією Виробників Комп'ютерів (ECMA) та Міжнародною Організацією Стандартів (ISO).

С# був розроблений Андерсом Хейлсбергом та його командою під час розробки .NET Framework. С# розроблено для загальномовної інфраструктури (CLI), яка складається з виконуваного коду та середовища виконання, що дозволяє використовувати різні мови високого рівня на різних комп'ютерних платформах та архітектурах [25].

Конструкції С# тісно відповідають традиційним мовам високого рівня: С та С++. С# дуже схожий з Java, та має численні потужні функції програмування, які роблять цю мову привабливою для багатьох програмістів у всьому світі. До важливих складових С# можна віднести:

- Булеві операції.
- Автоматичний збір сміття.
- Стандартна бібліотека.
- Керування версіями збірки
- Властивості та події.

- Делегати та управління подіями.
- Прості у використанні генерики.
- Індексатори.
- Умовна компіляція.
- Проста багатопоточність.
- LINQ та лямбда-вирази.
- Інтеграція з Windows.

.NET Framework — це клас бібліотек багаторазового використання (набір класів), наданих Microsoft для використання в інших програмах .NET, а також для розробки, створення та розгортання багатьох типів програм на платформі Windows, включаючи такі:

- Консольні програми.
- Програми Windows Forms.
- Програми Windows Presentation Foundation (WPF).
- Веб-додатки.
- Веб-сервіси.
- Служби Windows.

Бібліотека включає в себе усі необхідні класи та інтерфейси для зручної та швидкої розробки застосунків, а саме: класи генерик-колекцій, класи роботи з файловою системою, базові алгоритми тощо.

3.2.4 AR Foundation

ARFoundation — це фреймворк Unity, який дозволяє створювати програми доповненої реальності для Android та iOS без зміни SDK. ARFoundation об'єднує функціональні можливості ARCore та ARKit в єдині абстрактні інтерфейси, які, після компіляції, використовують нативну бібліотеку цільової платформи.

Такий підхід до розробки застосунків з доповненою реальністю підтримує кросплатформеність движку Unity, та дозволяє швидко і зручно створювати застосунки для обох платформ. Звісно, ARFoundation також надає

змогу використовувати унікальний функціонал кожної з платформ, таких як CloudAnchors у ArCore та ARWorldMap у ArKit. Абстрактна архітектура роботи ARFoundation зображена на рисунку 27.

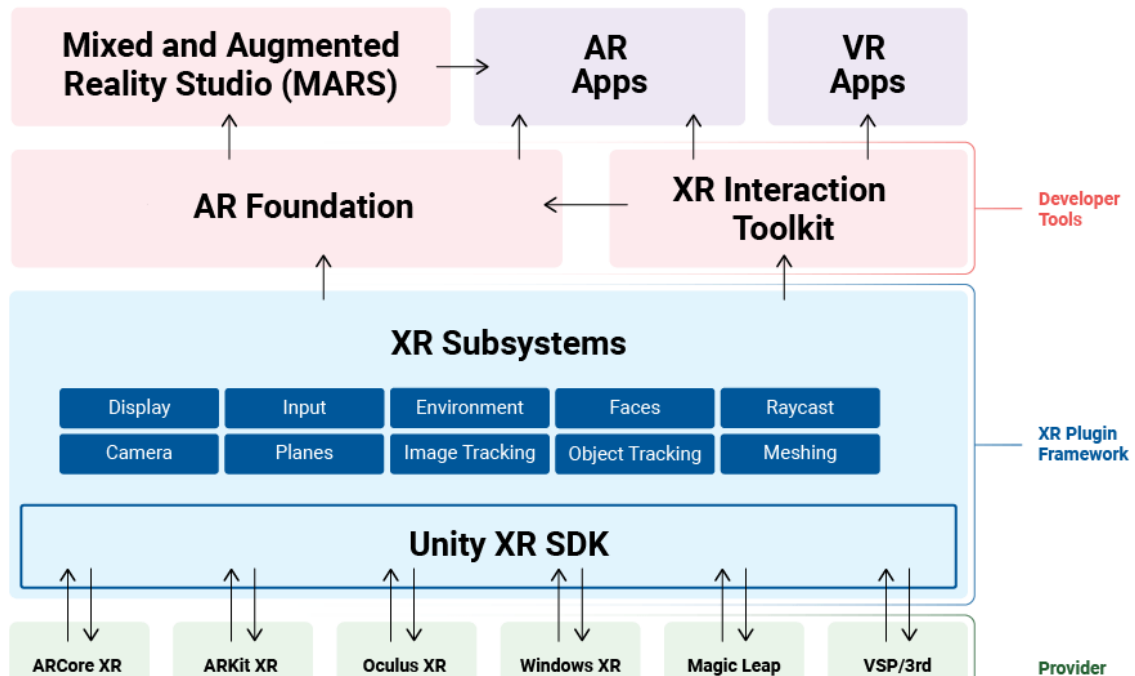


Рис. 27 Архітектура фреймворку ARFoundation

3.2.5 AR Kit

ARKit — це фреймворк для розробки застосунків доповненої реальності для пристроїв на операційній системі iOS. ARKit використовує візуальну одометрію для точного відстеження світу, поєднуючи дані датчика камери з даними датчиків руху. Ці дані дозволяють пристрою iOS точно визначати, як він рухається в кімнаті, усуваючи необхідність додаткового калібрування [26].

Для зйомки живої трансляції оточення і відстеження відмінностей у зображенні при зміні кута нахилу може використовуватися як основна, так і фронтальна камера.

ARKit здатний знаходити горизонтальні і вертикальні площини в кімнаті, наприклад на столі або підлозі, які потім можна використовувати для

розміщення на них цифрових об'єктів. ARKit також відстежує позиції віртуальних об'єктів, навіть коли об'єкт тимчасово виходить з кадру.

Система також використовує датчик камери для оцінки освітлення навколишнього середовища. Ці дані корисні для застосування світлових ефектів до віртуальних об'єктів, допомагаючи точно імітувати навколишнє середовище реального світу, створюючи ілюзію, що віртуальний предмет насправді знаходиться в фізичному світі.

ARKit використовує усі доступні ресурси пристрою для створення реалістичного світу доповненої реальності. До унікальних можливостей фреймворку можна віднести: відстеження обличчя з використанням сенсору глибини фронтальної камери, перекриття віртуального контенту реальними фізичними об'єктами та створення 3D мапи середовища з використанням LIDAR сенсору. Приклад використання LIDAR сенсору зображено на рисунку 28.

Для реалізації стійкості доповненої реальності було вирішено використати модуль ARWorldMap у ARKit. Такий вибір був обумовлений незалежністю від сторонніх онлайн сервісів та особливістю збереження мапи навколишнього середовища зі всіма анкерами, що є найбільш доречним для застосування навігації.

Збережені мапи навколишнього середовища є сумісними між різними версіями операційної системи iOS та різним апаратним забезпеченням пристроїв. Однак, слід зауважити, що під час розробки застосунку було випущено новий смартфон — iPhone 13 Pro Max, і мапи створенні на ньому не десералізувалися на минулих версіях пристрою. Ця проблема була усунена в версії iOS 15.1.1.

