

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Штучного інтелекту _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський) _____
(рівень вищої освіти)

_____ Дослідження технології доповненої реальності _____
_____ та розробка інтерактивного музейного AR-додатку _____
(тема)

Виконав:
студент 2 курсу, групи СШМ-18-1
Карпенко Д.І. _____
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки _____
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного
інтелекту (СШ) _____
(повна назва освітньої програми)

Керівник _____ доц. Чала Л.Е. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов _____
(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 – Комп'ютерні науки _____
(код і повна назва)

Тип програми освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2019 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Карпенку Даніілу Ігоровичу _____

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження технології доповненої реальності та розробка інтерактивного музейного AR-додатку _____

затверджена наказом по університету від 04 листопада 2019 р. № 1623Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20__ р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження AR додатків, Unity documentation, C# documentation _____

4. Перелік питань, що потрібно опрацювати в роботі аналіз ринку мобільних AR додатків, аналіз дидактичних матеріалів по доповненій реальності, аналіз ігрового ядра Unity та його компонентів, розробка музейного AR додатку _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Ідентифікація картини «Богатирі», Рисунок 2 – AR візуалізація назви картини «Поцілунок», Рисунок 3 – Ідентифікація картини «Поцілунок», Рисунок 4 – Ідентифікація картини «Створення Адама», Рисунок 5 – Ідентифікація картини «Вавілонська Вежа», Рисунок 6 – Ідентифікація картини «Народження Венери»

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Аналіз предметної галузі	доц. Чала Л.Е.		10.11.2019
Формування вимог до додатку	доц. Чала Л.Е.		15.11.2019
Розробка додатку	доц. Чала Л.Е.		23.11.2019
Аналіз готового продукту	доц. Чала Л.Е.		24.11.2019

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	04.11.19–10.11.2019	виконано
2	Аналіз ринку AR додатків	10.11.2019–14.11.2019	виконано
3	Формування вимог до додатку	15.11.2019	виконано
4	Засоби розробки додатку	16.11.2019	виконано
5	Програмна реалізація додатку	17.11.2019–23.11.2019	виконано
6	Аналіз готового продукту	24.11.2019	виконано
7	Написання пояснювальної записки	25.11.2019–10.12.2019	виконано
8	Попередній захист	17.12.2019	виконано
9	Захист перед ЕК	18.12.2019	

Дата видачі завдання 04 листопада 2019 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Чала Л.Е.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить 92 с., 34 рис., 2 дод., 21 джерел.

ДОДАТОК, ASSETS, AUGMENTED REALITY, UNITY
VUFORIA

Об'єкт дослідження – методи розробки додатків технології доповненої реальності для платформ Android і IOS.

Мета роботи – розробка та реалізація AR проекту для мобільних пристроїв, за допомогою використання графічного та фізичного ядра Unity.

Області застосування – музейні виставки.

Методи дослідження – аналіз технічної літератури, аналіз методів створення технології доповненої реальності, ринку музейних додатків та практична реалізація.

Результат атестаційної роботи – продуктом проведеної роботи є мобільний додаток, що пройшов апробацію.

ABSTRACT

The explanatory note to the performance appraisal contains 92 pages, 34 figures, 2 appendix, 21 sources.

APPLICATION, ASSETS, AUGMENTED REALITY, UNITY VUFORIA

The object of research is the methods of developing applications of augmented reality technology for the Android and IOS platforms.

The purpose of the project is to develop and implement an AR project for mobile devices, using the Unity Graphic and Physical kernel.

Areas of application – museum exhibitions.

Research methods – analysis of technical literature, analysis of methods of creating augmented reality technology, the market of museum applications and practical implementation.

The result of the certification work – the product of the work was a mobile application that has been tested.

РЕФЕРАТ

Пояснительная записка к аттестационной работы содержит 92 страницы, 34 рисунка, 2 приложения, 21 источник.

ПРИЛОЖЕНИЕ, ASSETS, AUGMENTED REALITY, UNITY
VUFORIA

Объект исследования – методы разработки приложений технологии дополненной реальности для платформ Android и IOS.

