

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних технологій

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Розробка застосунку для навігації по ОДЕКУ з  
використанням технологій доповненої реальності

Виконала студентка 2 курсу групи МІС-22  
спеціальності 122 Комп'ютерні науки  
Теліженко Анастасія Олегівна

Керівник к.г.н., доцент,  
Кузніченко Світлана Дмитрівна

Рецензент к.т.н., доцент,  
Гнатовська Анна Арнольдівна

Одеса 2023

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	9
ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ АНАЛОГІВ .....	12
1.1 Аналіз предметної області .....	12
1.2 Аналіз існуючих програмних систем .....	15
2 ОБҐРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ.....	19
2.1 Деталізація задач проекту.....	19
2.2. Вибір середовища розробки .....	20
2.3 Вибір інструментів для реалізації внутрішньої AR-навігації .....	22
3. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ AR-НАВІГАЦІЇ.....	29
3.1 Технологія доповненої реальності .....	29
3.2 Поняття технологій AR-навігації .....	31
3.3 Аналіз різних типів внутрішньої навігації.....	33
3.4 Задача пошуку шляху.....	36
4. ПРОЕКТУВАННЯ ЗАСТОСУНКУ .....	42
4.1 Проектування варіантів використання системи .....	42
4.2 Створення діаграми діяльності.....	43
4.3 Створення діаграми послідовностей .....	45
5. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ «НАВІГАТОР ПО ОДЕКУ» .....	47
5.1 Інтеграція Vuforia Engine з Unity.....	47
5.2 Підключення платформи Firebase .....	56
5.3 Реалізація навігації засобами Unity .....	61

	8
5.4 Опис та тестування застосунку .....	64
ВИСНОВКИ .....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	70

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ADT	–	Android Development Tools, інструменти для розробки для Android
API	–	Application Programming Interface, прикладний програмний інтерфейс
AR	–	Augmented Reality, доповнена реальність
BFS	–	Breadth-First Search, алгоритм пошуку у ширину
CASE	–	Computer-Aided Software Engineering, програмна інженерія для проектування програмного забезпечення
DFS	–	Depth-First Search, алгоритм пошуку в глибину
GPS	–	Global Positioning System, система глобального позиціонування
IDE	–	Integrated Development Environment, інтегроване середовище розробки
IPS	–	Indoor Positioning System, система внутрішнього позиціонування
JSON	–	JavaScript Object Notation, запис об'єктів JavaScript
NoSQL	–	Not Only SQL, не тільки SQL, нереляційна база даних
QR-код	–	Quick Response Code, код швидкої відповіді
SDK	–	Software Development Kit, набір засобів розробки
SLAM	–	Simultaneous Localization And Mapping, технології одночасної локалізації і картографування
SQL	–	Structured Query Language, структурована мова запитів
UML	–	Unified Modeling Language, уніфікована мова моделювання
UWB	–	Ultra-Wide Band, ультраширокий діапазон
ОДЕКУ	–	Одеський державний екологічний університет
ОС	–	Операційна система
ООН	–	Організація Об'єднаних Націй
ХПІ	–	Харківський політехнічний інститут

## ВСТУП

Незважаючи на складні часи, пов'язані з війною та нестабільністю, вища освіта залишається фундаментальною і навіть може набувати ще більшого значення. Освіта залишається ключем до особистого та професійного розвитку, а також до виходу зі складних ситуацій. Університети відіграють важливу роль, надаючи студентам знання, ресурси та підтримку, необхідні для подолання труднощів та досягнення успіху. В Україні студенти також продовжують прагнути до вступу до університетів.

При вступі до університету одним з складних завдань для студентів може стати орієнтація у корпусах та пошук потрібних аудиторій. Це особливо актуально для першокурсників, які не знайомі із будовою корпусів університету. У таких випадках розробка застосунку для навігації стає необхідною та дуже затребуваною.

Подібний застосунок може значно полегшити життя студентів, надаючи їм інтуїтивно зрозумілий інтерфейс для пошуку потрібних місць та аудиторій на карті університетського комплексу. Такий застосунок не тільки допоможе студентам швидше знаходити потрібні аудиторії, але й заощадить їх час, зменшивши ймовірність запізнь на заняття. Крім того, він може сприяти більш ефективному використанню ресурсів університету та підвищенню задоволеності студентів від університетського середовища.

Метою кваліфікаційної роботи є розробка мобільного застосунку «Навігація по ОДЕКУ» для локалізації користувачів на території Одеського державного екологічного університету та побудови маршруту до необхідної аудиторії і навігації з використанням технологій доповненої реальності. Для досягнення поставленої мети були сформульовані наступні завдання:

- провести аналіз предметної області щодо внутрішньої навігації;
- провести порівняльний аналіз існуючих програм-аналогів, які використовують технології доповненої реальності для навігації по закладах освіти;

- обґрунтувати вибір програмних засобів розробки та технологій;
- провести проектування системи з використанням мови UML;
- виконати реалізацію застосунку;
- підготувати інструкцію користувача;
- виконати тестування застосунку.

Структура кваліфікаційної роботи магістра складається з вступу, 5 розділів, висновків, переліку посилань на найменувань. Повний обсяг проекту становить 71 сторінку, містить 39 рисунків.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ АНАЛОГІВ

## 1.1 Аналіз предметної області

Навігація та позиціонування всередині приміщень – це технологія відстеження розташування в режимі реального часу, визначення розташування людей та маршрутизації.

Для ефективної роботи працівники сфери обслуговування, співробітники служби безпеки, офісні працівники та відвідувачі повинні легко орієнтуватися при переміщенні до пунктів призначення. На великих майданчиках це може бути складно, якщо місцями призначення є кімнати, і майже неможливо, коли місцями призначення є активи, такі як принтери, водяні крани або вогнегасники. Коли персонал переміщається у робочому просторі, у якому ніколи раніше не був, навігація всередині приміщень допомагає швидко визначити місцезнаходження.

Навігація та позиціонування всередині приміщень допомагають співробітникам та відвідувачам орієнтуватися через мобільні програми при пересуванні в якесь приміщення або на вулиці.

Навігація всередині приміщень – це ідентифікація розташування користувача вздовж маршруту по будівлі до вказаного місця призначення, наприклад, до кімнати, поверху або об'єкта. У сумісних мобільних програмах користувачі можуть визначати своє місцезнаходження з точністю до декількох метрів на картах приміщень, знаходити маршрути по всьому об'єкту, а також отримувати в режимі реального часу точні вказівки для переміщення до обраного пункту призначення. Навігація всередині приміщень працює з плануванням будівлі, щоб включати в маршрут такі елементи, як сходи, поверхи та кімнати.

На відміну від зовнішнього світу, внутрішній світ розширюється. У Звіті ООН про глобальний стан за 2017 рік сказано: «До 2060 року площа житлового сектора подвоїться, додавши до планети більше ніж 230 мільярдів квадратних

метрів шляхом будівництва нових будівель. Ці доповнення еквівалентні додаванню площі Японії щороку з цього моменту до 2060 року» [1].

Це величезне зростання внутрішнього простору ставить такі проблеми, як оптимальне використання простору, ефективніше споживання енергії та взаємодія з навколишнім середовищем.

Технологічний розвиток дає змогу перетворювати будівлі на розумні будівлі та створювати внутрішні простори, які допомагають організаціям не лише підвищити ефективність, але й покращити самопочуття на робочому місці.

Система внутрішнього позиціонування (англ. indoor positioning system, IPS) дозволяє користувачам знаходити себе та інших усередині будівлі в режимі реального часу [2].

Через перешкоди прямої видимості, спричинені стінами будівель та іншими перешкодами, сигнали GPS недостатньо потужні, щоб надати достатньо якісні результати позиціонування всередині будівель.

Ефективна навігація в приміщенні покладається на комбінацію технологій. Маяки Bluetooth, стратегічно розміщені всередині будівлі, обмінюються даними зі смартфонами та іншими пристроями користувачів, щоб надавати інформацію про місцезнаходження в реальному часі. Системи позиціонування Wi-Fi використовують існуючі мережі Wi-Fi для триангуляції позиції користувача, що підвищує точність. Інерційні датчики, зокрема акселерометри та гіроскопи, відстежують рухи користувачів, допомагаючи подолати прогалини в охопленні сигналу. Крім того, технології комп'ютерного зору можна використовувати для створення динамічних карт внутрішніх приміщень, які допомагають у точній навігації.

Застосування внутрішньої навігації різноманітне. У комерційних приміщеннях він сприяє ефективному потоку клієнтів, допомагає знайти продукти та навіть може пропонувати персоналізовані рекламні акції залежно від місця розташування користувача. У сфері охорони здоров'я це допомагає пацієнтам



знаходити конкретні відділення чи установи в лікарнях. Музеї та галереї використовують навігацію в приміщенні, щоб покращити враження від відвідувачів, пропонуючи інтерактивні виставки та екскурсії.

Університети з розгалуженими кампусами та численними будівлями чи просто великими площами будівель використовують технологію внутрішньої навігації, щоб покращити загальний досвід для студентів, викладачів і відвідувачів.

Навігаційні програми для приміщень допомагають студентам і відвідувачам легко знаходити аудиторії, бібліотеки, офіси та інші приміщення. Інтерактивні карти та функції доповненої реальності забезпечують візуальний посібник, роблячи орієнтацію більш інтуїтивно зрозумілою. Також навігація в приміщенні допомагає учасникам знаходити місця проведення, семінарські зали, туалети, покращуючи загальне враження від події. А під час надзвичайних ситуацій, таких як пожежа, стихійне лихо чи повітряна тривога, навігаційні системи в приміщеннях швидко направляють студентів і персонал до найближчих виходів або безпечних зон, забезпечуючи їх безпеку.

До того, університети навіть можуть оптимізувати використання свого простору, аналізуючи дані, зібрані за допомогою внутрішніх навігаційних систем. Аналіз моделей пішохідного руху може сприяти прийняттю рішень, пов'язаних із використанням будівлі, допомагаючи ефективному розподілу ресурсів.

Для внутрішньої навігації можна використовувати різні технології, зокрема:

- Wi-Fi: сигнали Wi-Fi можна використовувати для визначення місцезнаходження людини шляхом вимірювання потужності сигналу від різних точок доступу Wi-Fi.
- Bluetooth: маячки Bluetooth можна розмістити по всій будівлі, щоб надавати точнішу інформацію про місцезнаходження.

- Надширококутний діапазон: UWB – це високоточна технологія малого радіусу дії, яка може використовуватися для точного визначення місцезнаходження людини відносно інших пристроїв UWB.
- Комп'ютерний зір. Комп'ютерний зір можна використовувати для визначення орієнтирів у будівлі та відстеження руху людини за допомогою камер.

## 1.2 Аналіз існуючих програмних систем

В університетах всього світу використовуються різні технології внутрішньої навігації для підвищення ефективності, доступності та безпеки приміщень закладів. В даній роботі будуть розглянуті варіанти використання навігації саме в закладах вищої освіти України.

Віртуальний університет – веб-сайт для навігації по Міжнародному економіко-гуманітарному університету імені академіка Степана Дем'янука (рис. 1.1). [3]

Основні функції: детальні плани поверхів корпусів університету з підписами аудиторій та інших приміщень, доступ з пристрою з будь-якою ОС.

Недоліки: відсутнє позиціонування користувача та можливість навігації у реальному часі.



Рисунок 1.1 – Скріншоти сайту «Віртуальний університет»

KPI.maps – це інформаційний застосунок Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (рис. 1.2).

Основні функції: пошук потрібного корпусу, гуртожитку, банкомату або будь-якого іншого об'єкта за його назвою чи адресою, перегляд повної мапи університету, детальний план поверху потрібного корпусу чи гуртожитку, інформація про аудиторії: наявність Wi-Fi, розеток, проектора тощо [4].

Недоліки: доступний тільки на iPhone, навігація тільки за допомогою GPS, не передбачена побудова маршруту всередині будівлі.

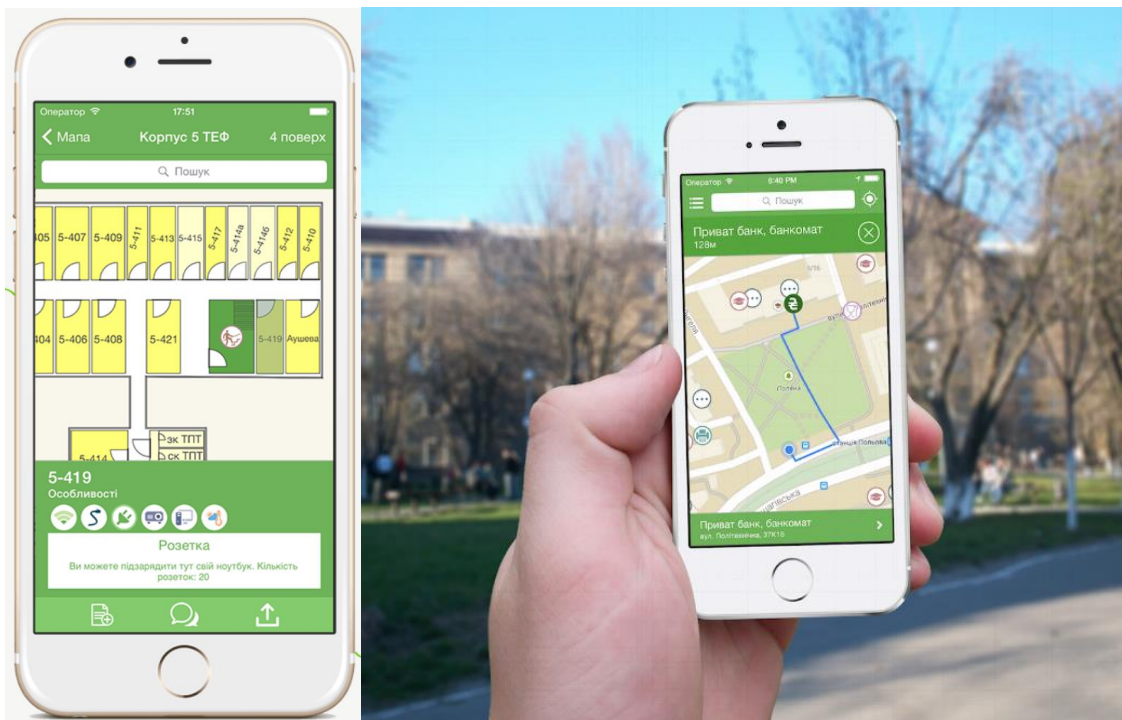


Рисунок 1.2 – Скріншоти застосунку «KPI.maps»

«AR CAMPUS\_KhPI» – застосунок для навігації по території Національного технічного університету «Харківський політехнічний інститут» (рис. 1.3).

Основні функції: перегляд 3D-карти території ХПІ, пошук потрібного навчального корпусу та отримання на карті найкоротшого і найкращого маршруту до нього в режимі AR.

Недоліки: відсутність навігації всередині корпусів.