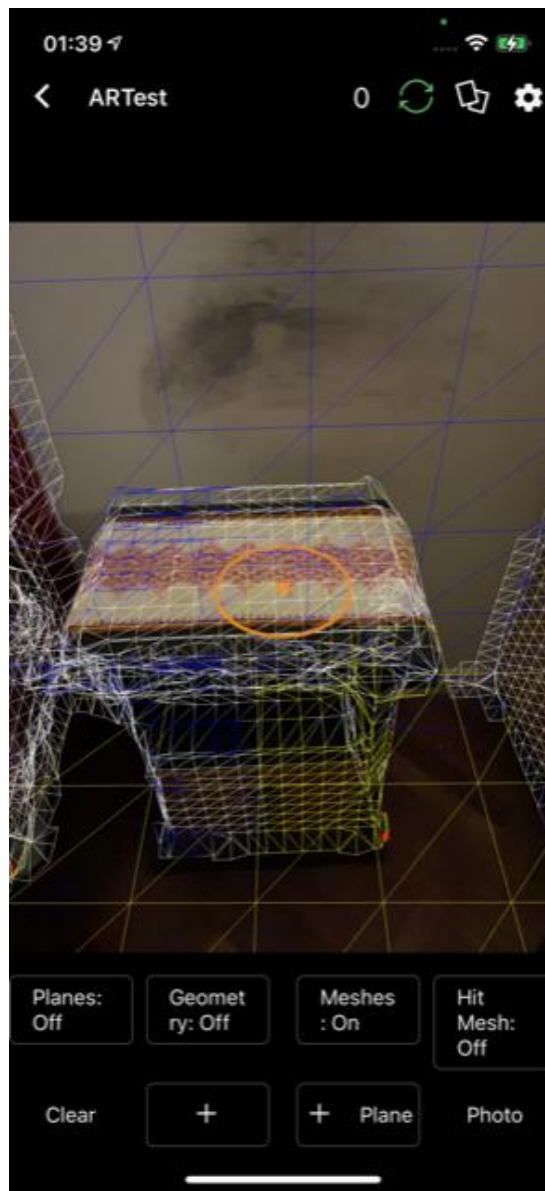


Рис. 28 Використання LIDAR сенсору для 3D-мапіну кімнати

3.2.6 Xcode

Xcode — це інтегроване середовище розробки (IDE) Apple, яке використовується для створення застосунків для продуктів Apple, включаючи iPad, iPhone, Apple Watch і Mac. Xcode надає інструменти для керування всім робочим процесом розробки — від створення програми до тестування, оптимізації та надсилання її в App Store.

Компіляції застосунків для iOS можлива *тільки* в XCode. Збірка проекту в Unity, з цільовою платформою iOS, створює проект для Xcode. Так як Xcode доступний тільки на платформах під керуванням MacOS, то створення

застосунку доповненої реальності для iOS потребує наявність двох пристроїв від компанії Apple, що, звісно, є обмеженням зручності розробки.

Для компіляції застосунку використовувалась остання стабільна версія Xcode — 13.1, саме в цій версії було додано інструменти для збірки застосунків на iOS 15.1.1.

3.3 Система клієнт-сервер

Створену систему навігації з використанням доповненої реальності та GPS було інтегровано в клієнтську частину застосунку соціальної мережі. Серверна частина створена з використанням фреймворку Node.js. Система навігації взаємодіє з віддаленим сервером шляхом відправлення та завантаження файлів навігації та мап навколишнього середовища. За комунікацію з сервером відповідає клас “ExploerARWebApi”.

3.4 Модулі та алгоритми

Розроблена система навігації з використанням доповненої реальності та GPS поділяється на чотири модулі, а саме:

1. Модуль створення шляху навігації
2. Модуль відображення GPS шляху
3. Модуль відображення шляху доповненої реальності
4. Модуль взаємодії з віддаленим сервером

3.4.1 Модуль створення шляху навігації

До модулю створення шляху навігації відносяться класи: PathCreator, ARWorldMapController, MapFileManager, NavPath, NavMap, Arrow, AnchorController. Архітектуру модулю зображено на рисунку 29.

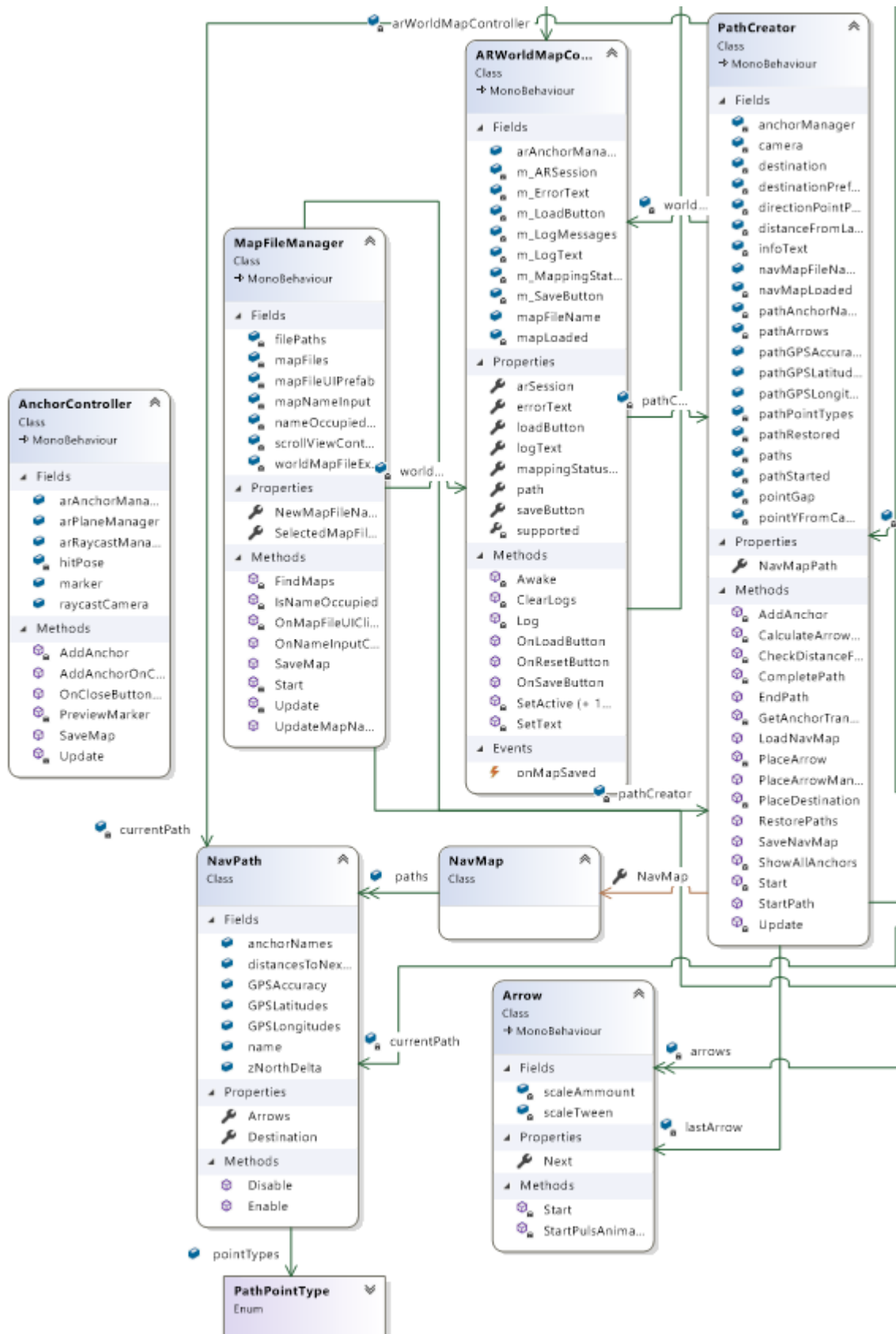


Рис. 29 Класова архітектура модулю створення шляху

Основними алгоритмами модулю створення шляху є: розміщення точки навігації, створення анкера, збереження мапи навколишнього середовища до файлу, завантаження файлу мапи та завершення шляху навігації.

В лістингу 1 наведено алгоритм розміщення точки навігації в просторі доповненої реальності.

Лістинг 1 Алгоритм розміщення точки навігації

```
private void PlaceArrow()
{
    Arrow nextArrow =
Instantiate(directionPointPrefab).GetComponent<Arrow>();
    Vector3 cameraRotation =
camera.transform.rotation.eulerAngles;
    if (lastArrow != null)
        lastArrow.Next = nextArrow;
    lastArrow = nextArrow;
    lastArrow.transform.position =
camera.transform.position;
    lastArrow.transform.Translate(-Vector3.up *
pointYFromCamera);
    lastArrow.transform.rotation =
Quaternion.Euler(0, cameraRotation.y, cameraRotation.z);
    Pose anchorPose = new
Pose(lastArrow.transform.position,
lastArrow.transform.rotation);
    lastArrow.transform.parent =
AddAnchor(anchorPose, PathPointType.Arrow).transform;
    pathArrows.Add(lastArrow);
}
```

Змінна `lastArrow` на початку виконання алгоритму є посиланням на останню точку в шляху навігації, для зв'язування точок по типу од-нозв'язного лінійного списку. Після зв'язування точок змінна `lastArrow` перевизначається посиланням на нову точку, якій визначається правильна поза тазначається анкер.