Цель работы – разработка и реализация AR проекта для мобильных устройств, с помощью использования графического и физического ядра Unity.

Области применения – музейные выставки.

Методы исследования – анализ технической литературы, анализ методов создания технологии дополненной реальности, рынка музейных приложений и практическая реализация.

Результат аттестационной работы – продуктом проведенной работы стало мобильное приложение, прошедшее апробацию.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі та постановка задачі.....	10
1.1 Аналіз предметної галузі.....	10
1.2 Маркетинговий аналіз.....	21
1.3 Постановка задачі.....	33
2 Теоретичні дослідження.....	35
2.1 Принципи побудови доповненої реальності.....	35
2.2 Типи систем доповненої реальності.....	38
2.3 Технологія QCAR.....	41
2.4 Алгоритм трекінга.....	45
3 Експериментальні дослідження.....	52
4 Розробка музейного AR-додатку	56
4.1 Засоби розробки.....	56
4.1.1 Unity.....	56
4.1.2 Vuforia.....	65
4.2 Розробка додатку.....	69
5 Аналіз реалізованого продукту.....	78
Висновки.....	83
Перелік посилань.....	85
Додаток А.....	87
Додаток Б.....	91

ВСТУП

Розвиток технологій у 21 столітті є невимовно швидким. Кожен рік на світових конференціях присвячених технологіям презентують величезну кількість нових речей. Їх настільки багато і їх масштаб може бути настільки великим, що людство може бути просто не готово до прийняття тієї чи іншої технології. Через те, що не готові люди, а найголовніше, не готові розробники, багато сміливих рішень так і залишаються нерозкритими до кінця. Саме тому навчальну систему необхідно постійно вдосконалювати, щоб вона встигала готувати спеціалістів гідних свого часу. Однією з таких технологій є Доповнена Реальність (AR). Ця технологія має безпосередній вектор у майбутнє. Її застосування можна легко представити у навігації. Наприклад, використовуючи смартфон (обладнаний камерою), можна буде без проблем знайти необхідний шлях, навіть якщо у вас є проблеми з орієнтацією на карті. Такий результат може бути досягнений завдяки синхронній роботі систем глобального позиціонування, завантаженим на смартфон карти, і алгоритмам, що зможуть прив'язати дані з карти на зображення, яке отримується з камери смартфона. Як результат ми зможемо побачити виділений шлях (наприклад, кольоровою лінією), що буде проектуватися у реальність і вказуватиме необхідну користувачеві дорогу.

Сьогодні можна сміливо говорити про стрімке старіння традиційних віконних графічних інтерфейсів, керованих клавіатурою і мишкою, що дозволяє за допомогою всесвітньої павутини користувачеві отримувати інформацію про навколишній світ і події, що відбуваються в ньому. Стрімкий розвиток інтерактивних мультимедійних технологій вимагає появи нових інтерфейсів впливу типу «Навколишній світ – машина – людина». Одним із прикладів новітніх інтерфейсів взаємодії між навколишнім світом, машиною і людиною є технологія, що отримала назву «Доповнена реальність».

«Доповнена реальність» в широкому сенсі – це комплекс технологій, що дозволяють разом з відчуттями, які надходять з реального світу, отримувати додаткові відчуття уявних об'єктів, зазвичай допоміжні, інформативної властивості.

Технології доповненої реальності дозволяють демонструвати абсолютно нові властивості об'єктів і отримувати нові відчуття від звичних реальних речей, використовуючи стандартний персональний комп'ютер і стандартні периферійні пристрої. Тому кількість областей, де можна (і потрібно) застосовувати ці технології, безмежна. На сьогоднішній день одним з ключових методів доповненої реальності є маркерна ідентифікація зображення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

У сучасному світі людина щодня стикається з графічними образами, створеними за допомогою комп'ютерів. Вони грають дуже важливу роль в нашому житті, допомагаючи виконувати щоденні завдання, спрощувати рутинні процеси і робити пошук. Але комп'ютери не тільки дають можливість спілкуватися з віртуальним світом, вони також є інтерфейсом, який дозволяє отримувати розширені дані про реальний світ.