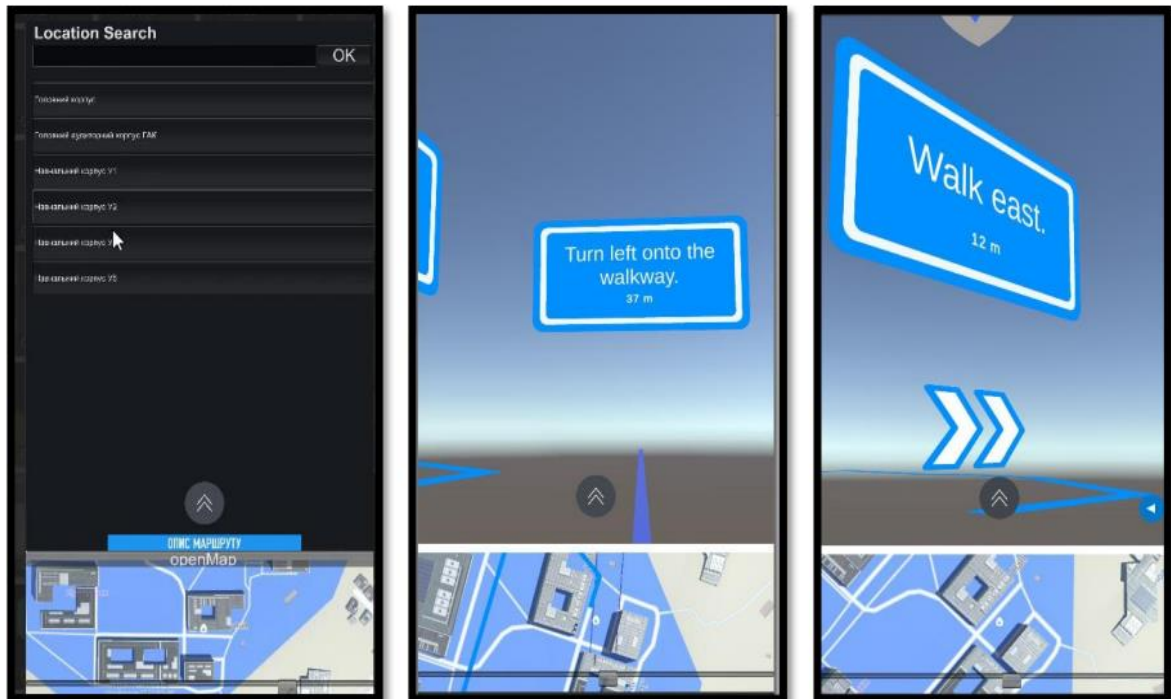


Рисунок 1.3 – Скріншоти застосунку AR CAMPUS\_KhPI

В результаті аналізу існуючих програмних систем було виявлено, що в Україні у відкритому доступі відсутні застосунки для внутрішньої навігації всередині приміщень університетів з функціями позиціонування користувача та використання технологій доповненої реальності. Тому було також досліджено досвід закордонних університетів.

Застосунок для навігації по приватному університету Санвей (Малайзія) дозволяє користувачам прокладати маршрути всередині приміщень університету та надає навігацію за допомогою AR-технологій (рис. 1.4).

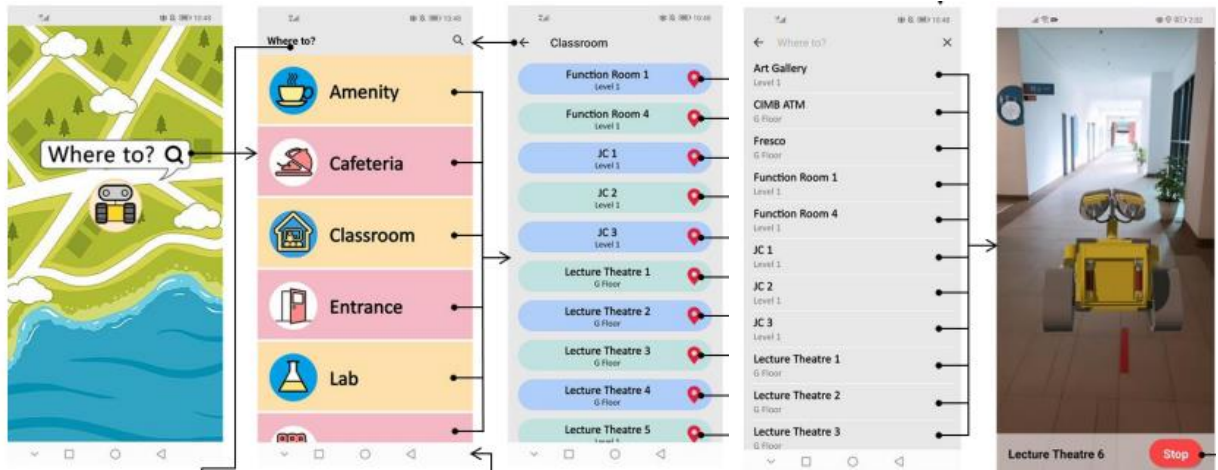


Рисунок 1.4 – Скріншоти застосунку для навігації по університету Санвей

У застосунку можна обрати пункт призначення з загального списку та у режимі доповненої реальності можна рухатись по побудованому маршруту [5].

Таким чином, під час виконання першого розділу магістерської роботи була обрана предметна область, в рамках якої розглянуті програмні аналоги і їх особливості. Переваги та недоліки розглянутих систем були взяті до уваги при розробці застосунку для AR-навігації по Одеському державному екологічному університету.

## 2 ОБҐРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ

### 2.1 Деталізація задач проекту

Метою кваліфікаційної роботи є розробка застосунку для навігації по Одеському державному екологічному університету з використанням технологій доповненої реальності. Даний застосунок буде корисним для студентів, особливо, першокурсників, викладачів та інших працівників і відвідувачів університету.

Для досягнення поставленої мети в роботі були сформульовані наступні завдання:

- провести аналіз предметної області використання систем внутрішньої навігації в різних організаціях та університетах;
- провести порівняльний аналіз існуючих аналогів;
- обґрунтувати вибір програмних засобів розробки та технологій;
- провести проектування системи;
- виконати програмну реалізацію застосунку;
- підготувати інструкцію користувача;
- виконати тестування розгорнутого проекту.

Після проведення детального аналізу предметної області наступними етапами є проектування та програмна реалізація застосунку. Після того, як буде виконана розробка застосунку для AR-навігації, буде проведено тестування. Недоліки, виявлені на етапі тестування, мають бути проаналізовані та виправлені.

Перед початком розробки застосунку були визначені функціональні та нефункціональні вимоги.

До функціональних вимог належать:

- відображення детальної мапи університету;
- позиціонування користувача в університеті;
- прокладення маршруту до обраної аудиторії;
- навігація в AR-режимі.

Нефункціональними вимогами до застосунку є надійність; адекватна реакція в разі збоїв в роботі, збереження всієї інформації в базі даних, така побудова системи, при якій буде легко створювати нові модулі та функції або виправляти код з метою покращення.

## 2.2. Вибір середовища розробки

Популярними середовищами розробки під ОС Android є Eclipse, Android Studio, Unity.

Eclipse – це інтегроване середовище розробки (IDE), яке використовується для програмування на різних мовах програмування, включаючи Java, C++, PHP, Python та інші. Його також можна використовувати для мобільної розробки, зокрема для розробки додатків для платформи Android.

Eclipse доступний безкоштовно, що робить його привабливим вибором для багатьох розробників. Його можна розширювати за допомогою плагінів для підтримки різних мов програмування та інструментів. Eclipse має потужний інтерфейс з багатьма функціями редагування, відлагодження, керування версіями, а також вбудованими інструментами для роботи з Java та іншими мовами програмування. З використанням Android Development Tools (ADT) плагіна, Eclipse може бути використаний для створення додатків для платформи Android.

Однак остання офіційна версія Eclipse (Eclipse IDE for Java Developers) була випущена досить давно, що може вплинути на доступність нових функцій та інструментів для мобільної розробки. Залежно від розміру проекту та використаних плагінів, Eclipse може бути вимогливим до ресурсів комп'ютера, що може вплинути на продуктивність при розробці мобільних додатків. Хоча ADT був популярним плагіном для розробки Android-додатків у Eclipse, офіційна підтримка цього плагіна припинена на користь Android Studio, який став основним інструментом для розробки Android.

Android Studio – це офіційне інтегроване середовище розробки (IDE) від Google для створення додатків для платформи Android. Це потужний інструмент, спеціально розроблений для розробки мобільних додатків.

Android Studio є офіційним інструментом для розробки Android, що означає швидше отримання оновлень та підтримку нових функцій. Має розширені інструменти для редагування коду, відлагодження, профілювання та інші, що спрощує процес розробки, надає зручний інструментарій для відлагодження та тестування додатків на емуляторах та реальних пристроях. Android Studio має вбудовану підтримку мови програмування Kotlin, що робить розробку більш продуктивною. Офіційна підтримка Android SDK та доступ до різноманітних Google Services спрощує розробку функціональних та інноваційних додатків.

Однак Android Studio не орієнтоване на розробку застосунків з використанням технології доповненої реальності. Багато платформ для роботи з AR (такі як HoloLens, Oculus, Vuforia тощо) призначені для Unity.

Unity – це популярне інтегроване середовище розробки (IDE), яке використовується для створення ігор, віртуальної реальності, розширеної реальності та інтерактивних додатків. Воно підтримує різні платформи, включаючи iOS, Android, Windows, MacOS, консолі для ігор та інші.

Unity має потужний графічний движок, який дозволяє створювати високоякісні графічні ефекти та візуально привабливі ігри та додатки. Unity дозволяє розробляти додатки для різних платформ, використовуючи єдиний код, що спрощує процес розробки та підтримки для різних пристроїв. У ньому використовуються скриптові мови, такі як C# та JavaScript, для програмування логіки гри та взаємодії об'єктів. Unity має вбудовані інструменти для відлагодження та налагодження програм, що допомагає виявляти та виправляти помилки під час розробки. Unity підтримує імпорт та обробку різних типів ресурсів, таких як графіка, звуки, моделі, що робить розробку більш гнучкою.



Unity дозволяє створювати різні типи додатків, не тільки ігри, а також віртуальну та доповнену реальність. Має велику спільноту користувачів та обширну документацію, що полегшує навчання та розвиток в ньому.

Серед недоліків Unity для мобільної розробки можна назвати ресурсомісткість, адже робота з Unity може вимагати потужний обладнання для оптимальної роботи, особливо під час розробки складних графічних додатків та розмір додатків, бо додатки, створені в Unity, можуть бути більшими за розмір, через включення додаткових бібліотек і функцій.

Таким чином, за підсумком аналізу існуючих середовищ розробки, було обрано середовище Unity, тому що воно є крос-платформним, найбільше підходить під потреби застосунку, що розроблюється, та має багато навчальної інформації і форум для підтримки.

### **2.3 Вибір інструментів для реалізації внутрішньої AR-навігації**

Існують спеціальні інструменти для розробки застосунків з використанням технології доповненої реальності та AR-навігації. Серед основних – AR Kit, AR Core, Vuforia Engine.

ARKit – це набір інструментів для розробки застосунків доповненої реальності (AR), розроблений компанією Apple для пристроїв з операційною системою iOS (рис. 2.1). Представлений у вересні 2017 року, ARKit надає розробникам широкі можливості створення інноваційних та захоплюючих AR-додатків для iPhone та iPad.



Рисунок 2.1 – Apple ARKit

Можливості ARKit [6]:

Горизонтальний та вертикальний трекінг поверхонь. ARKit дозволяє взаємодіяти з плоскими та вертикальними поверхнями, що робить можливим розміщення віртуальних об'єктів на стінах, столах тощо.

Розпізнавання обличчя. ARKit підтримує розпізнавання обличчя, що дозволяє створювати AR-додатки з функціями обробки обличчя.

Анімація AR-об'єктів. Розробники можуть створювати анімовані віртуальні об'єкти та інтерактивні сцени для користувача.

Локалізація та трекінг рухів. ARKit відслідковує рухи та локалізує користувача в просторі, що дозволяє створювати інтерактивні AR-застосунки, які реагують на рухи користувача.

Виявлення світла. Застосунки можуть адаптувати свою графіку до рівня освітлення навколишнього середовища для більш реалістичного відображення.

Узгодження між користувачами (Multiuser AR). ARKit 2.0 дозволяє спільно взаємодіяти з однією і тією ж AR-сценою на декількох пристроях одночасно.

Серед переваг SDK ARKit:

Широке поширення пристроїв, що підтримуються. Значна кількість користувачів iPhone та iPad забезпечує велику аудиторію для AR-додатків, створених за допомогою ARKit.

Інтеграція з екосистемою Apple. ARKit інтегрований з іншими технологіями Apple, такими як CoreML та Metal, що спрощує розробку комплексних AR-додатків.

Інтуїтивність і легкість використання. Інструменти ARKit створені з урахуванням зручності використання для розробників, що сприяє швидкому впровадженню інноваційних ідей.

Підтримка від Apple. Активна підтримка та оновлення зі сторони Apple, що забезпечує надійність та безпеку додатків.

Недоліки ARKit:

Обмежена сумісність. ARKit підтримується лише на пристроях з високоефективними процесорами, що може обмежити використання на старіших пристроях.

Обмежена кількість функцій на старших версіях iOS. Деякі функції ARKit можуть бути обмежені на старших версіях iOS, що може обмежувати можливості розробників.

Недоступність для Android. ARKit підтримується лише на пристроях iPhone та iPad, що відкидає велику кількість користувачів ОС Android.

AR Core – це платформа доповненої реальності від Google, яка розроблена для сумісних з Android пристроїв (рис. 2.2). Вона надає розробникам інструменти для створення додатків, які можуть взаємодіяти з реальним світом через віртуальні об'єкти та інформацію [7].



Рисунок 2.2 – Google AR Core

Основні можливості AR Core для розробників:

Трекінг поверхонь. AR Core дозволяє відслідковувати поверхні в реальному часі, надаючи можливість віртуальним об'єктам точно розміщуватися на фізичних об'єктах, таких як столи або підлога.

Розпізнавання об'єктів. Платформа може розпізнавати реальні об'єкти, що дозволяє створювати додатки з розширеною взаємодією з ними.

Керування освітленням. AR Core може аналізувати оточуюче освітлення і взаємодіяти з віртуальними об'єктами, враховуючи освітлення в реальному часі.

Виявлення руху. Застосунки можуть реагувати на рухи користувача, що дозволяє створювати ігри та додатки з інтерактивними елементами.

Виявлення глибини. AR Core вміє вимірювати відстань до об'єктів у кадрі, що розширює можливості взаємодії з реальним світом.

Хмарні маркери. Ця можливість дозволяє користувачам спільно взаємодіяти з одними й тими ж віртуальними об'єктами в реальному часі через декілька пристроїв, використовуючи хмарні сервіси.

Серед переваг AR Core можна виділити:

Широке охоплення. AR Core підтримує багато пристроїв на базі Android, що розширює потенційну аудиторію додатків.

Розвинена функціональність. Платформа надає розробникам багатий набір інструментів для створення високоякісних AR-додатків, включаючи трекинг поверхонь і розпізнавання об'єктів.

Підтримка Google Services. Інтеграція з Google сервісами, такими як Google Maps, спрощує створення додатків для навігації та маркування в реальному світі.

Серед недоліків AR Core можна назвати обмежену сумісність з старими пристроями Android та необхідна потужність для деяких функцій.

Vuforia Engine – це один із провідних інструментів для розробки додатків доповненої реальності (AR) в індустрії комп'ютерного зору (рис. 2.3). Розроблений компанією PTC, Vuforia Engine дозволяє розробникам створювати інтерактивні AR-додатки для різних платформ, включаючи Android, iOS та HoloLens [8].



Рисунок 2.3 – Vuforia Engine

Можливості, які надає розробникам Vuforia Engine:

Розпізнавання зображень. Vuforia Engine дозволяє визначати та відстежувати зображення в реальному часі. Це дозволяє розпізнавати товари, книги, знаки, зображення на плакатах тощо.

Трекінг об'єктів. Застосунки можуть відстежувати 3D об'єкти, надаючи можливість взаємодіяти з ними в реальному світі.

Виявлення поверхні. Виявлення плоских та нерівних поверхонь дозволяє віртуальним об'єктам точно розміщуватися в реальному просторі.

Керування жестами. Vuforia Engine підтримує розпізнавання жестів, дозволяючи користувачам взаємодіяти з AR-додатками за допомогою рухів.

Розширений відтворення відображення (Occlusion). Додатки можуть взаємодіяти з реальними об'єктами, приховуючи віртуальні об'єкти за реальними, що робить AR-взаємодію більш реалістичною.

Хмарні технології (Cloud Recognition). Можливість розпізнавання об'єктів за допомогою хмарних сервісів, що дозволяє обробляти великі набори даних для розпізнавання.