Властивості `lastArrow.transform.parent` присвоюється батьківський об'єкт-анкер для підтримки точності положення в доповненій реальності та відновлення точки після завантаження мапи навколишнього середовища при навігації.

Алгоритм створення анкеру наведено в лістингу 2.

Лістинг 2 Алгоритм створення анкеру

```
private ARAnchor AddAnchor(Pose pose, PathPointType
objectType)
{
    ARAnchor anchor =
anchorManager.AddAnchor(pose);
    pathAnchorNames.Add(anchor.name);
    pathPointTypes.Add(objectType);
    LocationInfo location =
GPSLocationManager.Instance.CurrentLocationInfo;
    pathGPSLatitudes.Add(location.latitude);
    pathGPSLongitudes.Add(location.longitude);

pathGPSAccuracy.Add(location.horizontalAccuracy);
    return anchor;
}
```

Аргумент методу `pose` має інформацію про позу, за якою має бути розташовано анкер.

Аргумент `objectType` — тип точки навігації, тобто проміжна точка чи кінцева.

Змінна `anchor` — анкер до якого прив'язується точка навігації.

Змінна `location` — структура, що має інформацію про поточне положення користувача в GPS координатах.

Мапа навколишнього середовища зберігається в файлову систему пристрою, після чого цей файл і файл сеарілізованого шляху навігації будуть завантажені на віддалений сервер. Алгоритм збереження мапи навколишнього середовища наведено в лістингу 3.

Лістинг 3 Алгоритм збереження мапи навколишнього середовища

```

void SaveAndDisposeWorldMap(ARWorldMap worldMap)
{
    var data = worldMap.Serialize(Allocator.Temp);
    Log(string.Format("Map " + mapFileName + " was
saved, size: {0} mb.", data.Length / 1000000));
    var file = File.Open(path, FileMode.Create);
    var writer = new BinaryWriter(file);
    writer.Write(data.ToArray());
    writer.Close();
    data.Dispose();
    worldMap.Dispose();
    onMapSaved?.Invoke();
    ITDimension.ExploreAR.Navigation.MapRemoteManager.SaveM
ap(path, pathCreator.NavMapPath);
}

```

Аргумент `worldMap` — мапа навколишнього середовища поточної сесії ARKit.

Змінна `data` — масив байтів сереалізованої мапи навколишнього середовища.

Метод `MapRemoteManager.SaveMap` — зберігає мапу навколишнього середовища на віддаленому сервері.

Алгоритм завантаження, десеріалізація та застосування мапи навколишнього середовища наведено в лістингу 4.

Лістинг 4 Алгоритм завантаження та десеріалізації мапи

```

IEnumerator Load()
{
    var file = File.Open(path, FileMode.Open);
    if (file == null)
    {
        yield break;
    }
    int bytesPerFrame = 1024 * 10;
    var bytesRemaining = file.Length;
    var binaryReader = new BinaryReader(file);
    var allBytes = new List<byte>();
    while (bytesRemaining > 0)
    {
        var bytes =
binaryReader.ReadBytes(bytesPerFrame);
        allBytes.AddRange(bytes);
        bytesRemaining -= bytesPerFrame;
        yield return null;
    }
    var data = new
NativeArray<byte>(allBytes.Count, Allocator.Temp);
    data.CopyFrom(allBytes.ToArray());
    ARWorldMap worldMap;
    if (ARWorldMap.TryDeserialize(data, out
worldMap))
        data.Dispose();
    if (!worldMap.valid)
    {
        yield break;
    }
    sessionSubsystem.ApplyWorldMap(worldMap);
    mapLoaded = true;
    arAnchorManager.anchorsChanged +=
OnAnchorsChanged;
}

```

Змінна `data` — нативний масив, що містить байти сереалізованої мапи навколишнього середовища.

Змінна `worldMap` — об'єкт фреймворку ARKit, що представляє мапу навколишнього середовища з анкерами, точками орієнтації та площинами.

Після завершення побудови шляху навігації відбувається серіалізація точок навігації, GPS координат, назв анкерів та кут до північного полюсу поточної сесії. Алгоритм завершення шляху наведено в лістингу 5.

Лістинг 5 Алгоритм завершення побудови шляху навігації

```

private void CompletePath()
{
    currentPath.anchorNames =
pathAnchorNames.ToArray();
    currentPath.pointTypes =
pathPointTypes.ToArray();
    currentPath.GPSLatitudes =
pathGPSLatitudes.ToArray();
    currentPath.GPSLongitudes =
pathGPSLongitudes.ToArray();
    currentPath.GPSAccuracy =
pathGPSAccuracy.ToArray();
    float zNorthDelta =
Vector3.SignedAngle(Vector3.forward,
GPSLocationManager.Instance.NorthFromCamera, Vector3.up);
    currentPath.zNorthDelta = zNorthDelta;
    currentPath.name =
UIManager.Instance.PathNameInputText;
    currentPath.Arrows = pathArrows.ToArray();
    currentPath.Destination = destination;
    currentPath.distancesToNextArrow =
CalculateArrowDistances();
    currentPath.Disable();
    paths.Add(currentPath);
    pathManager.Paths = paths;
    currentPath = null;
    pathAnchorNames = null;
    pathPointTypes = null;
}

```

Змінна `currentPath` — об'єкт для серіалізації шляху навігації з полями: масив назв анкерів, масиви GPS координат та кут до північного полюсу поточної сесії.

3.4.2 Модуль відображення GPS шляху

Модуль відображення GPS шляху відповідає за побудову шляху навігації за GPS координатами з десеріалізованого файлу шляху та за перехід до навігації з використання точок навігації доповненої реальності після відновлення мапи навколишнього середовища. До модулю входять класи: `GPSLoca-`

tionManager, GPSPathManager, GPSEstimator, GPSEncoder, GPSToArPathManager. Архітектуру модуля зображено на рисунку 30.