Доповнена реальність – революційно нова технологія в області взаємодії з цільовою аудиторією за допомогою доповнення існуючої реальності образами та інформацією з віртуальної реальності. Суть технології полягає в тому, що при наведенні камери смартфона або планшета на тригер доповненої реальності користувачеві відкривається інтерактивний контент, тобто користувач бачить віртуальний 3D об'єкт з анімацією або відео, а також інформаційний блок (в залежності від конкретного випадку), яким може управляти в реальному просторі. За допомогою доповненої реальності виробники зможуть вигідно підкреслити гідності своєї продукції, надати користувачеві інформацію про продукцію безпосередньо в момент вибору і в момент прийняття рішення про покупку. Шари доповненої реальності дають можливість користувачам досліджувати повнофункціональну 3D-модель будь-якої продукції або наочно демонструють весь цикл надання тієї чи іншої послуги. При цьому користувач отримує максимально реалістичні відчуття від продукту або послуги, отримує інформацію, яка підтверджує якість товару або послуги, а також інформацію про компанію-виробника. Ця технологія відкриває необмежену кількість нових можливостей у просуванні бренду, в маркетингових і PR-комунікаціях.

Технологія доповненої реальності (augmented reality, AR) забезпечує найбільш інтуїтивну і зрозумілу форму взаємодії людини з реальним світом через комп'ютер. Реальний світ поєднується на мобільному пристрої або комп'ютері з електронними даними, що накладаються поверх реального відеозображення.

Доповнена реальність – це технологія інтерактивної візуалізації, яка доповнює зображення реального світу віртуальними елементами. Щоб побачити доповнену реальність, необхідні веб-камера комп'ютера або камера мобільного пристрою (смартфон, планшет, AR-окуляри), а також спеціальний додаток, який накладає цифрову інформацію (тривимірні моделі, відео, аудіо, тексти) на зображення реального світу, що отримується з камери, і виводить результат на екран.

Сам термін «доповнена реальність», імовірно, був введений дослідником корпорації Boeing Томом Коделом (англ. Tom Caudell) в 1990 році. Існує кілька визначень доповненої реальності: дослідник Рональд Азума в 1997 році визначив її як систему, яка має наступні можливості:

- поєднує віртуальне і реальне;
- взаємодіє в реальному часі;
- працює в 3D.

У 1994 році Пол Мілгром і Фуміо Кісін описали континуум «віртуальність-реальність» – простір між реальністю і віртуальністю, між якими розташовані доповнена реальність (ближче до реальності) і доповнена віртуальність (ближче до віртуальності). Доповнена реальність – результат додавання до більш прийнятних елементів реального світу уявних об'єктів (зазвичай в якості допоміжної інформації).

Іноді як синоніми використовують терміни «розширена реальність», «поліпшена реальність», «збагачена реальність», «збільшена реальність». Правда, таке використання названих термінів в загальному випадку неправильно, – терміни «розширена реальність», «збільшена реальність», «збагачена реальність» застосовні лише для визначення певних форм і

аспектів практичного застосування доповненої реальності, тоді як застосовність терміну «поліпшена реальність» зовсім сумнівна.

Доповнена реальність, або AR (англ. Augmented reality) – технологія, істотно розширює область даних сприймаються користувачем. Актуальність цього питання пов'язана з:

- доступністю інформації в реальному часі;
- інтерактивністю;
- «wow»-ефектом – неординарним способом представлення інформації, що дозволяє привертати увагу, а також посилювати запам'ятовування;
- реалістичністю;
- іноваційністю.

Кількість компаній, що використовують AR, зростає з кожним днем. За оцінками аналітиків, сукупний обсяг ринку доповненої реальності вже перевищив 5 млрд. доларів. Markets and Markets оцінює середньорічне зростання глобального ринку з 2013 по 2018 рр. в 132% і більше [1].