Vuforia Engine має потужні алгоритми розпізнавання, які дозволяють точно ідентифікувати об'єкти та зображення, навіть у складних умовах. Підтримка різних платформ, включаючи Android, iOS та HoloLens, дозволяє розробникам створювати додатки для різних пристроїв, а активна спільнота розробників і широкий спектр документації забезпечують підтримку та можливість вирішення проблем. Можливість розпізнавати зображення, трекінг об'єктів і використовувати хмарні технології робить Vuforia Engine вельми гнучким і потужним інструментом для розробників.

Недоліки Vuforia Engine полягають у необхідності потужності апаратної частини. Певні функції, зокрема розширене відтворення відображення, можуть вимагати потужного обладнання, що може вплинути на продуктивність на старших або менш потужних пристроях. Деякі функції Vuforia Engine мо-

жуть вимагати плати за використання, що може стати обмеженням для невеликих команд та розробників з обмеженим бюджетом. Також необхідно враховувати можливі проблеми сумісності з різними пристроями і версіями операційних систем.

В даній кваліфікаційній роботі було вирішено використовувати Vuforia Engine з огляду на те, що даний інструмент володіє потужними алгоритмами розпізнавання зображень, доступний на різних платформах, має широкий спектр функцій для роботи з доповненою реальністю, включаючи розширені можливості відображення 3D об'єктів, анімації, а також взаємодії з ними, а також має активну спільноту користувачів та детальну документацію.

### **3. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ AR-НАВІГАЦІЇ**

#### **3.1 Технологія доповненої реальності**

Доповнена реальність (AR) – це інноваційна технологія, яка поєднує у собі віртуальний та реальний світи, дозволяючи користувачам взаємодіяти з віртуальними об'єктами та інформацією у реальному часі. AR змінює спосіб взаємодії з навколишнім світом, додаючи до нього комп'ютерну графіку, звук та інші сенсорні враження. Основний принцип роботи AR полягає в тому, що вона використовує пристрої, такі як смартфони, планшети або спеціальні окуляри для відображення віртуальних об'єктів поверх реальних об'єктів у реальному часі [9].

Ключові аспекти та технічні деталі технології AR полягають у наступному:

##### **1. Визначення позиції та відстеження руху:**

AR-пристрої, такі як смартфони або AR-окуляри, використовують різні сенсори, включаючи гіроскопи, акселерометри та магнітометри, щоб безперервно відстежувати рухи та орієнтацію пристрою у просторі. Ця інформація дозволяє програмам AR коректно відображати віртуальні об'єкти відповідно до рухів користувача.

##### **2. Розпізнавання та відстеження об'єктів:**

AR-програми можуть використовувати технології комп'ютерного зору для розпізнавання об'єктів та маркерів у реальному світі. Маркери можуть бути спеціальними зображеннями чи унікальними зразками, які використовуються для визначення позиції віртуальних об'єктів. Це дозволяє програмі точно позиціонувати віртуальні елементи в реальному світі.

##### **3. Проекція та відображення:**

AR використовує методи проекції у тому, щоб віртуальні об'єкти здавалися частиною реального світу. AR проектує зображення або відео в реаль-



ному часі на екран пристрою, додаючи віртуальні об'єкти до картини навколишнього світу. Це відбувається настільки швидко, що користувач бачить віртуальні об'єкти так, ніби вони існують у реальному світі.

#### 4. Інтерактивна взаємодія:

Одним із ключових аспектів AR є можливість взаємодії користувача з віртуальними об'єктами. Це може бути сенсорне введення через екран пристрою або жести користувача. Наприклад, користувач може натиснути на віртуальний об'єкт, щоб взаємодіяти з ним або використовувати жести, щоб керувати рухом або поворотом.

#### 5. Змішування реальності та віртуальності:

У AR відбувається змішування реальних та віртуальних елементів у реальному часі. Це означає, що користувач бачить як реальні об'єкти, так і віртуальні об'єкти одночасно, що створює враження, що вони існують в тому самому просторі.

#### 6. Зворотній зв'язок та адаптація:

AR-програми можуть отримувати зворотний зв'язок від користувача та адаптувати віртуальні об'єкти відповідно до цього зворотного зв'язку. Наприклад, якщо користувач переміщає віртуальний об'єкт, система може адаптувати його положення та орієнтацію у реальному часі.

#### 7. Визначення глибини:

Деякі пристрої AR використовують технології, такі як стереокамери або структуроване світло, щоб визначити глибину сцени. Це дозволяє більш точно позиціонувати віртуальні об'єкти у тривимірному просторі, що покращує реалізм та взаємодію з навколишнім світом.

#### 8. Хмарні обчислення та інтелектуальні системи:

Деякі AR-додатки використовують хмарні обчислення та штучний інтелект для аналізу даних та покращення візуального досвіду. Використання хмарних обчислень дозволяє обробляти складні обчислення та графіку за межами можливостей самого пристрою, що дає більше свободи у створенні вражаючих візуальних ефектів.

Для роботи з технологією AR потрібно мати пристрій з AR, спеціальні програми, хорошу камеру та сенсори. Для роботи з AR необхідний пристрій, який підтримує цю технологію. Це можуть бути сучасні смартфони, планшети або спеціальні AR-окуляри. Для взаємодії з доповненою реальністю необхідні спеціальні програми, які можуть працювати з AR-технологією. Ці програми можуть бути розроблені для різних цілей, від освіти та розваг до бізнесу та медицини. Пристрій повинен мати високоякісну камеру та інші сенсори, такі як гіроскопи та акселерометри, для точного відстеження рухів та позиції пристрою [10].

Одним із найбільш поширених застосувань технології доповненої реальності є AR-навігація.

### **3.2 Поняття технологій AR-навігації**

Навігація в приміщенні за допомогою технологій доповненої реальності (Indoor AR navigation, AR-навігація) – це технологія, яка використовує доповнену реальність, щоб допомогти людям орієнтуватися в приміщеннях. Вона поєднує потужність доповненої реальності з системами позиціонування в приміщенні, щоб допомогти користувачам орієнтуватися в приміщенні з точністю та легкістю [11].

Використовуючи камери та датчики, такі як GPS, Wi-Fi та Bluetooth, системи AR знімають оточення користувача, ідентифікуючи ключові елементи, як-от стіни, двері та інші орієнтири. Ці датчики можуть включати. Одночасно система відстежує рухи користувача, дозволяючи створювати динамічну карту середовища в приміщенні в режимі реального часу. Після того, як система зрозуміє оточення користувача, накладання AR застосовуються до реального світу, що відображається на пристрої користувача, наприклад смартфоні або окулярах AR. Ці накладення можуть включати інтерактивні карти, стрілки на-

прямків і корисні анотації, які накладаються на вигляд фізичного простору користувача. Завдяки інтерфейсу AR користувачі отримують інтуїтивно зрозумілі покрокові вказівки, що робить навігацію легкою та зручною.

Наприклад, навігаційна програма з доповненою реальністю може показати користувачеві карту його поточного місцезнаходження з вказівками до місця призначення. Додаток також може показувати необхідні місця, такі як туалети, ліфти та виходи. Користувачу можуть навіть відображатись стрілки на підлозі, які показують шлях.

AR-навігація в приміщенні має низку переваг перед традиційними методами навігації. Однією з переваг є те, що він більш точний. Традиційні методи навігації, такі як GPS, можуть бути неточними в приміщенні через перешкоди сигналу та похибку визначення місцезнаходження. AR-навігація в приміщенні може використовувати комбінацію датчиків, щоб точніше визначати місцезнаходження користувача.

Ще одна перевага AR-навігації в приміщенні полягає в тому, що вона дуже зручна для користувача. Користувачі можуть просто дивитися в тому напрямку, в якому вони хочуть рухатися, і навігаційна система AR покаже їм шлях.

AR-навігація в приміщенні все ще є відносно новою технологією, але вона стрімко набирає популярність. Вона використовується в різних місцях, включаючи аеропорти, лікарні, музеї та торгові центри.

Області використання AR-навігації:

Аеропорти: навігацію в приміщенні з доповненою реальністю можна використовувати, щоб допомогти пасажиром зорієнтуватися в завантажених аеропортах. Наприклад, програма може показувати пасажиром напрямок до їх воріт, туалетів та інших зручностей.

Лікарні: AR-навігацію можна використовувати, щоб допомогти пацієнтам і відвідувачам орієнтуватися в лікарнях. Наприклад, програма може показувати пацієнтам напрямки до кабінету лікаря, аптеки та кафе.

Музеї: навігація в приміщенні з використанням AR може допомогти відвідувачам зорієнтуватися в музеях і дізнатися більше про експонати. Наприклад, програма може показувати відвідувачам маршрути до певних експонатів, надавати інформацію про експонати та навіть дозволяти відвідувачам взаємодіяти з експонатами.

Торгові центри: AR-навігація в приміщенні може бути використана, щоб допомогти покупцям орієнтуватися в торгових центрах і знаходити магазини та продукти, які вони шукають. Наприклад, програма може показувати покупцям маршрути до певних магазинів, надавати інформацію про розпродажі та акції та навіть дозволяти покупцям віртуально приміряти одяг і аксесуари.

Окрім наведених прикладів AR-навігація в приміщенні може допомогти відвідувачам орієнтуватися в офісних будівлях, школах, будівлях університетів, спортивних об'єктах, готелях, складах тощо.

### **3.3 Аналіз різних типів внутрішньої навігації**

Існує кілька видів indoor AR-навігації, кожен з яких має свої плюси та мінуси [12].

#### **1. Маркерна AR-навігація.**

Перевагами маркерної навігації є точність. Маркери, які можуть бути реалізовані як фізичні об'єкти або QR-коди, забезпечують високу точність навігації, оскільки вони є опорними точками визначення положення користувача. Також відсутня залежність від мережі. Навігація не вимагає постійного підключення до Інтернету, оскільки використовує маркери у реальному світі. Простота застосування полягає у тому, що не потрібна складна інфраструктура для підтримки маркерної навігації.

Однак при цьому виді навігації обмежена область покриття: користувач обмежений доступними маркерами, і їхнє розміщення має бути заздалегідь сплановане.

2. Технології одночасної локалізації і картографування (англ. simultaneous localization and mapping – SLAM-технології).

Основна ідея SLAM полягає в тому, щоб датчики, які вимірюють відстані та кути, використовувалися для створення карти навколишнього простору та одночасної локалізації робота чи пристрою в цій карті. Плюсом є свобода переміщення, адже SLAM-технології дозволяють користувачеві вільно переміщатися всередині приміщення без обмежень маркерів. Навігація ґрунтується на сприйнятті навколишнього середовища, що дозволяє охопити великі простори. Також системи SLAM можуть адаптуватися до змін у навколишньому середовищі, таких як рухомі об'єкти або перестановка меблів.

Мінусами є вимоги до апаратури, адже потрібне потужне обладнання, включаючи камери та сенсори, що може підвищити вартість та енергоспоживання пристрою, та складність програмування: розробка точних та надійних систем SLAM потребує складного програмування та налаштування.

### 3. Інфраструктурна AR-навігація.

Плюсами є точність та надійність. Інфраструктурна AR-навігація використовує фіксовані сенсори та інфраструктуру (наприклад, Bluetooth-маяки або Wi-Fi точки доступу), що забезпечує високу точність навігації. Може інтегруватися з іншими системами розумного будинку або управління будинками, надаючи розширений функціонал для користувачів.

Недоліками є високі витрати на використання, бо потрібна установка спеціальної інфраструктури, що може бути дорогим і трудомістким. Також обмежена масштабованість.

Кожен вид внутрішньої AR-навігації має свої сильні та слабкі сторони, і вибір певного підходу залежить від конкретних потреб проекту, бюджету та технічних можливостей.

Таким чином, з урахуванням цих типів внутрішньої навігації загальний алгоритм внутрішньої навігації може бути наступним:

1. Визначення початкового положення. Щоб встановити початкове положення користувача в просторовій моделі, можуть використовуватись маркери, QR-коди, датчики, SLAM-технології, Bluetooth-маяки або Wi-Fi точки доступу тощо.

2. Відстеження. Після встановлення початкового положення місцезнаходження користувача відстежується за допомогою компонента відстеження пози. Цей компонент зазвичай є частиною AR-камери. Він фіксує візуально відмінні особливості в середовищі, щоб працювати як якір і відстежувати рухи та обертання пристрою відносно положення виявлених функцій. Такий компонент також підтримується датчиками мобільного телефону (гіроскоп і акселерометр).

3. Навігація. Різні алгоритми застосовуються для пошуку найкоротшого маршруту для навігації до місця призначення. Детально вони описані в підрозділі 3.4.

Діаграма загальної взаємодії між компонентами позиціонування та навігації відображена на рис. 3.1.

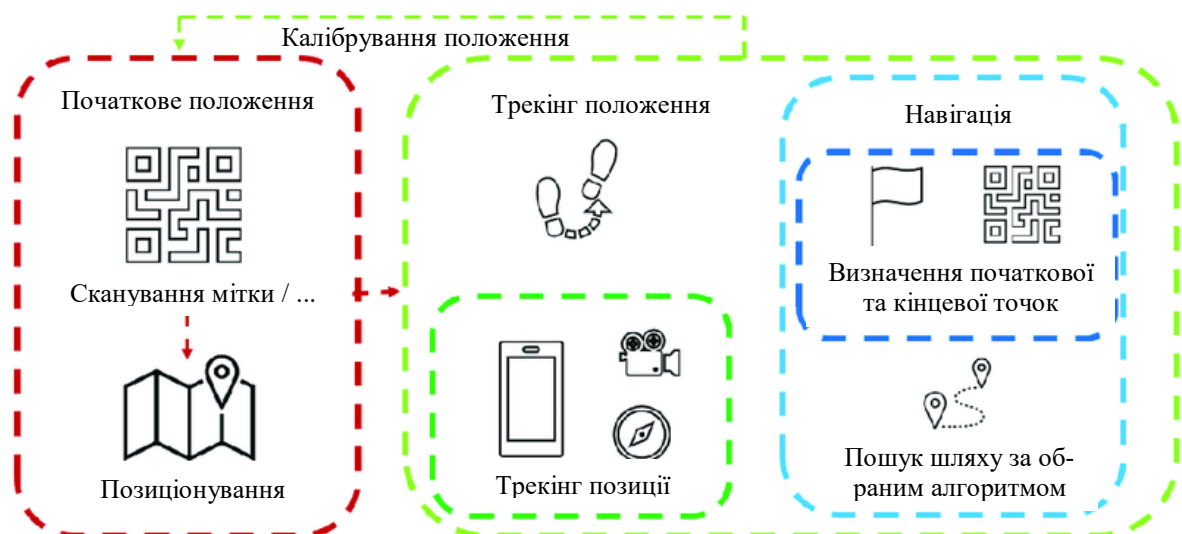


Рисунок 3.1 – Діаграма компонентів позиціонування та навігації

### 3.4 Задача пошуку шляху

Задача пошуку шляху – це завдання знаходження оптимального маршруту чи шляху між двома чи кількома точками у графі чи просторі. Це завдання зустрічається у багатьох областях, включаючи інформатику, штучний інтелект, географічні інформаційні системи, робототехніку, транспорт та навіть у біології.

Вона може використовуватися для різних цілей, таких як пошук найкоротшого шляху для транспорту або пересування робота, оптимізація маршрутів доставки, планування подорожей, пошук оптимального шляху комп'ютерних ігор або навіть аналіз мереж, таких як соціальні мережі або мережі передачі даних.

Для вирішення задачі пошуку шляху застосовуються різні алгоритми, такі як алгоритм Дейкстри, алгоритм A\*, пошук у ширину, пошук у глибину та інші, залежно від особливостей задачі та структури даних, які необхідно обробити.

Алгоритм Дейкстри – це алгоритм пошуку найкоротшого шляху з однієї початкової вершини (точки) графа до решти вершин. Він був розроблений нідерландським вченим Едсгер Дейкстрою в 1956 році.

Алгоритм Дейкстри працює для орієнтованих та неорієнтованих графів з невід'ємною вагою ребер. Він починає з початкової вершини і поступово будує найкоротші шляхи до інших вершин графа [13].