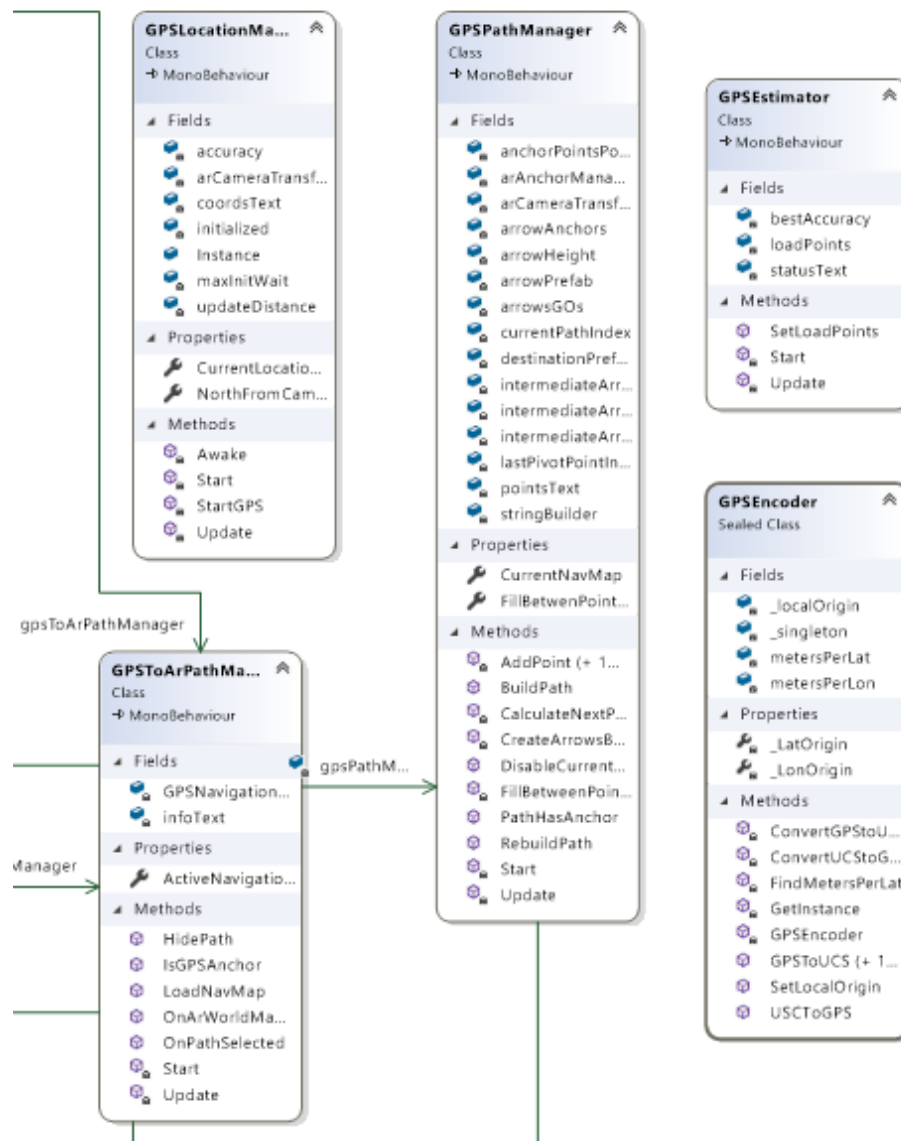


Рис. 30 Класова архітектура модулю відображення GPS шляху

Основними алгоритмами модулю відображення GPS шляху є: алгоритм додавання точки навігації до шляху та алгоритм приведення GPS координат до координатної системи доповненої реальності.

Алгоритм додавання точки навігації до шляху за GPS координатами з десеріалізованого файлу навігації наведено в лістингу 6.

Лістинг 6 Додавання точки навігації до GPS шляху

```

private void AddPoint(Vector2 pointGPS, Vector3
cameraUCS, PathPointType pointType)
{
    Vector3 pointUCS =
GPSEncoder.GPSToUCS(pointGPS);
    Vector3 pointPos = arCameraTransform.position +
(pointUCS - cameraUCS);
    pointPos.y = arrowHeight;
    float zNorthDelta =
Vector3.SignedAngle(Vector3.forward,
GPSLocationManager.Instance.NorthFromCamera, Vector3.up);
    GameObject pointGO = Instantiate(pointType ==
PathPointType.Arrow ? arrowPrefab : destinationPrefab);
    ARAnchor pointAnchor =
arAnchorManager.AddAnchor(new Pose(pointPos,
Quaternion.identity));
    pointGO.transform.parent =
pointAnchor.transform;
    pointGO.transform.position = pointPos;
    pointGO.transform.RotateAround(Vector3.zero,
Vector3.up, zNorthDelta);
    if (arrowsGOs.Count > 0)
    {
        GameObject prevPoint =
arrowsGOs[arrowsGOs.Count - 1];
prevPoint.transform.LookAt(pointGO.transform);
        pointGO.transform.forward =
prevPoint.transform.forward;
    }
    stringBuilder.Append(pointPos + " | ");
    arrowsGOs.Add(pointGO);
    arrowAnchors.Add(pointAnchor);
}

```

Аргумент `pointGPS` — GPS координати поточної точки навігації.

Аргумент `cameraUCS` — поточне GPS положення користувача в координатній системі доповненої реальності.

Аргумент `pointType` — тип поточної точки навігації (проміжна чи кінцева).

Змінна `pointUCS` — приведені до координатної системи доповненої реальності GPS координати поточної точки навігації.

Змінна `zNorthDelta` — кут між північню доповненої реальності та реальним північним полюсом.

Для оптимізації обчислення приведення GPS координат до координатної системи доповненої реальності деякі операції в формулі представлені у вигляді констант. Обчислення значень довжини GPS координат у новій координатній системі наведено в лістингу 7.

Лістинг 7 Обчислення значень довжини GPS координат

```
private void FindMetersPerLat(float lat)
{
    float m1 = 111132.92f;
    float m2 = -559.82f;
    float m3 = 1.175f;
    float m4 = -0.0023f;
    float p1 = 111412.84f;
    float p2 = -93.5f;
    float p3 = 0.118f;
    lat = lat * Mathf.Deg2Rad;
    metersPerLat = m1 + (m2 * Mathf.Cos(2 *
(float)lat)) + (m3 * Mathf.Cos(4 * (float)lat)) + (m4 *
Mathf.Cos(6 * (float)lat));
    metersPerLon = (p1 * Mathf.Cos((float)lat)) +
(p2 * Mathf.Cos(3 * (float)lat)) + (p3 * Mathf.Cos(5 *
(float)lat));
}
```

Змінна `metersPerLat` — коефіцієнт відношення GPS широти до одиниці вимірювання у координатній системі доповненої реальності.

Змінна `metersPerLon` — коефіцієнт відношення GPS довготи до одиниці вимірювання у координатній системі доповненої реальності.

3.4.3 Модуль відображення шляху доповненої реальності

Модуль відображення шляху доповненої реальності відповідає за побудову шляху маршрутизації в приміщенні, використовуючи дані про позицію точок та назви анкерів з десеріалізованого файлу шляху. До модулю входять

класи: PathManager, NavPath та Arrow. Архітектура модулю зображена на рисунку 31.

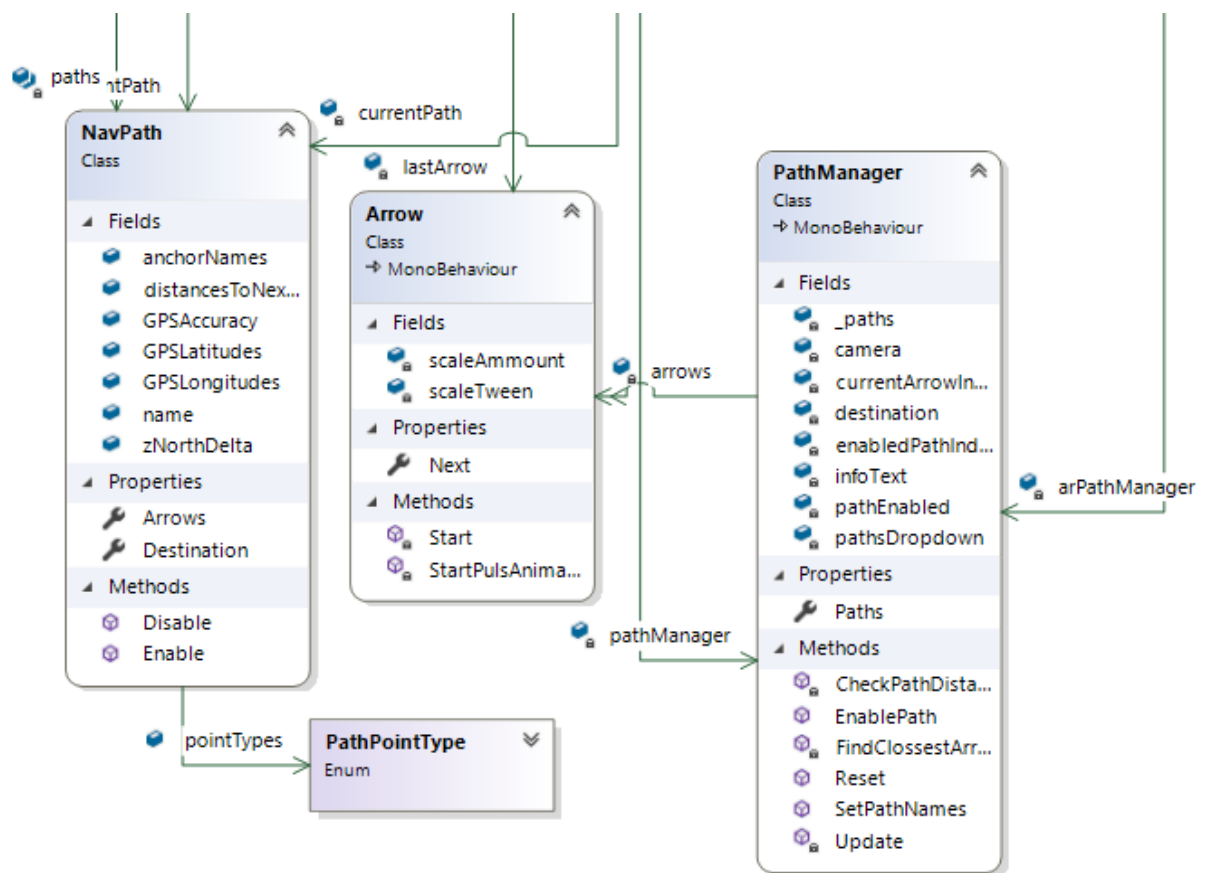


Рис. 31 Класова архітектура модулю відображення шляху доповненої реальності

Основним алгоритмом цього модулю є алгоритм відображення наступної точки навігації в маршруті, який наведено в лістингу 8.

Лістинг 8 Алгоритм відображення наступної точки навігації

```

private void CheckPathDistance()
{
    Vector3 arrowPos =
arrows[currentArrowIndex].transform.position;
    Vector3 playerPos = camera.transform.position;
    playerPos.y = arrowPos.y;
    float distanceToArrow =
Vector3.Distance(arrowPos, camera.transform.position);
    if (distanceToArrow <= 1f)

```

```

        {
            if (arrows[currentArrowIndex].Next != null)
            {
                currentArrowIndex++;
arrows[currentArrowIndex].gameObject.SetActive(true);
            }
            else
            {
                destination.SetActive(true);
                pathEnabled = false;
                infoText.SetText("Destination
reached");
            }
        }
    }
}

```

Змінна `arrows` — масив об'єктів точок навігації маршруту.

Змінна `distanceToArrow` — відстань від позиції користувача до позиції поточної точки навігації маршруту.

Змінна `destination` — посилання на об'єкт кінцевої точки маршруту.

3.4.4 Модуль взаємодії з віддаленим сервером

Модуль взаємодії з віддаленим сервером відповідає за збереження та завантаження файлів мапи навколишнього середовища та маршруту на віддалений сервер. Модуль має наступні класи: `ExploerWebApi` та `MapRemoteManager`. Архітектура модулю зображена на рисунку 32.

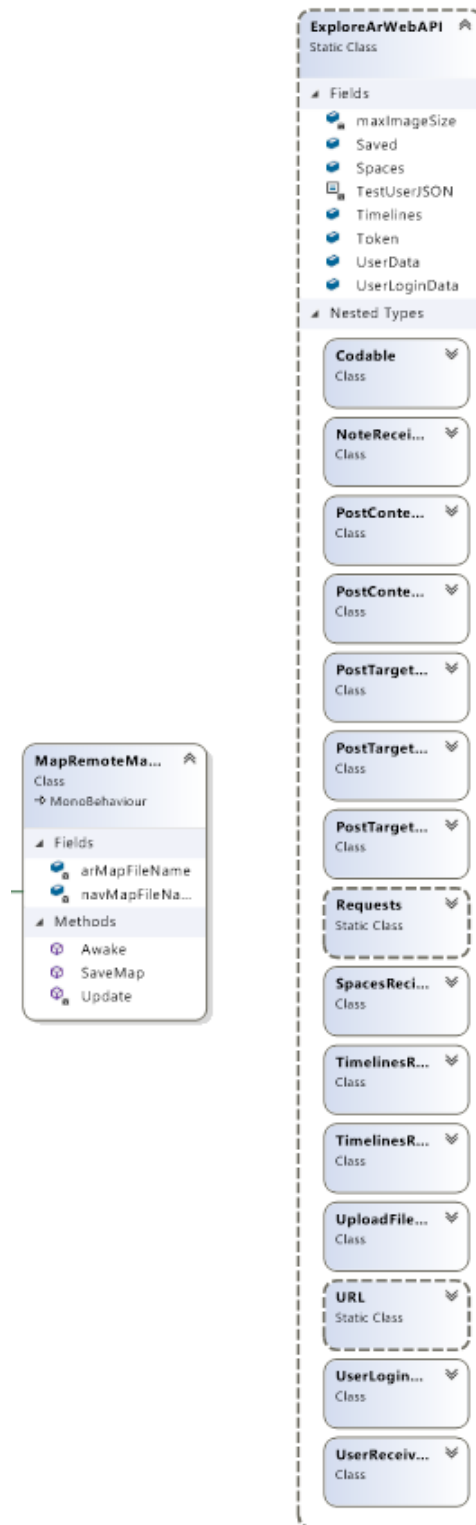


Рис. 32 Класи модулю взаємодії з віддаленим сервером

Модуль було спроектовано використовуючи реактивну парадигму програмування. Основним алгоритмом модулю є алгоритм завантаження файлу

мапи навколишнього середовища до віддаленого серверу. Реалізацію цього алгоритму наведено в лістингу 9.

Лістинг 9 Завантаження мапи навколишнього до віддаленого серверу

```

...
ExploreArWebAPI.Requests.Post.UploadFile(arMapBytes,
arMapFileName, "application/octet-stream")
    .Then(arMapRes =>
    {
        arMapResponseText = arMapRes.Text;
        byte[] navMapBytes =
File.ReadAllBytes(navMapFilePath);
        string navMapFileName =
Path.GetFileNameWithoutExtension(navMapFilePath);
        navMapFileName +=
System.DateTime.Now.ToString("MMddyyyhhmmsstt") +
Path.GetExtension(navMapFileName);
        ExploreArWebAPI.Requests.Post.UploadFile(navMapBytes,
navMapFileName, "application/json")
            .Then(navMapRes =>
            {
                var arMapData =
JsonConvert.DeserializeObject<ExploreArWebAPI.UploadFileRes
ponse>(arMapResponseText);
                var navMapData =
JsonConvert.DeserializeObject<ExploreArWebAPI.UploadFileRes
ponse>(navMapRes.Text);
                ...
                contentData.items.Add(new
ExploreArWebAPI.PostContentItem() { title = "AR Map",
content_url = arMapData.data.file, status = "completed"});
                contentData.items.Add(new
ExploreArWebAPI.PostContentItem() { title = "Nav Map",
content_url = navMapData.data.file, status = "completed"});
            });
        ExploreArWebAPI.Requests.Post.CreateContent(contentData)
    });
...

```

Змінна `arMapBytes` та змінна `navMapBytes` є масивами байтів файлу мапи навколишнього середовища та шляху навігації відповідно.

Змінна `contentData` — клас моделі даних для відправки на віддалений сервер.

3.5 Проект інтерфейсу та візуалізації маршруту

Для доступу до маршрутів навігації використовується користувацький інтерфейс клієнтського застосунку соціальної мережі “Explor”. Інтерфейс клієнтського застосунку зображено на рисунку 33.