Принцип роботи алгоритму Дейкстри полягає у підтримці списку найкоротших відомих шляхів від початкової вершини до інших вершин та їх послідовному оновленні. Алгоритм поступово переглядає суміжні вершини та оновлює довжину найкоротшого шляху до них, якщо такий шлях коротший, ніж вже відомий. Він продовжує цей процес, доки не буде знайдено найкоротший шлях до всіх вершин.

Одне з основних застосувань алгоритму Дейкстри – пошук найкоротших шляхів у мережах зв'язку, автоматизоване планування маршрутів, оптимізація

транспортних мереж та інші завдання, де потрібне знаходження найменшого шляху між вершинами графа.

Кожній вершині з  $V$  ставиться мітка – відома мінімальна відстань від цієї вершини до  $a$ . Алгоритм працює покроково – на кожному кроці він "відвідує" одну вершину  $i$  і намагається зменшувати мітки. Робота алгоритму завершується, коли усі вершини відвідані (рис. 3.2).

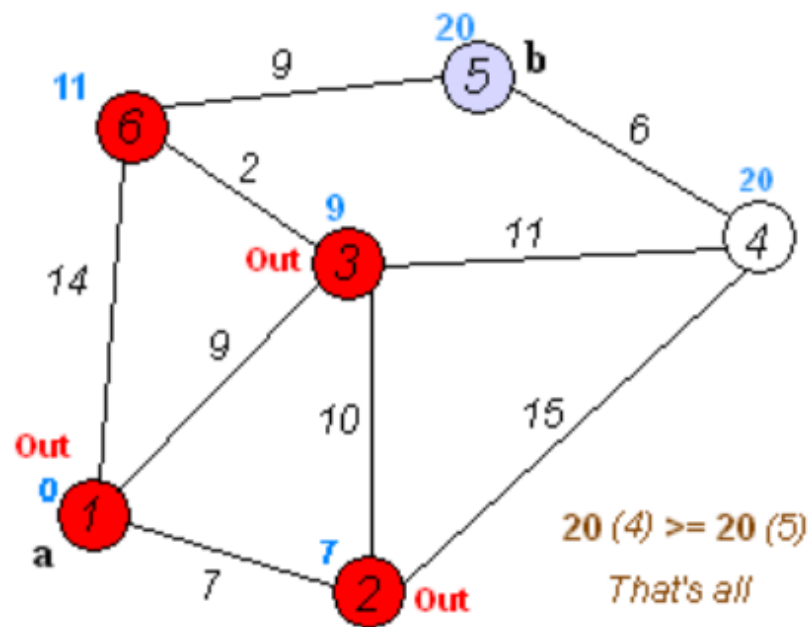


Рисунок 3.2 – Приклад розв'язання задачі пошуку шляху за алгоритмом Дейкстри

Мітка самої вершини  $a$  належить рівною 0, мітки інших вершин – нескінченності. Це показує, що відстані від  $a$  до інших вершин поки що невідомі. Усі вершини графа позначаються як невідвідані.

Якщо всі вершини відвідані, алгоритм завершується. В іншому випадку з ще не відвіданих вершин вибирається вершина  $u$ , що має мінімальну мітку. Розглядаються різні маршрути, в яких  $u$  є передостаннім пунктом. Вершини, які ведуть ребра з  $u$  – це сусіди цієї вершини. Для кожного сусіда вершини  $u$ ,



крім зазначених як відвідані, розглядається нова довжина шляху, що дорівнює сумі значень поточної мітки  $u$  та довжини ребра, що з'єднує  $u$  з цим сусідом.

Якщо отримане значення довжини менше значення мітки сусіда, значення мітки замінюється на отримане значення довжини. Розглянувши всіх сусідів, вершина позначається як відвідана і крок алгоритму повторюється.

Алгоритм  $A^*$  (A-star) – це популярний алгоритм пошуку найкоротшого шляху у графі чи просторі станів. Він є комбінацією евристичного та поінформованого пошуку [14].

Алгоритм  $A^*$  ефективно знаходить оптимальний шлях від початкової до цільової точки, використовуючи евристичну оцінку відстані від поточної точки до мети. Основна ідея полягає в тому, щоб вибирати наступну точку дослідження, комбінуючи вартість проходження цієї точки з евристичною оцінкою її віддаленості від мети.

Алгоритм  $A^*$  використовує два показники для кожної точки:

- $g(x)$  – вартість шляху від початкової точки до точки  $x$ .
- $h(x)$  – евристична оцінка вартості від точки  $x$  до мети (це може бути пряма відстань, евристична оцінка або будь-яка інша функція).

На кожному кроці алгоритм вибирає наступну точку для розкриття підсумовуючи вартість проходження цієї точки ( $g(x)$ ) з евристичною оцінкою ( $h(x)$ ), і вибирає точку з найменшою сумою. Це дозволяє алгоритму рухатися у бік мети, чекаючи швидкого досягнення оптимального шляху.

$A^*$  покроково переглядає всі шляхи, що ведуть від початкової вершини до кінцевої, поки не знайде мінімальний. Як і всі інформовані алгоритми пошуку, він переглядає спочатку маршрути, які «здаються» провідними до мети. Від жадібного алгоритму, який теж є алгоритмом пошуку за першим найкращим збігом, його відрізняє те, що при виборі вершини він враховує, крім іншого, весь пройдений до неї шлях. Складова  $g(x)$  – це вартість шляху від початкової вершини, а не від попередньої, як у жадібному алгоритмі.

На початку роботи проглядаються вузли, суміжні з початковим; вибирається той із них, який має мінімальне значення  $f(x)$ , після чого цей вузол розкривається. На кожному етапі алгоритм оперує з безліччю шляхів з початкової точки до всіх ще не розкритих (листових) вершин графа — безліччю приватних рішень, які розміщуються в черзі з пріоритетом. Пріоритет шляху визначається за значенням  $f(x) = g(x) + h(x)$ . Алгоритм продовжує свою роботу до тих пір, поки значення  $f(x)$  цільової вершини не виявиться меншим, ніж будь-яке значення в черзі, або поки дерево не буде переглянуто. З багатьох рішень вибирається рішення з найменшою вартістю (рис. 3.3).

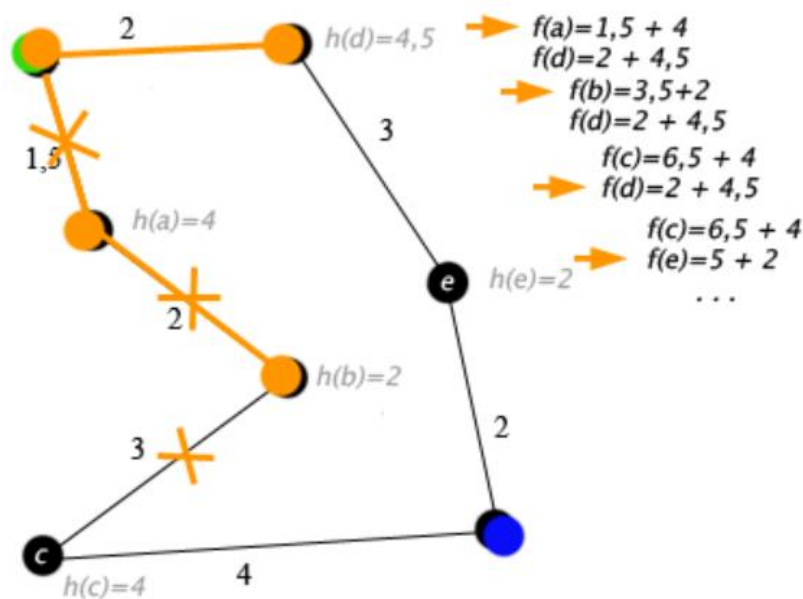


Рисунок 3.3 – Приклад розв’язання задачі пошуку шляху за алгоритмом  $A^*$

Алгоритм пошуку в ширину (BFS, Breadth-First Search) – це алгоритм пошуку шляху або обходу графа, який починає із заданої початкової вершини і досліджує всі суміжні з нею вершини на поточному рівні графа перед переходом до вершин наступного рівня [15].

Принцип роботи алгоритму пошуку у ширину полягає в тому, що він досліджує всі найближчі сусідні вершини від початкової точки, потім руха-

ється на наступний рівень і досліджує їх сусідів і так далі, доки не будуть досліджені всі вершини графа або поки не буде знайдено цільовий елемент (рис. 3.4).

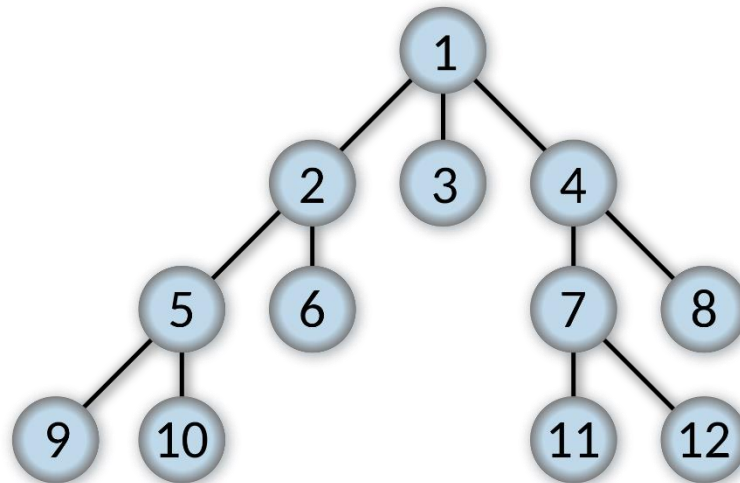


Рисунок 3.4 – Порядок обходу вершин дерева в алгоритмі пошуку в ширину

Алгоритм пошуку у ширину використовує структуру даних чергу для відстеження порядку обробки вершин. Це гарантує, що вершини обробляються в порядку віддаленості від початкової точки: спочатку всі вершини на відстані 1, потім на відстані 2 і так далі.

Алгоритм пошуку в глибину (DFS, Depth-First Search) – це алгоритм обходу або пошуку шляху в графі або дереві, який починає із заданої початкової вершини і послідовно досліджує всі можливі шляхи, заглиблюючись вглиб доти, доки не буде досягнуто кінцевої вершини, після чого повертається до попередньої вершини та продовжує дослідження інших шляхів.

Принцип роботи алгоритму пошуку в глибину полягає в тому, що він вибирає одну із суміжних недосліджених вершин, переходить до неї та продовжує дослідження з цієї нової вершини. Якщо він досягає вершини, яка більше не має недосліджених сусідів, алгоритм повертається до попередньої вершини і продовжує пошук інших шляхів (рис. 3.5).

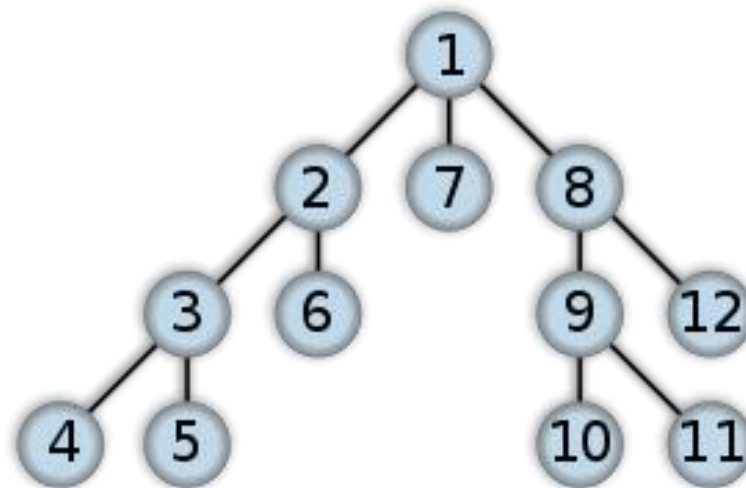


Рисунок 3.5 – Порядок обходу вершин дерева в алгоритмі пошуку в глибину

DFS використовує рекурсію або стек для відстеження шляху та відвіданих вершин. Він глибоко досліджує кожну гілку графа до того часу, поки досягне кінцевої точки чи обійде всі вершини.

Використання певного алгоритму пошуку шляху часто залежить від того, які інструменти та бібліотеки доступні розробнику. Іноді необхідність вибору може бути обумовлена вже наявними рішеннями або обмеженнями в системі. У більшості ситуацій використання вже існуючих рішень або інструментів може бути більш практичним і ефективним, ніж створення нового алгоритму з нуля.

## **4. ПРОЕКТУВАННЯ ЗАСТОСУНКУ**

Після проведення аналізу предметної області, аналогів та вибору засобів розробки, необхідним етапом є проектування інформаційної системи. На цьому етапі було створено UML-діаграми варіантів використання (прецедентів), активності та послідовностей. Всі UML-діаграми було розроблено за допомогою об'єктно-орієнтованого CASE-засобу для проектування інформаційних систем IBM Rational Rose. Цей інтегрований засіб не лише надає можливості проектування архітектури, аналізу та моделювання, а й забезпечує розробку інформаційних систем у цілому.

### **4.1 Проектування варіантів використання системи**

Діаграма варіантів використання (прецедентів) відображає функціональне призначення проектованої програмної системи. Суть діаграми прецедентів полягає в тому, що систему представляють як групу акторів, які за допомогою варіантів використання взаємодіють із нею.

Актор – це сутність, що взаємодіє з системою для вирішення деяких завдань. Актором може бути людина, інша система, пристрій або програмний засіб. В даному випадку акторами визначено користувача мобільного застосунку та адміністратора.

Після визначення акторів системи, необхідно сформулювати перелік усіх варіантів використання, з якими будуть взаємодіяти визначені актори. Серед основних варіантів використання системи: з боку користувача – перегляд мапи університету, перегляд маршруту в AR, позиціонування на основі номера найближчої аудиторії, отримання інформації про аудиторії, вибір аудиторій для побудови маршруту, з боку адміністратора – додавання або редагування інформації в базі даних.

За результатами сформованих варіантів використання та акторів системи було розроблено діаграму варіантів використання, яку наведено на рис. 4.1.

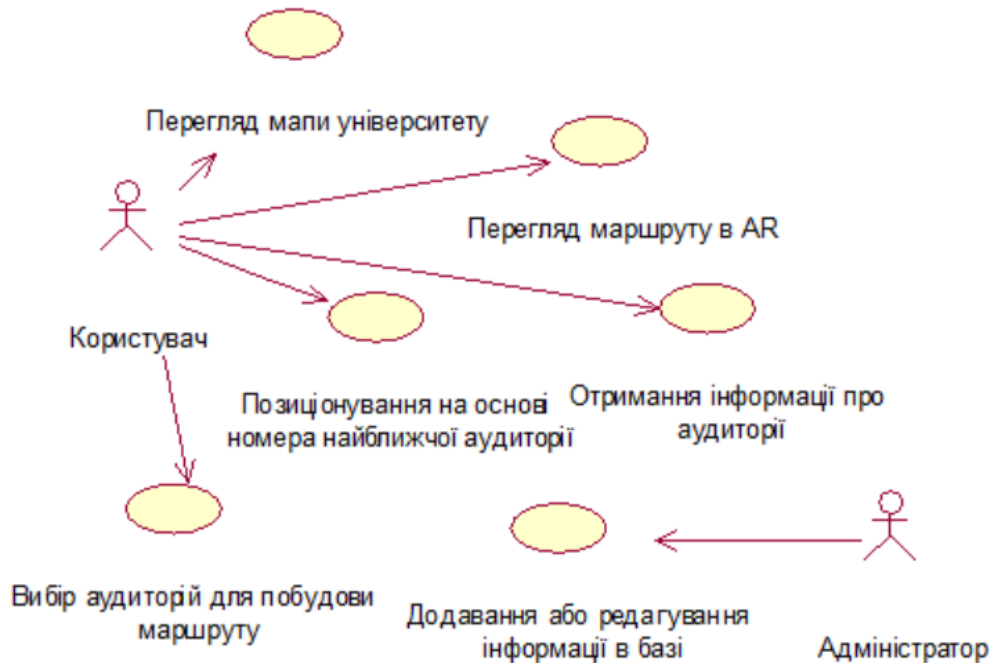


Рисунок 4.1 – Діаграма прецедентів

## 4.2 Створення діаграми діяльності

Для уявлення процесу виконання операцій в рамках мови UML застосовують діаграми діяльності, що відомі також як діаграми активностей. Ці графічні зображення дозволяють відтворити послідовність дій або переходів від однієї діяльності до іншої, зосереджуючись на результаті виконання. Цей результат може вплинути на стан системи або повернути певне значення. Діаграма діяльності відображає послідовність дій або контрольний потік у системі, що нагадує блок-схему або діаграму потоку даних. Такі діаграми можуть описувати етапи в діаграмі варіантів використання. Вони можуть моделювати як послідовні, так і одночасні дії. В обох випадках, діаграма діяльності має відображення початку (початковий стан) та кінця (кінцевий стан).