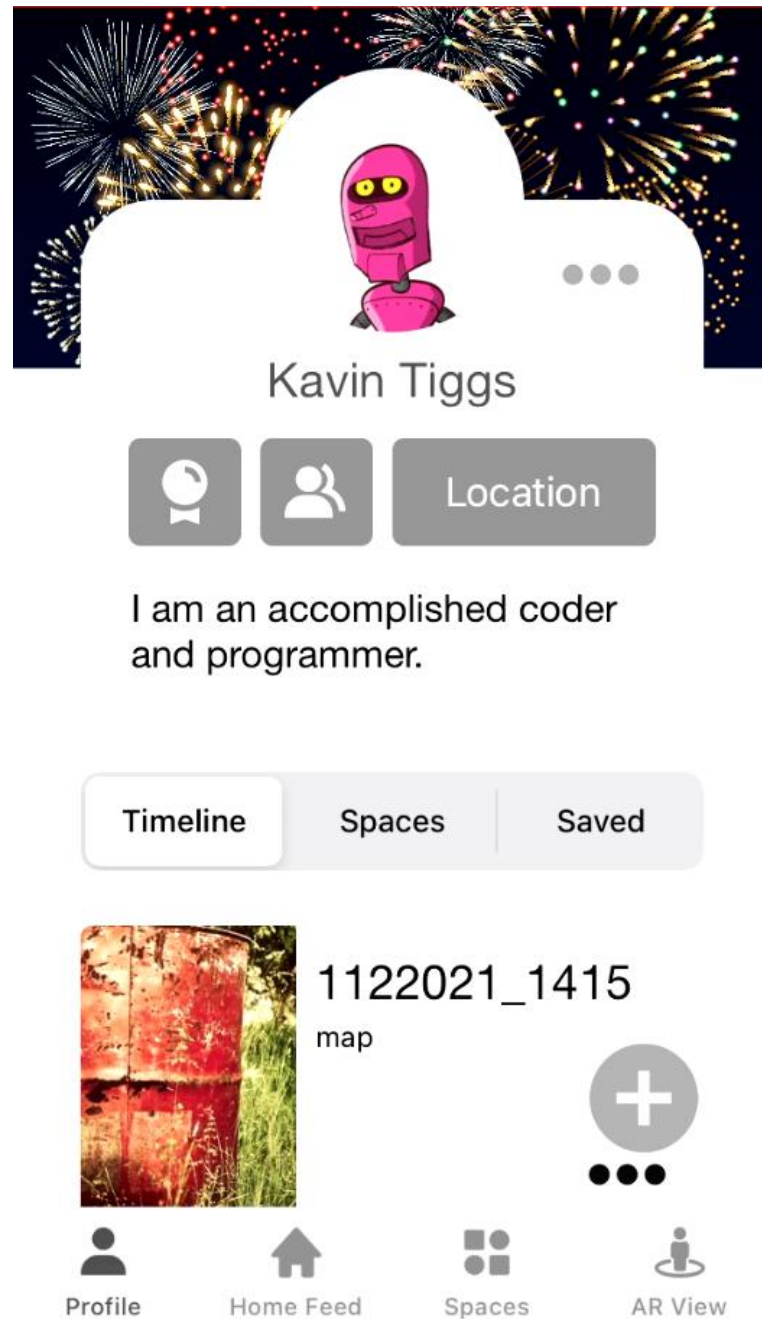


Рис. 33 Інтерфейс клієнтського застосунку

При ввімкненому режимі налаштування на екрані відображається інформація про: поточні GPS координати, назва маршруту, режим навігації (GPS чи AR) та інформація про процес відновлення шляху. Режим налаштування зображено на рисунку 34.



Рис. 34 Користувацький інтерфейс в режимі налаштування

При побудові шляху навігації користувачу надається інформація про відстань до наступного автоматичного розташування точки навігації, а також

надається можливість самостійно поставити точку на поточне місцеположення. На рисунку 35 зображено інтерфейс побудови шляху навігації.

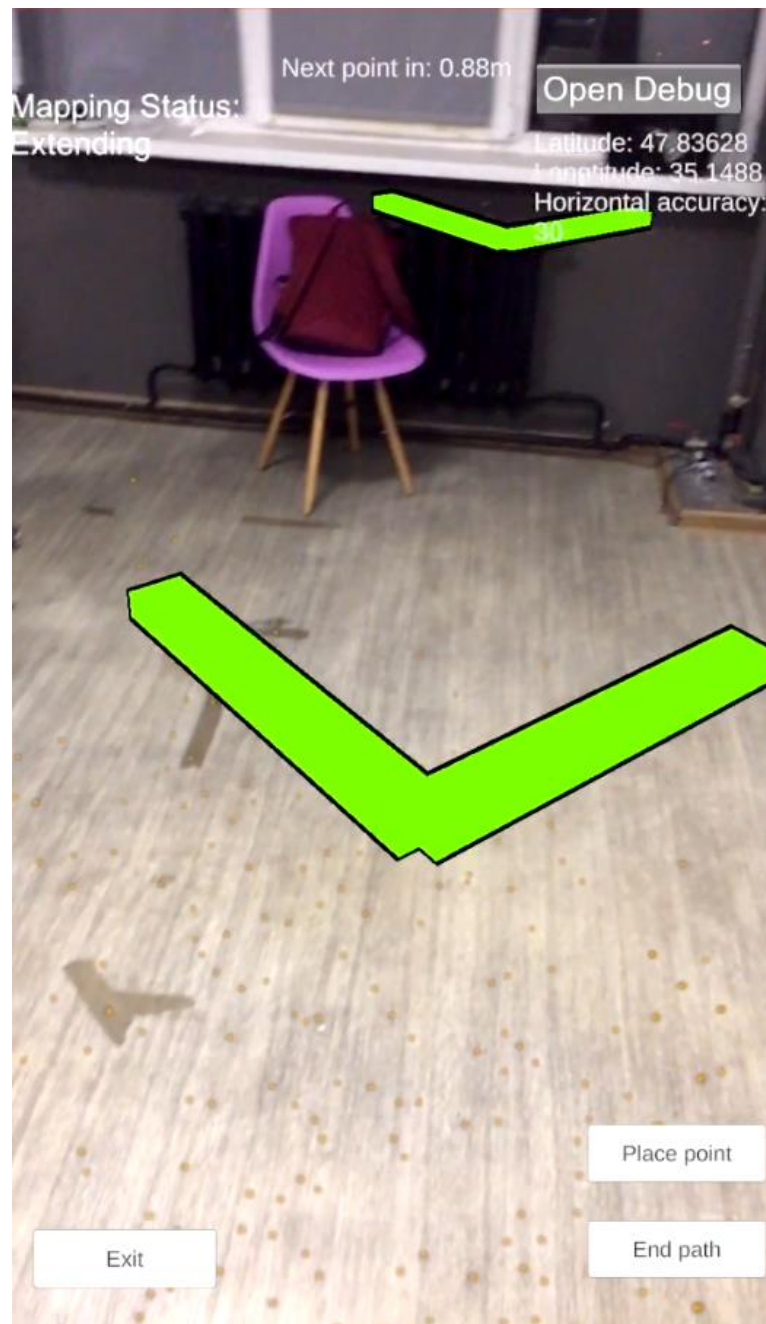


Рис. 35 Процес створення шляху навігації

Кінцева точка навігації в шляху автоматично замінюється на 3D текст, що повідомляє про досягнення цільової позиції. Кінець шляху зображено на рисунку 36.



Рис. 36 Кінець шляху навігації

Після створення шляху навігації користувач може зберегти його до свого профілю, задавши перед цим ім'я мапі. При завантаженні файлу до віддаленого серверу відображається інформація про розмір файлу та блокується можливість виходу з застосунку, як зображено на рисунку 37.

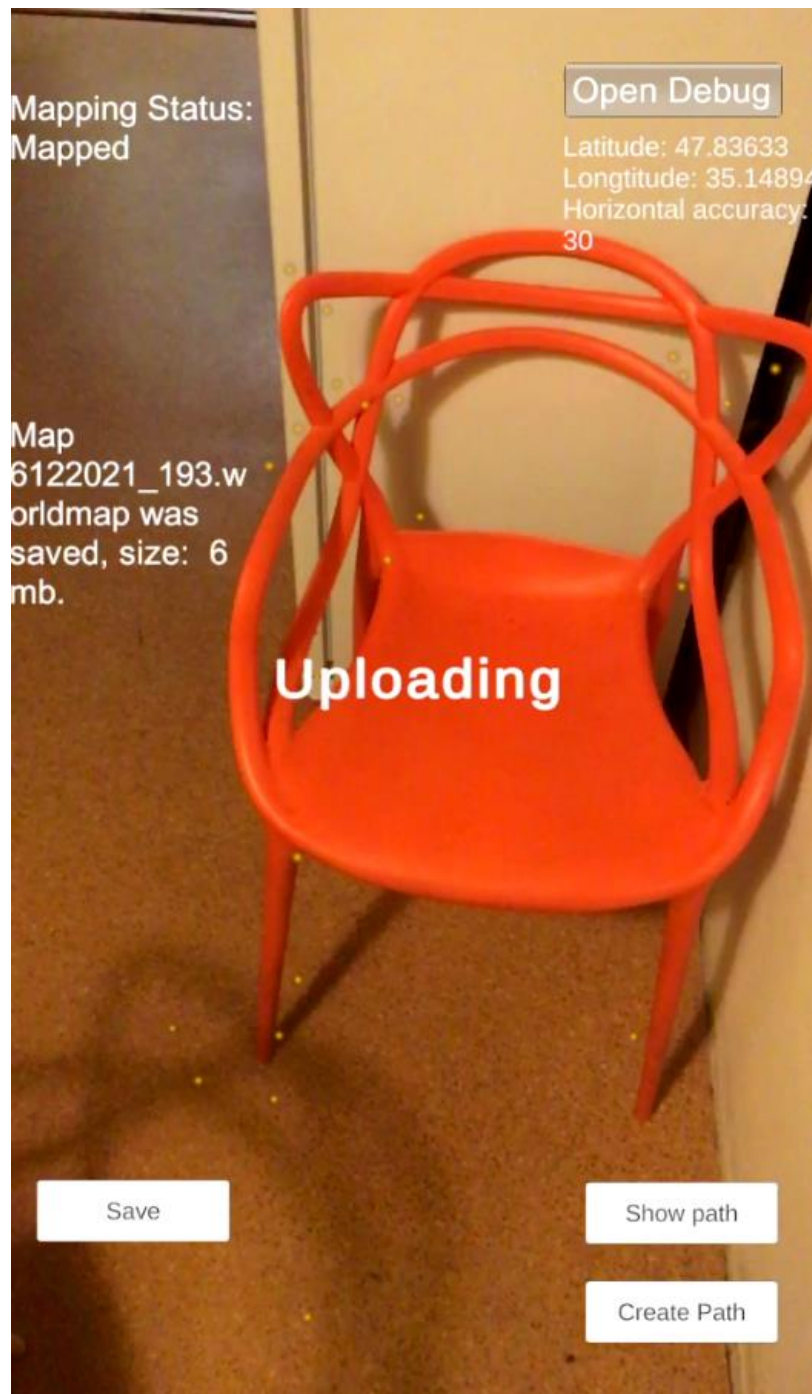


Рис. 37 Завантаження фалу мапи до віддаленого серверу

3.5 Вимоги до апаратного забезпечення

Для підтримки сумісності мап навколишнього середовища між усіма пристроями, що підтримують ARKit, було визначено наступні вимоги до апаратного забезпечення:

- Модель iPhone не старіша за SE, чи iPad Pro будь якої версії.

- iOS 15.1.1.
- Підключення до інтернету.

Саме в аспекті вимог до апаратного забезпечення проявляється одна з головних переваг розробки програмного забезпечення для iOS пристроїв — майже усі доступні девайси на цій операційній системі будуть підтримувати розроблений застосунок.

3.6 Опис функціональних можливостей

Функціональними можливостями розробленого застосунку є:

1. Створення маршруту навігації, який коректно працює в приміщенні і на вулиці.
2. Збереження маршруту навігації на віддаленому сервері.
3. Завантаження маршруту навігації з віддаленого сервера.
4. Відновлення маршруту за даними GPS
5. Безшовний перехід до маршруту побудованого за даними доповненої реальності.
6. Візуалізація точок навігації та підказок для користувача.

РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ НАВІГАЦІЇ

4.1 Аналіз швидкості та точності побудови маршрутів навігації з використанням GPS даних

Точність розташування точок навігації в маршруті з використанням GPS даних залежить від горизонтальної точності GPS. Для дослідження точності GPS даних було проведено ряд експериментів на вулиці та в приміщенні. Усі експерименти на вулиці проводилися з використанням iPhone SE та при однакових умовах, а саме: одне місцеположення, відкрита місцевість та безхмарне небо.

Результати експериментів швидкості отримання максимальної точності GPS на вулиці наведено в таблиці 2.

Таблиця 2

Швидкість досягання максимальної точності GPS на вулиці

Номер експерименту	Максимальний радіус горизонтальної точності GPS	Затрачений час на досягнення максимальної точності
1	5 метрів	12 секунд
2	5 метрів	20 секунд
3	7.5 метрів	12 секунд
4	10 метрів	7 секунд
5	5 метрів	13 секунд
6	5 метрів	10 секунд

Експерименті дослідження в приміщенні проводилися з використанням iPhone SE в одній кімнаті, результати експериментів наведено в таблиці 3.

Таблиця 3

Швидкість досягнення максимальної точності GPS в приміщенні

Номер експерименту	Максимальний радіус горизонтальної точності GPS	Затрачений час на досягнення максимальної точності
1	100 метрів	8 секунд
2	75 метрів	25 секунд
3	100 метрів	10 секунд
4	250 метрів	5 секунд
5	50 метрів	20 секунд
6	100 метрів	13 секунд

Виходячи з результатів дослідження можна зробити висновок, що використання GPS для побудови точних маршрутів навігації в приміщенні не є можливим.

4.2 Аналіз швидкості та точності побудови маршрутів з використанням доповненої реальності

Швидкість побудови маршруту навігації з використанням доповненої реальності залежить від часу затраченого на стабілізацію системи доповненої реальності після старту сесії, тобто знаходження першої площини чи групи точок орієнтації.

За характеристику точності побудови маршруту було прийнято частоту порушення стабільності мапи доповненої реальності при побудові одного маршруту довжиною в 100 метрів.

Експериментальні дослідження на вулиці проводились восени, на місцевості з дорогою та рослинністю при достатній освітленості. Результати дослідження точності та швидкості побудови маршрутів з використанням доповненої реальності на вулиці наведено в таблиці 4.

Таблиця 4

Точність та швидкість побудови маршруту в доповненій реальності на вулиці

Номер експерименту	Швидкість стабілізації доповненої реальності	Число порушень стабільності доповненої реальності
1	7 секунд	1
2	9 секунд	0
3	5 секунд	1
4	10 секунд	2
5	6 секунд	0
6	6 секунд	3

Експериментальні дослідження в приміщенні проводилися в умовах статичного освітлення та відсутності відражаючих поверхонь. Результати дослідження точності та швидкості побудови маршрутів з використанням доповненої реальності в приміщенні наведено в таблиці 4.

Таблиця 5

Точність та швидкість побудови маршруту в доповненій реальності на вулиці

Номер експерименту	Швидкість стабілізації доповненої реальності	Число порушень стабільності доповненої реальності
1	6 секунд	0
2	3 секунди	0
3	2 секунди	1
4	4 секунди	0
5	4 секунди	0
6	2 секунди	0

Виходячи з результатів дослідження можна зробити висновок, що використання доповненої реальності для побудови точних маршрутів навігації як на вулиці, так і в приміщенні, є можливим.

4.3 Аналіз швидкості та точності відновлення маршрутів в стійкій доповненій реальності

Швидкість відновлення маршрутів навігації в доповненій реальності залежить від швидкості десеріалізації файлу мапи навколишнього середовища та відновлення сесії. Після вдалого відновлення сесії доповненої реальності без помітної затримки відбувається побудова маршрутів навігації. Експериментальні дослідження швидкості відновлення маршрутів було проведено в приміщенні при таких же умовах, як при побудові. Результати експериментів наведено в таблиці 6.

Таблиця 6

Швидкість відновлення мапи навколишнього середовища

Номер експерименту	Час десеріалізації мапи навколишнього середовища	Час відновлення мапи навколишнього середовища
1	8 секунд	4 секунди
2	7 секунд	2 секунди
3	12 секунд	4 секунди
4	10 секунд	6 секунд
5	7 секунд	4 секунди
6	13 секунд	9 секунд

4.4 Порівняння якісних характеристик GPS та AR маршрутів

Технології GPS та доповненої реальності мають свої переваги і недоліки в побудові маршрутів навігації. В результаті проведеного дослідження було встановлено, що для маршрутів на вулиці більш стабільною є маршрутизація з використанням GPS, а для навігації в приміщенні — доповненої реальності. Тому доречність об'єднання цих двох технологій для створення одного застосунку для одночасної навігації в приміщенні та на вулиці можна вважати доведеною. Приклад відновленого маршруту в приміщенні за даними GPS зображено на рисунку 38, а за даними AR — на рисунку 39.

Порівняння якісних характеристик GPS та AR маршрутів наведено в таблиці 7.



Рис. 38 Відновлений шлях навігації GPS

Як можна спостерігати на рисунку 36, поточна точність GPS в 41 метр не дає змогу правильно відновити шлях навігації — точки розташовані за будівлею.



Рис. 39 Відновлений шлях навігації доповненої реальності

Шлях, зображений на рисунку 37, відновлений використовуючи десеріалізовану мапу навколишнього середовища і, на відміну від GPS, точки розташовані на правильних місцях.