Для застосунку була створена діаграма діяльності, яка відображає основний процес проектованої системи – позиціонування та навігацію користувача (рис. 4.2).



Рисунок 4.2 – Діаграма діяльності

Користувач, який хоче побудувати маршрут від свого поточного місцезнаходження до певної аудиторії, має навести камеру на номер найближчої аудиторії. Він отримає результат розпізнавання – номер та інформацію про аудиторію. Якщо аудиторія розпізнана помилково, він може повторити сканування. Якщо все правильно і користувач позиціонований в університеті, йому необхідно обрати зі списку номер цільової аудиторії. Після цього йому буде запропонований оптимальний маршрут та відображений з використанням технологій доповненої реальності. Коли користувач дістанеться цільової аудиторії, процес навігації буде завершено.

### **4.3 Створення діаграми послідовностей**

Діаграма послідовностей - це вид діаграми, для візуалізації взаємодії між об'єктами чи компонентами системи у певний час. Вона дозволяє показати послідовність обміну повідомленнями між об'єктами чи компонентами у часі, ілюструючи порядок виконання. Ці діаграми корисні при проектуванні програмного забезпечення, оскільки вони дозволяють легше зрозуміти, як об'єкти чи компоненти взаємодіють один з одним у межах певної функціональності чи процесу. Шляхом візуалізації послідовності дій та повідомлень між об'єктами можна краще зрозуміти, як система поводитиметься у різних сценаріях використання. До того, діаграми послідовностей допомагають при налагодженні та тестуванні, оскільки вони дозволяють виявити потенційні проблеми у логіці взаємодії між компонентами системи.

На рис. 4. 3 зображена діаграма головної послідовності сценарію розпізнавання номеру аудиторії в застосунку.





Рисунок 4.3 – Діаграма послідовностей

Користувач сканує номер аудиторії за допомогою камери через застосунок, отримане зображення обробляється Vuforia Engine, яка знаходить відповідну ціль у хмарній базі даних та повертає дані у застосунок. За результатом розпізнавання формується запит до хмарної бази даних Firebase Realtime Database, яка повертає відповідну інформацію про аудиторію, і ця інформація відображається для користувача у застосунку.

Таким чином, на етапі проектування системи були створені UML-діаграми прецедентів, активності і послідовності, які допомогли зрозуміти основні варіанти використання системи та логіку роботи агента.

## 5. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ «НАВІГАТОР ПО ОДЕКУ»

### 5.1 Інтеграція Vuforia Engine з Unity

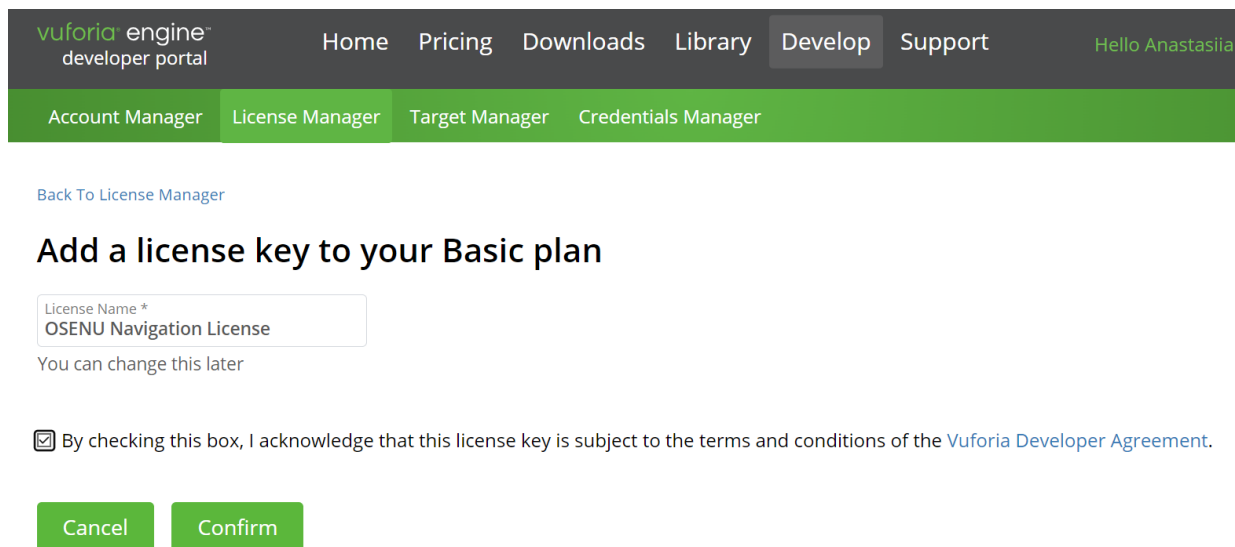
Для роботи з Vuforia Engine та використання її функціоналу необхідно зареєструватися на порталі розробників Vuforia Engine. Для кожного нового проекту необхідно отримати ліцензійний ключ. Існує базовий план (Basic Plan) та преміум план (Premium Plan).

Безкоштовний базовий план дозволяє створювати ліцензійні ключі Vuforia Engine, які розблокують використання кількох функцій і служб Vuforia. Ліцензійні ключі, створені за базовим планом, підтримують необмежену розробку та публікацію додатків без водяного знака з такими функціями Vuforia: розпізнавання зображень (Image Targets), кількох цілей (Multi Targets), циліндричних цілей (Cylinder Targets), спеціальних цілей Vuforia VuMarks та плоских поверхонь (Ground Plane). До того ж, доступна робота з AR Foundation, Advanced Camera API, файловим драйвером. Безкоштовний план має такі ліміти: хмарне розпізнавання – до 1000 зображень у базі і до 1000 розпізнавань зображень на місяць.

Базовий план дозволяє розробку за допомогою функцій Model Targets і Area Targets, але програму не можна опублікувати в магазині програм. Крім того, водяний знак Vuforia буде присутній під час використання цих функцій разом із базовим ліцензійним ключем.

Преміум-план – це річна підписка, яка дозволяє отримувати ліцензійні ключі Vuforia Engine, які можна використовувати з функціями Model Target і Area Target Vuforia. Ці ліцензійні ключі розблоковують усі функції Vuforia, які підтримує базовий план, і додатково розширюють необмежену розробку та публікацію програм без водяного знака для розпізнавання моделей (Model Targets), областей (Area Targets) та сканеру штрих-кодів [16].

Базовий план достатній для невеликих проектів. Щоб створити ліцензійний ключ базового плану, необхідно перейти до Менеджеру ліцензій на Порталі розробників (рис. 5.1).



The screenshot shows the Vuforia Engine developer portal interface. At the top, there is a navigation bar with the logo 'vuforia engine developer portal' on the left and links for 'Home', 'Pricing', 'Downloads', 'Library', 'Develop', and 'Support' in the center. On the right, it says 'Hello Anastasiia'. Below the navigation bar is a secondary menu with 'Account Manager', 'License Manager', 'Target Manager', and 'Credentials Manager'. The 'License Manager' tab is active. Below the menu, there is a link 'Back To License Manager'. The main heading is 'Add a license key to your Basic plan'. There is a text input field for 'License Name \*' containing 'OSENU Navigation License'. Below the field, it says 'You can change this later'. There is a checked checkbox with the text 'By checking this box, I acknowledge that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.' At the bottom, there are two buttons: 'Cancel' and 'Confirm'.

Рисунок 5.1 – Створення ліцензійного ключа базового плану Vuforia Engine

Vuforia Engine for Unity надається як архівний пакет, який можна додати до локальних проектів через менеджер пакетів Unity. Додати Vuforia до проекту Unity можна декількома способами: додавання пакета через сценарій редактора, додавання залежності в менеджер пакетів або завантаження пакета та додавання його вручну.

У кваліфікаційній роботі використовується версія Vuforia Engine 10.19.3. Пакет .unitypackage був завантажений з порталу розробників Vuforia Engine (рис. 5.2).

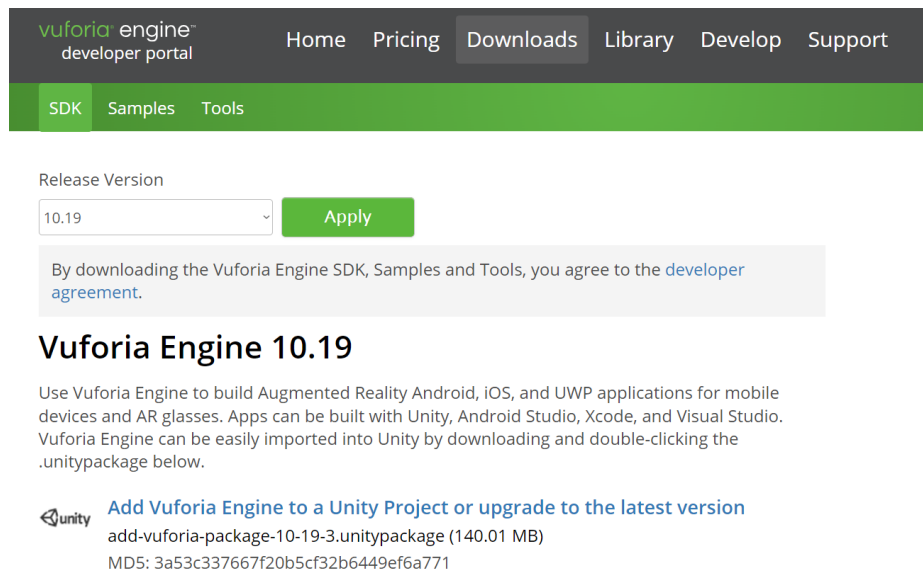


Рисунок 5.2 – Завантаження Vuforia Engine з порталу розробників

При імпорті пакету завантажуються два файли, які містять конфігураційні налаштування Vuforia та самі ресурси для Vuforia Engine, такі як моделі, текстури, файли конфігурації та інші активи, необхідні для роботи з доповненою реальністю (рис. 5.3).

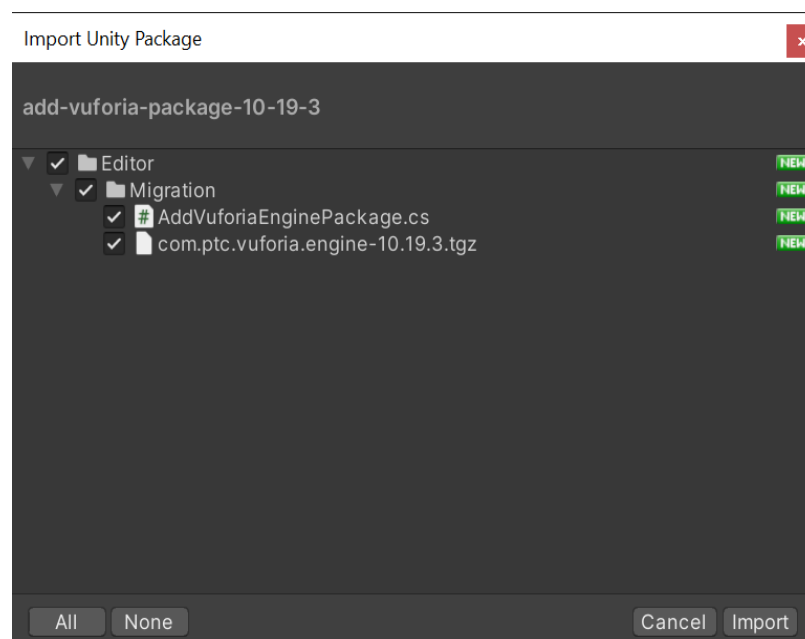


Рисунок 5.3 – Імпорт Vuforia Engine в Unity

Після успішного імпорту ресурсів Vuforia Engine, необхідно відкрити конфігурацію Vuforia та додати створений раніше ліцензійний ключ розробника Vuforia у поле ліцензійного ключа програми App License Key (рис. 5.4).

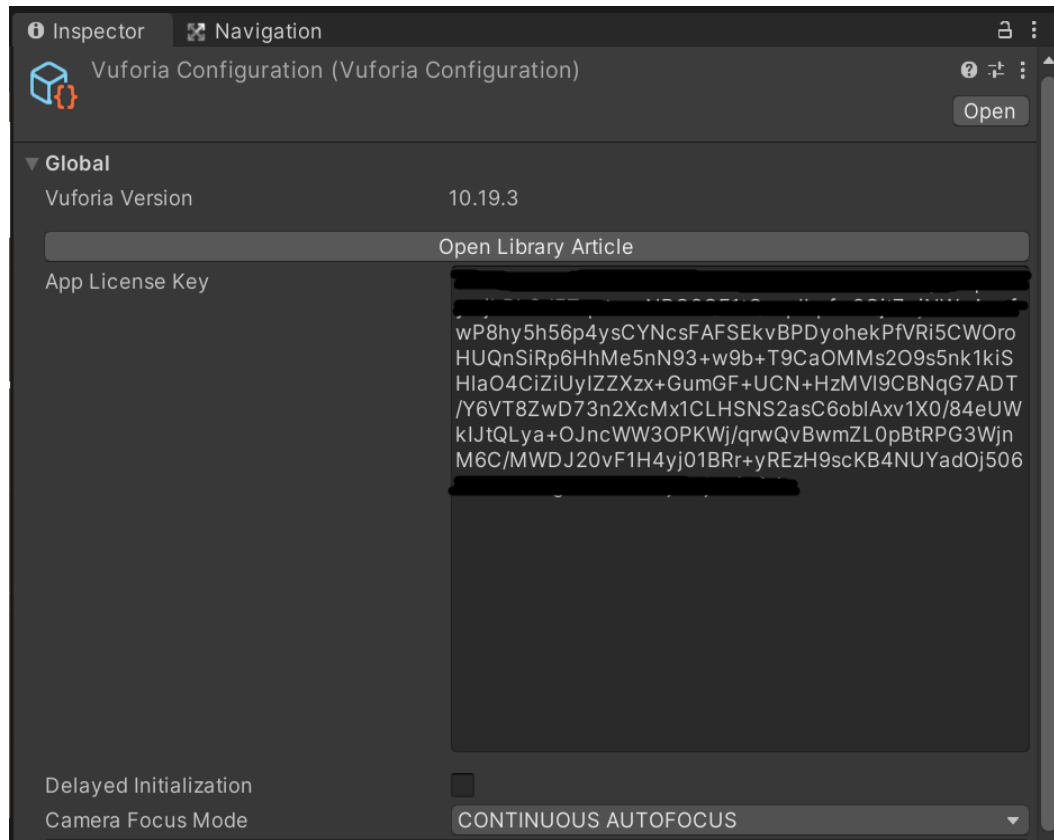


Рисунок 5.4 – Конфігурація Vuforia Engine

Для зберігання зображень для розпізнавання необхідно створити базу даних Vuforia. У Vuforia Engine є три види баз даних: локальна, хмарна та база VuMark.

Локальна база даних (Device Database, база даних пристрою) – це набір ресурсів часу виконання та авторських файлів для цілей Vuforia. База даних доступна локально та містить цілі, які відстежує програма [17].

Хмарна база даних – це набір зображень, розміщених на серверах Vuforia, готових до запитів. Коли зображення розпізнається, воно працюватиме як цільове зображення.

Хмарні бази даних надають програмам на основі Vuforia велику кількість цілей. Хмарні бази даних зберігаються в Інтернеті та здатні здійснювати пошук у мільйонах зображень. Ідентифікація Cloud Image Targets може зайняти трохи більше часу, залежно від підключення до мережі, але вона забезпечує ефективне розпізнавання зображень [18].

VuMarks відрізняються за своїми варіантами використання від Image Targets і можуть використовуватися в ситуаціях, коли потрібна велика кількість копій цілі для розрізнення подібних об'єктів, наприклад, кодування серійного номера продукту.

У кваліфікаційній роботі використовується хмарна база даних з огляду на велику кількість зображень у перспективі, які необхідно розпізнавати. Щоб створити хмарну базу даних, необхідно на порталі розробників перейти в розділ Target Manager (Менеджер цілей) та обрати тип бази Cloud (рис. 5.5).

## Create Database

Database Name \*  
OSENUNavigationDatabase

Type:

Device

Cloud

VuMark

OSENU Navigation License

To create a license key please go to the [License Manager](#)

Cancel Create

Рисунок 5.5 – Створення хмарної бази даних Vuforia

Після створення бази даних в розділі Database Access Keys доступні ключі клієнтського доступу та ключі доступу до серверу.

Після створення нового проекту в Unity необхідно додати об'єкти хмарного розпізнавання для роботи з Vuforia. У сцені, в якій буде реалізовано розпізнавання номерів аудиторій, було додано AR Camera – спеціальний тип камери, який використовується для створення розширеної реальності. Основна функція AR Camera – це обробка вхідних зображень з камери пристрою та їх аналіз для визначення розташування та орієнтації у реальному світі для подальшого розпізнавання об'єктів та відображення віртуальних об'єктів. Отримане зображення з камери пристрою обробляється для виявлення об'єктів, що розпізнаються, у реальному часі.

Також було додано об'єкт Cloud Recognition – основний компонент, який встановлює зв'язок з хмарною базою та дозволяє розпізнавати об'єкти на основі хмарного розпізнавання. У параметрах цього об'єкта необхідно вказати ключі клієнтського доступу створеної раніше бази даних (рис. 5.6).

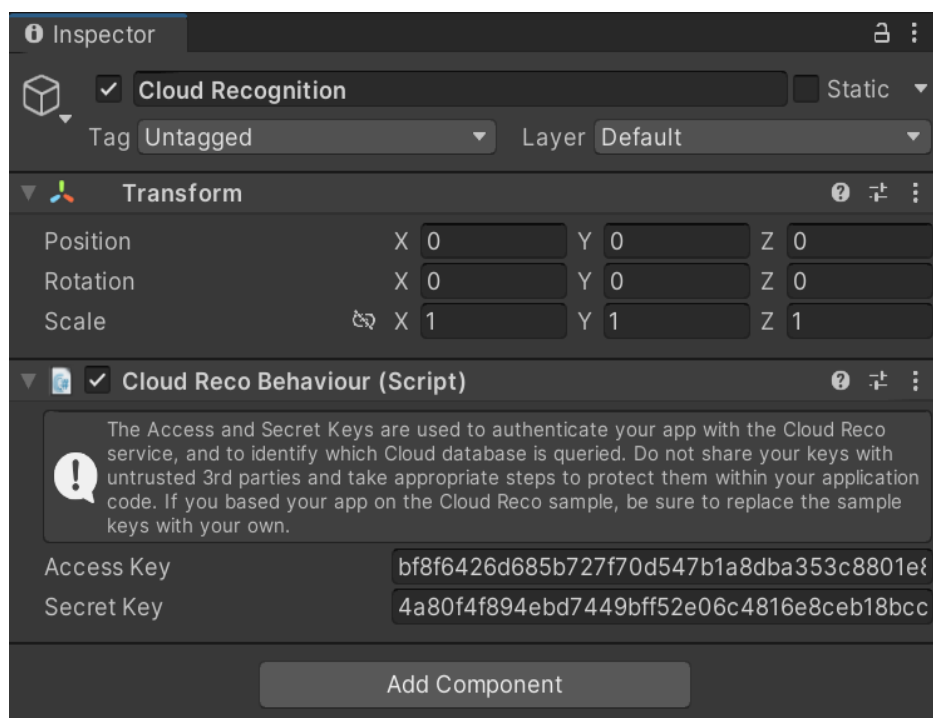


Рисунок 5.6 – Налаштування об'єкту Cloud Recognition

Ще одним обов'язковим об'єктом у сцені є Cloud Image Target. Він використовується для розпізнавання зображень з хмарних джерел, що дозволяє динамічно завантажувати та оновлювати набори даних для розпізнавання.

Додавати цільові зображення до хмарної бази даних можна через Менеджер цілей на порталі розробників Vuforia Engine. Для цього необхідно обрати створену раніше базу та натиснути «Add Target». При додаванні цілі необхідно завантажити зображення, вказати його ширину у метрах, за бажанням додати файл з метаданими та вказати ім'я цільового зображення (рис. 5.7). Дане ім'я буде ідентифікатором зображення в базі.

### Add Target

#### Target Image File:

.jpg or .png (max file 2mb)

#### Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

#### Metadata Package: (Optional)

max file 1mb

#### Name:

Рисунок 5.7 – Додавання цілі до хмарної бази даних Vuforia



Цільові зображення можна створювати за допомогою Vuforia Target Manager, використовуючи зображення JPG або PNG у RGB або відтінках сірого. Розмір вхідних зображень має становити 2,25 МБ або менше та мати мінімальну ширину 320 пікселів.

Цілі зображення виявляються на основі природних особливостей, які витягуються з цільового зображення, а потім порівнюються під час виконання з особливостями на зображенні камери в реальному часі. Рейтинг цілі коливається від 1 до 5 зірок. Незважаючи на те, що цілі з низьким рейтингом зазвичай можна виявляти та відслідковувати, краще використовувати цілі з 4 або 5 зірками для досягнення найкращих результатів.

Особливості, які є основою рейтингу, – це гострі деталі зображення, якот у текстурованих об’єктів – місця перепадів кольорів, контрасту, форми тощо (рис. 5.8). Аналізатор зображень представляє особливості у вигляді маленьких жовтих хрестиків, які можна переглянути для кожного зображення у базі (рис. 5.9).

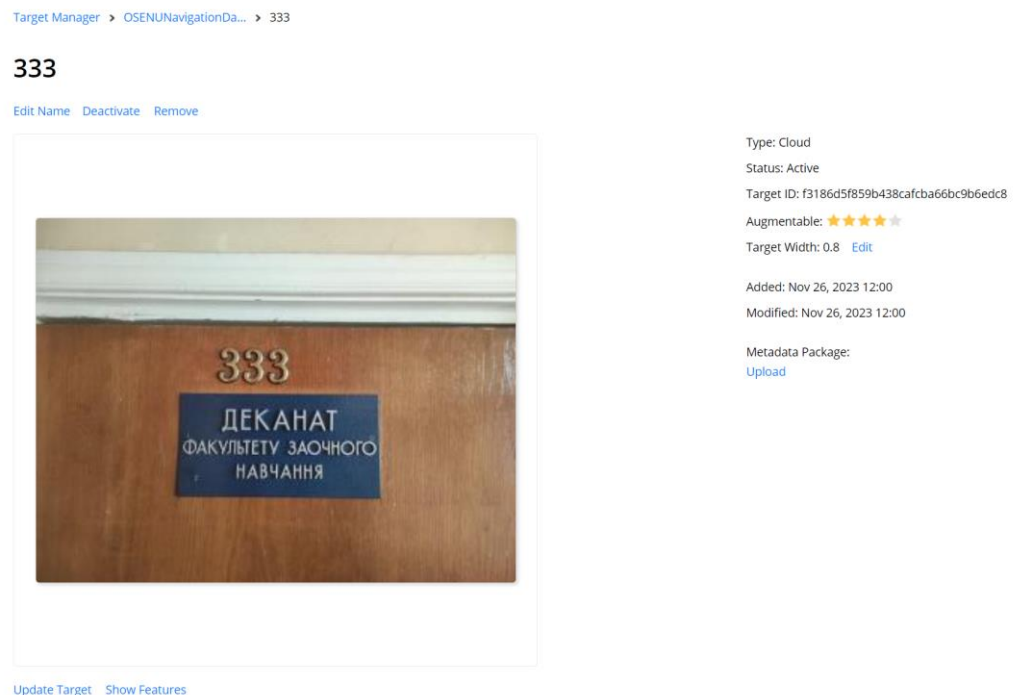


Рисунок 5.8 – Перегляд цільового зображення з 4 зірками



Рисунок 5.9 – Виявлені природні особливості цільового зображення

Для обробки результатів розпізнавання в Unity необхідно додати скрипт C# до об'єкту Cloud Recognition та в методі OnNewSearchResult отримати ідентифікатор зображення з хмарної бази.

```
public void OnNewSearchResult
(CloudRecoBehaviour.CloudRecoSearchResult cloudRecoSearchResult)
{
    picId = cloudRecoSearchResult.TargetName;
    ...
}
```

Для зберігання даних про розпізнані зображення можна використовувати метадані. Vuforia надає можливість додати до кожного зображення в базі файл метаданих у форматі JSON. Однак для оновлення даних це незручно, адже кожного разу необхідно видаляти старий файл, створювати новий та завантажувати його до цільових зображень. В даній роботі для зберігання даних про аудиторії використовується хмарна база даних Firebase Realtime Database.

## 5.2 Підключення платформи Firebase

Firebase – це платформа, розроблена компанією Google, яка надає набір інструментів та сервісів для розробки мобільних та веб-додатків. Вона пропонує різноманітні функції, що допомагають розробникам забезпечити аутентифікацію користувачів, зберігати та синхронізувати дані, аналізувати та вдосконалювати застосунки, відправляти повідомлення та робити багато іншого. Основні інструменти платформи Firebase наведені на рис. 5.10. [19]

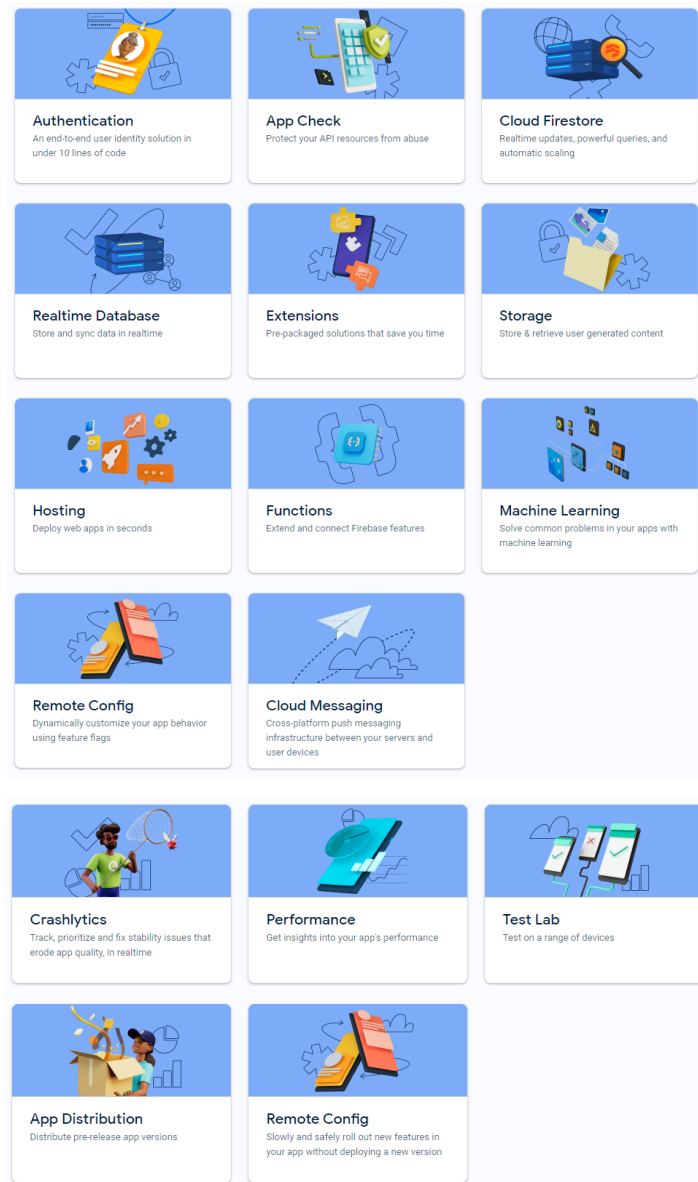


Рисунок 5.10 – Основні інструменти платформи Firebase

Щоб інтегрувати проект Unity з платформою Firebase, необхідно створити новий проект Firebase. Платформа Firebase надає повну інструкцію для додавання своїх інструментів в застосунки для iOS, Android, Web, Unity та Flutter (рис. 5.11).

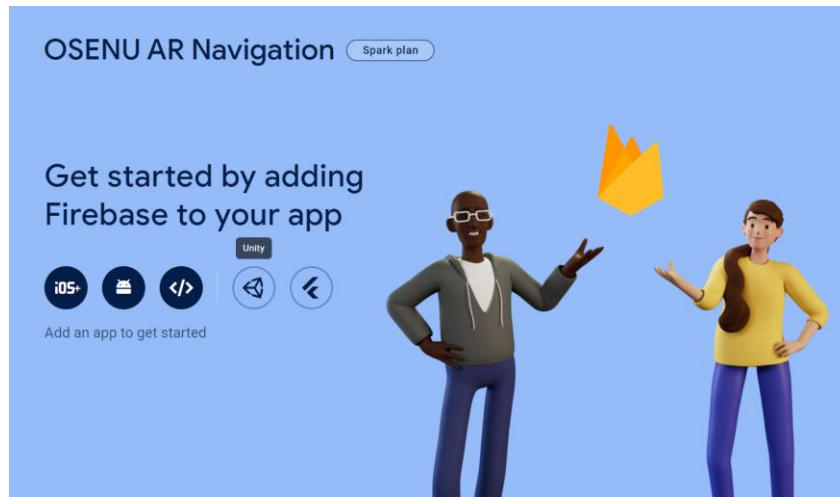


Рисунок 5.11 – Початок роботи з платформою Firebase

При додаванні підтримки Unity необхідно зареєструвати застосунок як застосунок Android (рис. 5.12). При цьому важливо вказати ім'я пакету Android так, як він називається в проекті Unity (рис. 5.13).

Register as Android app

Android package name ⓘ

com.osenu.osenunavigation

App nickname (optional) ⓘ

OSENUE AR Navigation

Register app

Рисунок 5.12 – Реєстрація застосунку Android на Unity у проекті Firebase

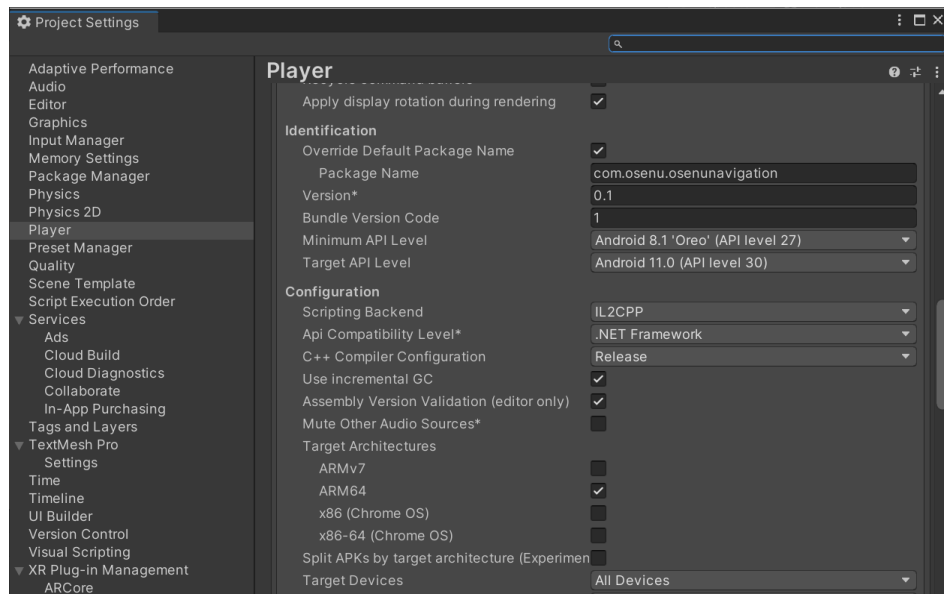


Рисунок 5.13 – Налаштування проекту в Unity

Після реєстрації застосунку Firebase створює файл конфігурації `google-services.json`, в якому прописані основні властивості проекту, ключі доступу тощо. Його необхідно додати до папки Assets в Unity. Наступний крок – завантаження Firebase Unity SDK. Це архів з пакетами різноманітних сервісів Firebase, які можна інтегрувати безпосередньо в проект Unity (рис. 5.14).