Таблиця 7

Порівняння характеристик AR та GPS шляхів

Характеристика	GPS маршрут	AR маршрут
Точність положення точок навігації на вулиці	В радіусі 5 метрів	Максимально точна або відсутня
Точність положення точок навігації в приміщенні	В радіусі 50-100 метрів	Максимально точна
Швидкість побудови маршруту	Відразу	~9 секунд
Розмір сереалізованого файлу маршруту	< 1 МВ	1 – 5 МВ

ВИСНОВКИ

1. Встановлена і доведена актуальність розробки застосунку для навігації з використанням технології доповненої реальності та GPS.
2. Проведено дослідження способів створення стійкого досвіду доповненої реальності.
3. Проведені дослідження ефективності поєднання технологій доповненої реальності та GPS для одночасної навігації на вулиці та в приміщенні. Дослідження підтвердило раціональність поєднання.
4. У ході дослідження та розробки основним інструментом для реалізації системи було використано ігровий движок Unity та фреймворк доповненої реальності ARKit.
5. Розроблено програмний застосунок для смартфонів на операційній системі iOS, який було інтегровано в клієнтський застосунок мобільної соціальної мережі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Lengthy History of Augmented Reality. Huffington Post. URL: https://www.huffpost.com/entry/the-history-of-augmented-_b_9955048 (дата звернення: 23.11.2021).
2. Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented reality: a class of displays on the reality-virtuality continuum. *Proceedings the SPIE: Telemanipulator and Telepresence Technologies*, 2351, 282—292.
3. Azuma, R. T. (1997). A survey of augmented reality. *Presence-Teleoperators and Virtual Environments*, 6(4), 355—385
4. Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, Blair MacIntyre, “Recent Advances in Augmented Reality”, IEEE, November/December.
5. Regrebsubla, Namron (2015). *Determinants of Diffusion of Virtual Reality*. GRIN Publishing. p. 5. [ISBN 9783668228214](https://www.grin.com/document/view?id=9783668228214).
6. History of Mobile Augmented Reality, 2009. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.697.3463&rep=rep1&type=pdf> (дата звернення: 23.11.2021).
7. Horizon Report Johnson, Laurence F. and Smith, Rachel S. 2005 Horizon Report. Austin, TX: The New Media Consortium, 2005.
8. Real-Time Projection-Based Augmented Reality System for Dynamic Objects in the Performing Arts. URL: <https://www.mdpi.com/2073-8994/7/1/182/html> (дата звернення 23.11.2021).
9. Marker Based Augmented Reality. Anuroop Katiya , Karan Kalra and Chetan Garg URL: https://www.krishisanskriti.org/vol_image/04Jul201511074828%20%20%20%20%20%20%20%20Anuroop%20Katiyar%20%20%20%20%20%20%20%20%20%20%20%20441-445.pdf (дата звернення:23.11.2021).

10. A Survey of Online Monocular Markerless Augmented Reality
Veronica Teichrieb , Joao Paulo Silva do Monte Lim, Eduardo Lourenc,o
Apolinario, Thiago Souto Maior Cordeiro de Farias Marcio Augusto Silva Bueno ,
Judith Kelner, and Ismael H. F. Santos. URL:
[http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.329.1187&rep=rep1&ty
pe=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.329.1187&rep=rep1&type=pdf) (дата звернення: 23.11.2021).

11. Technologies for Augmented Reality: Calibration for Real-Time
Superimposition on Rigid and Simple-Deformable Real Objects Yann Argotti,
Valerie Outters, Larry Davis¹, Ami Sun, and Jannick P. Rolland. URL:
https://link.springer.com/content/pdf/10.1007/3-540-45468-3_81.pdf (дата звер-
нення: 23.11.2021).

12. Mekni, Mehdi, and Andre Lemieux, "Augmented reality:
Applications, challenges and future trends.", Applied Computational Science—
Proceedings of the 13 th International Conference on Applied Computer and
Applied Computational Science (ACACOS '14) Kuala Lumpur, Malaysia. 2014:
pp. 205, 207 – 209.

13. Tang, S. L., Kwoh, C. K., Teo, M. Y., Sing, N. W., & Ling, K. V.,
(1998). "Augmented reality systems for medical applications", IEEE engineering
in medicine and biology magazine, 17(3): pp. 51.

14. AR & Entertainment: A Combination poised to disrupt the future.
URL: <https://augray.com/blog/augmented-reality-entertainment/> (дата звернення:
23.11.2021).

15. Usage of Augmented Reality in Education. URL:
[https://medium.com/inbornexperience/usage-of-augmented-reality-in-education-
be783e0159a](https://medium.com/inbornexperience/usage-of-augmented-reality-in-education-be783e0159a) (дата звернення: 24.11.2021).

16. Augmented reality navigation systems. Wolfgang Narzt, Gustav
Pomberger, AloisFerscha, Dieter Kolb, Reiner Muller, Jan Wieghardt, Horst
Hortner, Christopher Lindinger. URL:
[https://www.researchgate.net/publication/220606626_Augmented_reality_navigati
on_systems](https://www.researchgate.net/publication/220606626_Augmented_reality_navigation_systems) (дата звернення: 24.11.2021).

17. M. Golfarelli, D. Maio, and S. Rizzi. Elastic correction of dead-reckoning errors in map building. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98), pages 905–911, Victoria, Canada, 1998.
18. S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robot. *Machine Learning and Autonomous Robots*, 31(5):1 – 25, 1998.
19. J. J. Leonard and H. J. S. Feder. Decoupled stochastic mapping. Technical report, MIT, 12 1999.
20. P. C. R. Smith, M. Self. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167 – 193, 1988.
21. W. H. Press, S. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes, Second Edition*. Cambridge University Press, 1992.
22. Cloud Anchors developer guide for Android. URL: <https://developers.google.com/ar/develop/java/cloud-anchors/developer-guide> (дата звернення 05.12.2021).
23. ARKIT ARWorldMap Documentation. URL: <https://developer.apple.com/documentation/arkit/arworldmap> (дата звернення 05.12.21).
24. J. Santana-Fernandez, J. G ´omez-Gil, and L. del Pozo-San-Cirilo. Design and implementation of a gps guidance system for agricultural tractors using augmented reality technology. *Sensors*, 10(11):10435– 10447, 2010.
25. A Tour of the C# language. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (дата звернення 07.12.2021).
26. Apple Augmented Reality with ARKit 5. URL: <https://developer.apple.com/augmented-reality/arkit/> (дата звернення 08.12.2021).
27. Apple Xcode 13. URL: <https://developer.apple.com/xcode/> (дата звернення 08.12.2021).

28. G. R. King, W. Piekarski, and B. H. Thomas. Arvino outdoor augmented reality visualisation of viticulture gis data. In Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 52–55. IEEE Computer Society, 2005.
29. K.-L. Ma. In situ visualization at extreme scale: Challenges and opportunities. IEEE Computer Graphics and Applications, 29(6):14–19, 2009.
30. About iOS 15 updates. URL: <https://support.apple.com/en-us/HT212788> (дата звернення 08.12.2021).
31. Воропаєв С.О., Полякова Н.П. Навігація з використанням технологій доповненої реальності та GPS: Матеріали I Всеукраїнської науково-практичної конференції здобувачів вищої освіти, аспірантів та молодих вчених «Актуальні питання сталого науково-технічного та соціально-економічного розвитку регіонів України» 19-21 жовтня 2021 р. Запоріжжя : ЗНУ, 2021. С. 342–343.
32. Воропаєв С.О., магістрант, Полякова Н.П., доцент — науковий керівник. Навігація з використанням технологій доповненої реальності та GPS. *Молода наука-2021* : зб. наук. праць студентів, аспірантів і молодих вчених. Запоріжжя : ЗНУ, 2021. Т. 5. С. 96–97.

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Воропаєв Сергій Олегович, студент 2 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти se216-02@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему «Навігація з використання технологій доповненої реальності та GPS» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії

згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2021 Підпис  Воропаєв Сергій Олегович
(студент)

Дата 30.11.2021 Підпис  Полякова Наталія Петрівна
(науковий керівник)