Имя	Размер	Сжат	Тип
..			Локальный диск
📦 FirebaseAnalytics.unitypackage	89 952 525	88 627 929	Unity package file
📦 FirebaseAppCheck.unitypackage	91 171 598	89 850 672	Unity package file
📦 FirebaseAuth.unitypackage	93 391 959	92 046 138	Unity package file
📦 FirebaseCrashlytics.unitypackage	89 160 109	87 842 140	Unity package file
📦 FirebaseDatabase.unitypackage	99 769 543	98 401 711	Unity package file
📦 FirebaseDynamicLinks.unitypackage	89 425 956	88 115 642	Unity package file
📦 FirebaseFirestore.unitypackage	109 905 305	108 463 658	Unity package file
📦 FirebaseFunctions.unitypackage	89 984 375	88 666 030	Unity package file
📦 FirebaseInstallations.unitypackage	89 535 707	88 229 246	Unity package file
📦 FirebaseMessaging.unitypackage	90 720 916	89 400 667	Unity package file
📦 FirebaseRemoteConfig.unitypackage	91 333 344	90 003 734	Unity package file
📦 FirebaseStorage.unitypackage	96 042 129	94 694 103	Unity package file

Рисунок 5.14 – Пакети Firebase Unity SDK

Після розархівування потрібні пакети необхідно імпортувати в проект Unity (рис. 5.15).

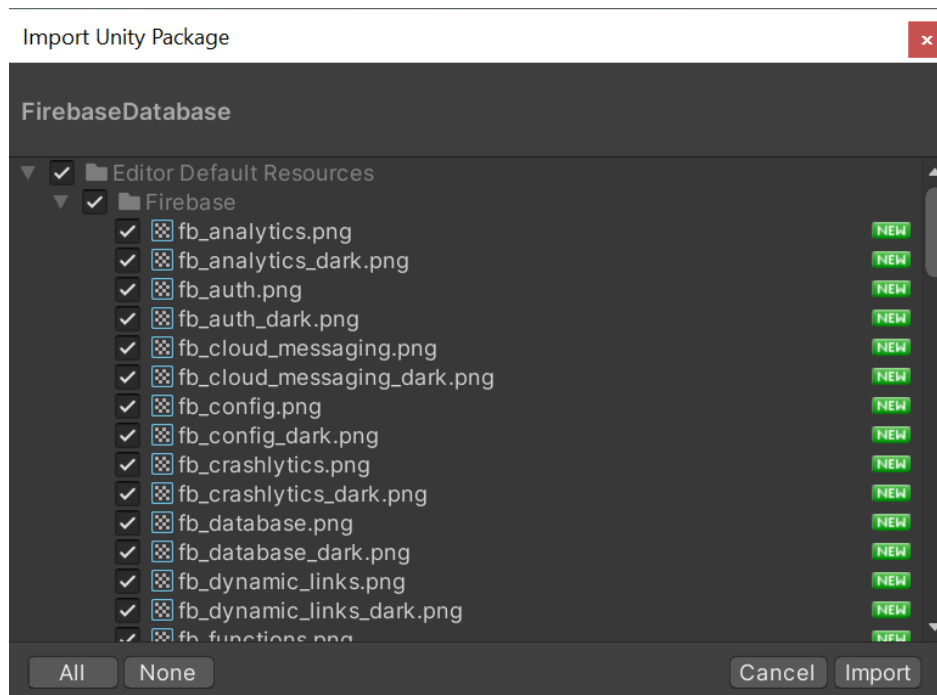


Рисунок 5.15 – Імпорт пакету Firebase Database в Unity

Firebase Realtime Database – це хмарна база даних в реальному часі, що надається Google в рамках платформи Firebase. Вона призначена для зберігання та синхронізації даних між різними пристроями та платформами в реальному часі. Дані зберігаються у вигляді JSON-подібних об'єктів, які містять пари ключ-значення. Це дозволяє легко структурувати дані та отримувати доступ до них. Дані організовані у вигляді деревоподібної структури, де кожен вузол може містити інші вузли або значення. Це дозволяє створювати структуровані дані, які можуть бути легко керовані та оновлювані.

Для створення бази даних Firebase необхідно на сторінці проекту Firebase в розділі Build обрати Realtime Database, вказати бажаний регіон для зберігання даних та налаштувати правила доступу (рис. 5.16).

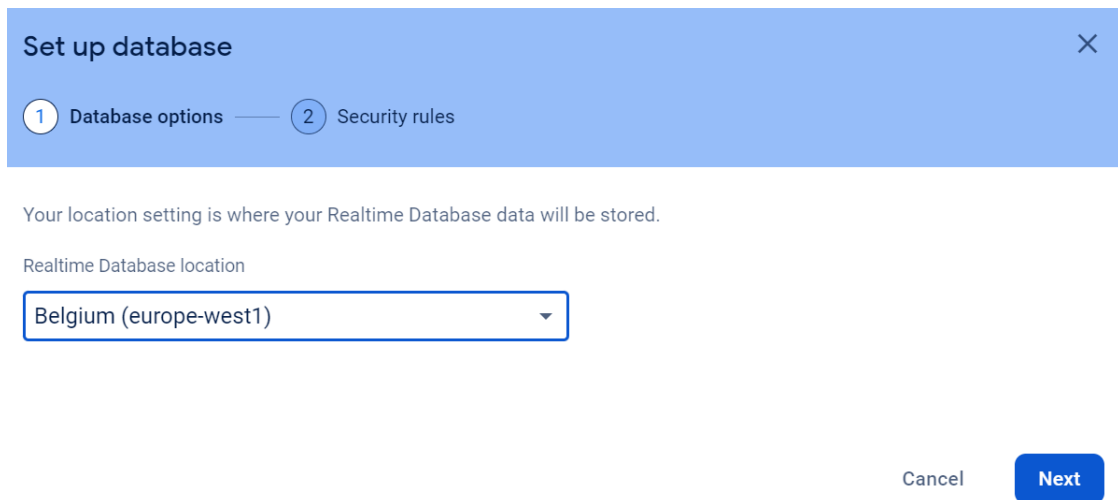


Рисунок 5.16 – Створення бази Firebase Realtime Database

База даних Realtime Database – NoSQL (Not Only SQL), тобто нереляційна. Вона відрізняється від традиційних реляційних баз даних за своєю структурою та моделлю зберігання даних. Відсутність строго визначених схем або таблиць дозволяє легко змінювати та додавати нові типи даних без необхідності зміни всієї структури бази даних. NoSQL бази даних забезпечують високу швидкість операцій завдяки спрощеному управлінню даними, особливо в великих обсягах.

Про кожну аудиторію зберігається наступна інформація: номер, назва (наприклад деканат або кафедра, або просто аудиторія з вказанням, до якої кафедри відноситься), корпус університету та поверх (рис. 5.17). Також зберігаються координати аудиторії (прив’язка до карти в проекті Unity).

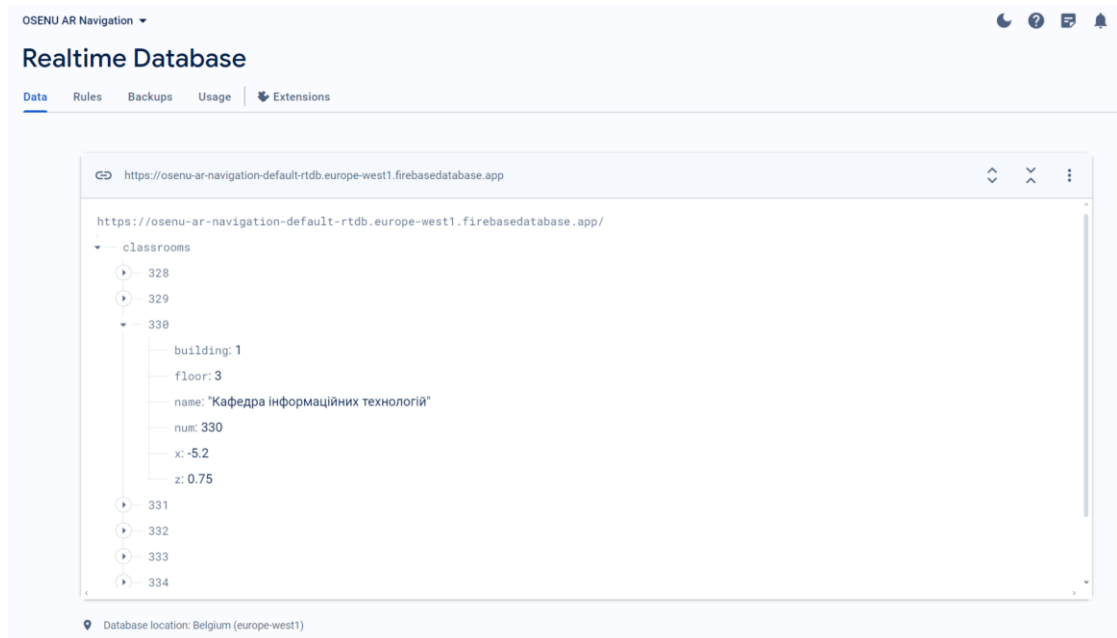


Рисунок 5.17 – Об’єкт бази даних Firebase Realtime Database

### 5.3 Реалізація навігації засобами Unity

Для реалізації навігації по університету було створено мапу. У кваліфікаційній роботі для демонстрації роботи застосунку використовується тільки коридор третього поверху першого корпусу, для якого була побудована наступна мапа (рис. 5.18).

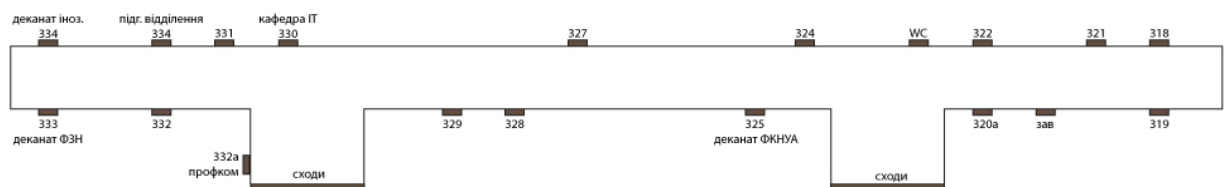


Рисунок 5.18 – Мапа частини третього поверху першого корпусу ОДЕКУ

Ця мапа була завантажена в проєкт Unity та на її основі був побудований NavMesh (рис. 5.19).



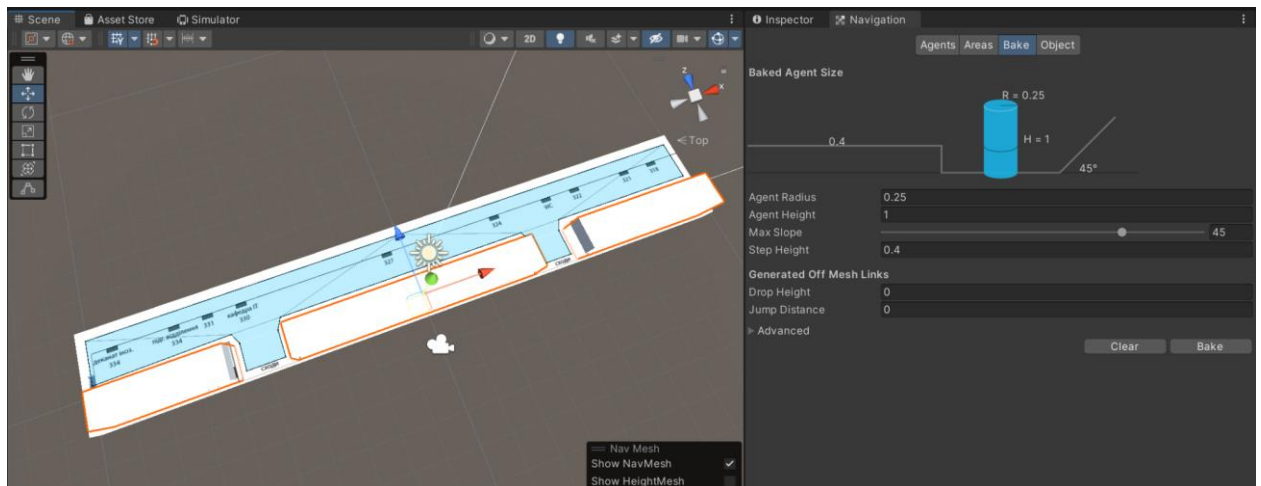


Рисунок 5.19 – NavMesh на основі мапи університету

NavMesh (Navigation Mesh) в Unity – це графічна топологія, яка використовується для програмування шляхів руху об'єктів у віртуальному середовищі. NavMesh визначає області, по яким об'єкти можуть переміщатися, а також визначає оптимальні шляхи для їхньої навігації [20].

NavMesh створює мережу, яка визначає прохідні та непрохідні області у віртуальному просторі. Це дозволяє об'єктам знаходити оптимальний шлях для переміщення. Unity має вбудовані алгоритми пошуку шляху, які використовують NavMesh для обчислення оптимального маршруту від однієї точки до іншої, уникнення перешкод та взаємодії з навколишнім середовищем. Unity надає широкі інструменти для створення та редагування NavMesh. Розробники можуть визначати області, де об'єкти можуть рухатися, встановлювати обмеження, додавати точки відпочинку та інші параметри для оптимізації навігації.

Також можна програмувати поведінку об'єктів на основі NavMesh за допомогою скриптів. Це дозволяє об'єктам автоматично знаходити шляхи та виконувати різноманітні дії на основі цих шляхів.

Об'єкт типу NavMeshAgent необхідно додати до об'єкту, що відповідає за переміщення користувача.

Оптимальний шлях між відправною точкою та пунктом призначення розраховується за допомогою методу NavMesh.CalculatePath. Цей метод дозволяє отримати оптимальний маршрут для переміщення об'єктів від однієї точки до іншої у віртуальному просторі на основі наявної мережі NavMesh.

Щоб створити NavMesh, необхідно обрати геометрію сцени, куди треба додати NavMesh, у вікні інспектора натиснути Додати компонент > NavMesh Surface та натиснути Bake. NavMesh генерується та відображається на сцені як синє накладання на базову геометрію сцени щоразу, коли вікно навігації відкрите та видиме.

Для відображення побудованого шляху використовується компонент LineRenderer. LineRenderer може бути використаний для візуалізації шляхів, які обчислюються в системі NavMesh. Коли об'єкт здійснює рух вздовж NavMesh, можна використовувати LineRenderer, щоб візуально показати його маршрут. Наприклад, шлях, обчислений за допомогою алгоритмів пошуку шляху в NavMesh, може бути відображений за допомогою LineRenderer, щоб користувач міг побачити шлях, яким йде персонаж (рис. 5.20). Можна налаштувати вигляд лінії: її колір, товщину, висоту над рівнем поверхні землі тощо.

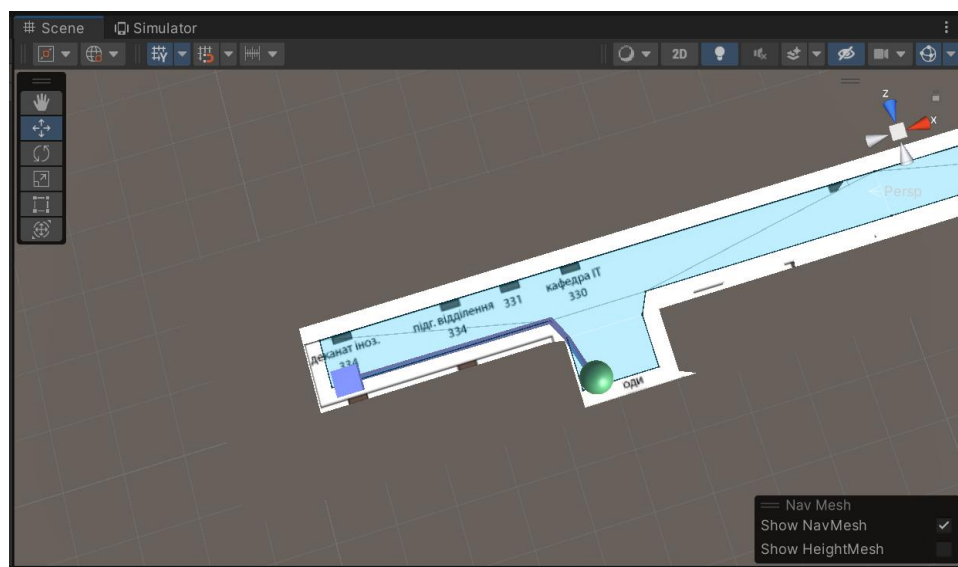


Рисунок 5.20 – Візуалізація лінії оптимального маршруту

#### 5.4 Опис та тестування застосунку

Головна сторінка застосунку «Навігатор по ОДЕКУ» – це сцена сканування. Користувач повинен навести камеру на номер найближчої аудиторії, щоб дізнатись своє розташування в університеті (рис. 5.21).

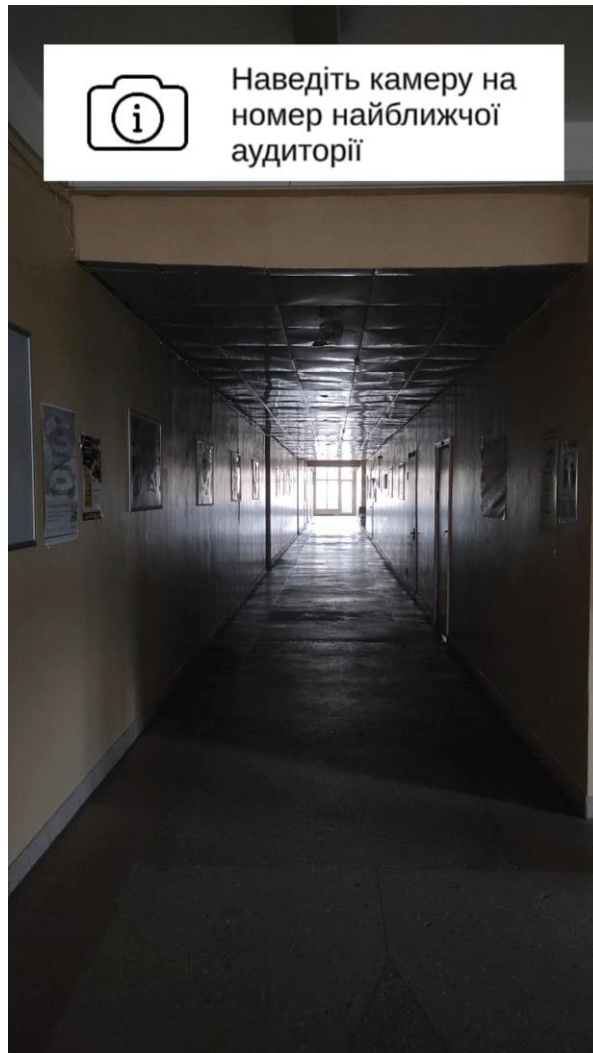


Рисунок 5.21 – Головна сторінка застосунку

Коли користувач наводить камеру на номер аудиторії, за допомогою Vuforia Engine Cloud Recognition розпізнається відповідний об'єкт з хмарної бази. Дані з Firebase Realtime Database завантажуються за ідентифікатором розпізнаного зображення та відображаються на екрані (рис. 5.22).

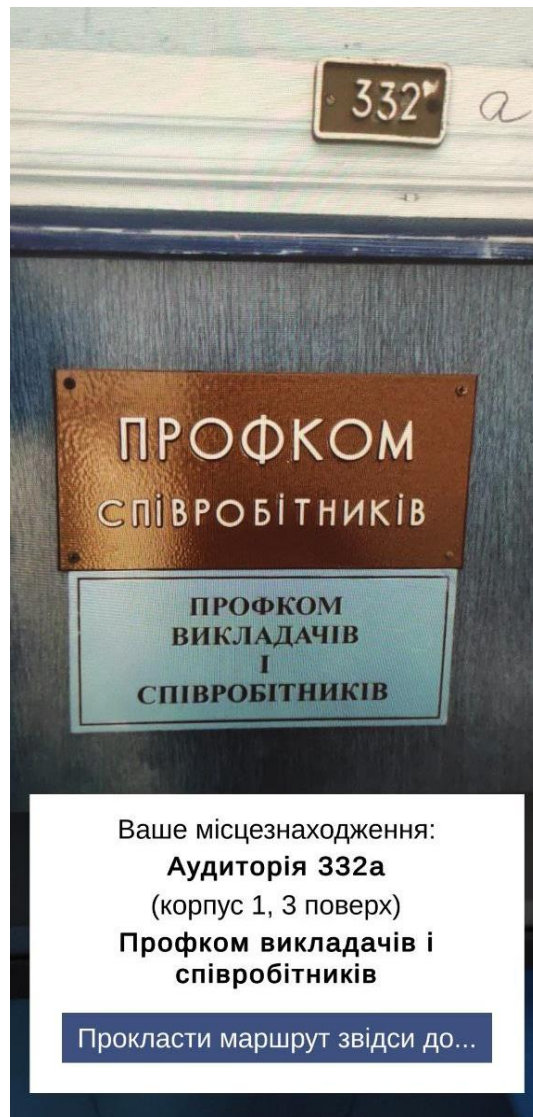


Рисунок 5.22 – Розпізнана аудиторія

Після того, як початкова аудиторія розпізнана, користувач може навести камеру ще раз на ту саму аудиторію або іншу, якщо вона була розпізнана помилково. У разі, якщо все вірно, користувач може натиснути кнопку Прокласти маршрут звідси до.... При натисканні цієї кнопки користувач переходить до іншої сцени, в якій відображається список цільових аудиторій. Цей список генерується динамічно з тих аудиторій, інформація про які є в базі даних Firebase Realtime Database (рис. 5.23)

## < Прокласти маршрут до аудиторії...

Введіть номер аудиторії або назву 

- 328 (Аудиторія кафедри інформаційних технологій)
- 329 (Аудиторія кафедри інформаційних технологій)
- 330 (Кафедра інформаційних технологій)
- 331 (Аудиторія кафедри інформаційних технологій)
- 332 (Аудиторія кафедри інформаційних технологій)
- 333 (Деканат факультету заочного навчання)
- 334 (Деканат по роботі з іноземними учнями)
- 334 (Підготовче відділення)

Рисунок 5.23 – Список доступних цільових аудиторій

Цільову аудиторію можна обрати, натиснувши на рядок з її номером та назвою. Серед аудиторій можна шукати потрібну за номером або її назвою. Після натискання на аудиторію, відкривається сцена навігації (рис. 5.24). В режимі AR користувач може бачити прокладений маршрут та інформацію про нього (відстань до цільової аудиторії, до повороту, напрям тощо).

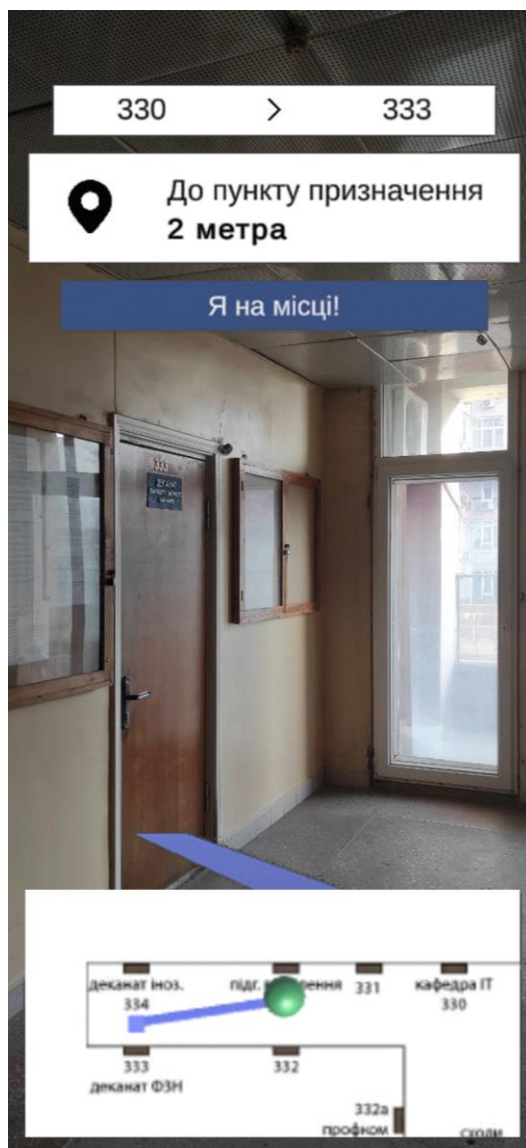


Рисунок 5.24 – Навігація по університету з використанням технологій доповненої реальності

Користувач може скасувати навігацію у будь-який момент, натиснувши на хрестик. Коли користувач підходить на певну відстань до цільової аудиторії, з'являється кнопка «Я на місці!». При натисканні на неї навігацію буде зупинено.

Застосунок «Навігація по ОДЕКУ» було протестовано в частині Одеського державного екологічного університету (третьй поверх першого корпусу), для якої була створена мапа. Застосунок був успішно використаний для навігації між аудиторіями, які там розташовані.

## ВИСНОВКИ

Метою даної кваліфікаційної роботи є розробка застосунку для навігації по Одеському державному екологічному університету з використанням технологій доповненої реальності для студентів, викладачів та інших працівників і відвідувачів університету.

У результаті виконання кваліфікаційної роботи бакалавра був розроблений мобільний застосунок «Навігатор по ОДЕКУ» з можливістю локалізації та навігації у реальному часі по будівлі університету. За допомогою застосунку користувачі можуть зручно отримати інформацію щодо розташування аудиторій ОДЕКУ та побудувати маршрут до необхідної аудиторії. Крім того, «Навігатор по ОДЕКУ» може зацікавити абітурієнтів, адже надає незвичайний та унікальний спосіб навігації по університету.

В ході роботи було проаналізовано предметну область, що дало можливість визначити актуальність розробки. Проведений порівняльний аналіз існуючих помічників-аналогів показав, що університети України не використовують технології доповненої реальності для навігації по своїй території, тож не мають повноцінного аналога розробленого «Навігатора по ОДЕКУ».

Для вибору інструментів розробки були проаналізовані різні засоби та платформи і було обрано платформу Vuforia Engine з огляду на її потужні алгоритми розпізнавання зображень, доступність на різних платформах, широкий спектр функцій для роботи з доповненою реальністю, включаючи розширені можливості відображення 3D об'єктів, анімації, а також взаємодії з ними, а також активну спільноту користувачів та детальну документацію. На етапі проектування були створені UML-діаграми варіантів використання, активності і послідовності, які дозволили унаочнити логіку роботи застосунку.

Розробка застосунку проводилась в середовищі Unity, яке є крос-платформним, найбільше підходить під потреби застосунку, що розроблюється, адже має багато вбудованих інструментів для роботи з доповненою реальністю, навігацією та побудовою маршрутів. Так, для розв'язання задачі пошуку

шляху був використаний інструмент Unity Navigation Mesh, який знаходить оптимальний шлях від початкової до цільової точки, використовуючи евристичну оцінку відстані від поточної точки до мети.

Для зберігання даних застосунку було обрано хмарну базу даних Google Firebase Database, яка надає широкий набір інструментів для розробки мобільних та веб-застосунків та також легко інтегрується з Unity та Vuforia Engine.

В роботі представлені можливості застосунку та приклади його використання. Було виконано тестування роботи на доданій ділянці мапи університету, яку в майбутньому планується розширити. Проблем при використанні виявлено не було.



**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. The Global Status Report 2017 [Електронний ресурс]. – Режим доступу: <https://globalabc.org/resources/publications/2017-global-status-report-buildings-and-construction> – Загол. з екрана. – Дата звернення: 25.10.2023
2. Patrick D. Archeoguide: System Architecture of a Mobile Outdoor Augmented Reality System. / Patrick D. Karigiannis, John N., 2009. – p. 345 – ISBN 9780769517810
3. Віртуальний університет [Електронний ресурс]. – Режим доступу: <http://virtual.regi.rovno.ua/megu/floor13/floor13.htm> – Загол. з екрана. – Дата звернення: 26.10.2023
4. KPImaps [Електронний ресурс]. – Режим доступу: <http://kpimaps.kpi.ua> – Загол. з екрана. – Дата звернення: 26.10.2023
5. Ng, Xin & Ning, Lim. (2020). Design of a Mobile Augmented Reality-based Indoor Navigation System. 1-6. 10.1109/ISMSIT50672.2020.9255121
6. More to explore with ARKit 6 [Електронний ресурс]. – Режим доступу: <https://developer.apple.com/augmented-reality/arkit> – Загол. з екрана. – Дата звернення: 01.11.2023
7. AR Core. Make the world your canvas [Електронний ресурс]. – Режим доступу: <https://developers.google.com/ar> – Загол. з екрана. – Дата звернення: 01.11.2023
8. Vuforia: Market-Leading Enterprise AR [Електронний ресурс]. – Режим доступу: <https://www.ptc.com/en/products/vuforia> – Загол. з екрана. – Дата звернення: 02.11.2023
9. Paul Mealy. Virtual & Augmented Reality [Текст] / P.Mealy. –John Willey & Sons, Inc, 2018. – 347 с.
10. Steve Aukstakalnis. Practical augmented reality [Текст] / S. Aukstakalnis. – Addison-Wesley Professional, 2016. – 489 с.
11. Karimi Hassan A. Indoor Wayfinding and Navigation / Hassan A. Karimi – Birmingham: Packt Publishing, 2013. – 367 p. – ISBN 9785978412723

12. Kaluža, Marin & Vukelic, Bernard. (2017). Analysis of an indoor positioning systems. Zbornik Veleučilišta u Rijeci. 5. 13-32. 10.31784/zvr.5.1.2.
13. Знаходження найкоротшого шляху в орієнтованому графі за алгоритмом Дейкстри [Електронний ресурс]. – Режим доступу: <https://www.mathros.net.ua/znahodzhennja-najkorotshogo-shljahu-v-orijentovanomu-grafi-za-algorytmom-dejkstry.html> – Загол. з екрана. – Дата звернення: 08.11.2023
14. Stuart Russell, Peter Norvig: Artificial Intelligence: A Modern Approach, 2004. – Prentice Hall, ISBN 3-8273-7089-2
15. Difference between BFS and DFS [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/difference-between-bfs-and-dfs> – Загол. з екрана. – Дата звернення: 08.11.2023
16. Pricing and Licensing Options [Електронний ресурс]. – Режим доступу: <https://developer.vuforia.com/library/faqs/pricing-and-licensing-options> – Загол. з екрана. – Дата звернення: 09.11.2023
17. Working with Device Databases [Електронний ресурс]. – Режим доступу: <https://developer.vuforia.com/library/objects/device-databases> – Загол. з екрана. – Дата звернення: 09.11.2023
18. Working with Cloud Databases [Електронний ресурс]. – Режим доступу: <https://developer.vuforia.com/library/cloud-recognition/cloud-databases> – Загол. з екрана. – Дата звернення: 09.11.2023
19. Accelerate and scale app development without managing infrastructure [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/products-build> – Загол. з екрана. – Дата звернення: 10.11.2023
20. Navigation System in Unity [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/540/Documentation/Manual/NavigationSystem.html> – Загол. з екрана. – Дата звернення: 12.11.2023