Основні характеристики, властиві середовищу доповненої реальності:

- сполучуваність – можливість синтезу з об'єктивною реальністю і суспільними процесами;
- штучність – її існування обумовлено і підкріплено інформаційно-комп'ютерними технологіями;

Доповнена реальність знаходить застосування в найрізноманітніших областях. Перші прийоми доповненої реальності, хоча такої назви ще не було, знайшли широке застосування у фантастичній літературі і пов'язаним з нею образотворчому мистецтві в жанрі альтернативної історії, а також в продукції телебачення і кінофільмах, де змішані та взаємодіють реальні об'єкти і персонажі з такими ж, створеними мультиплікацією і комп'ютерною графікою.

Сьогодні можна виділити декілька найбільш перспективних ринкових ніш, де використання AR найбільш добре проглядається. Це, перш за все,

маркетингові комунікації, продажу, навчання і обслуговування, військова сфера, сфера медицини, сфера розваг і дозвілля.

Застосування доповненої реальності в PR-комунікаціях дозволяють найбільш ефективно з'єднати споживачів з компанією і вибудувати довгострокові відносини між ними.

Доповнена реальність – революційно нова технологія в області взаємодії з клієнтами. При наведенні камери смартфона або планшета на тригер доповненої реальності користувачеві відкривається інтерактивний контент, він бачить віртуальний 3D об'єкт з анімацією або відео, яким може керувати в реальному просторі. За допомогою доповненої реальності виробники зможуть вигідно підкреслити переваги своєї продукції, даючи можливість покупцеві досліджувати повнофункціональні 3D-моделі і отримати максимально реалістичні відчуття від продукту. Дана технологія відкриває необмежену кількість нових можливостей у просуванні бренду і залученні нових клієнтів.

Доповнена реальність дозволяє побудувати взаємодію безпосередньо в момент контакту з рекламним повідомленням. При цьому користувачі зі сторонніх спостерігачів перетворюються в активних учасників процесу, залучаючи і оточуючих теж. Використовуючи ігрові механіки із залучення споживачів, доповнена реальність дозволяє значно збільшити тривалість контакту та запам'ятовуваність продукту.

Що ж стосується сфери продажу, доповнена реальність – це інструмент, який дозволяє максимально скоротити тривалість циклу продажу для великого спектру товарів.

У сфері навчання на сьогоднішній день традиційна форма навчання все більше поступається місцем сучасним методам, зокрема, дистанційній формі навчання [2]. Для навчального процесу необхідні наочні посібники і проведення лабораторних експериментів, а в разі дистанційної освіти забезпечити всіх студентів посібниками неможливо. І в цьому випадку

абсолютно очевидним є застосування програмних продуктів, що використовують доповнену реальність.

У сфері охорони здоров'я технологія доповненої реальності використовується найактивніше. Створюються спеціальні програмні забезпечення для підвищення професійних навичок лікарів, що призводить до зменшення операцій з летальним результатом. Наприклад, коли хірург видаляє пухлину, на зображення на ендоскопі накладається картинка, яку зробили попередньо, за допомогою цього хірург може бачити, де всередині органу знаходиться пухлина, і внаслідок цього звести втрати здорової тканини органу пацієнта до мінімуму. У сфері охорони здоров'я можливості доповненої реальності дозволяють моделювати ситуації і досліджувати об'єкти, що дає максимально практичну орієнтованість майбутнім медикам. Студенти-медики можуть вивчати будову людського тіла, проходити тести і практикуватися, перебуваючи за тисячі кілометрів від лабораторії.

Застосування доповненої реальності у військовій сфері також дуже перспективно. Створення військових тренажерів, пояснення теорії тактики і стратегії військової справи, моделювання ситуацій дозволяють військовим збільшити ефективність освоєння будь-якої інформації.

У сучасній військовій техніці використовуються індикатори, які надають важливу інформацію для швидкого аналізу ситуації, розрахунку траєкторії пострілу, розрахунку траєкторії приземлення у пілотів. Таким чином підвищується ефективність військових та військова міць країни.

Використання технологій доповненої реальності в сфері дозвілля і розваг безмежна. Програми, що використовують AR, можуть бути дуже видовищними, цікавими і незвичайними.

Технологія доповненої реальності в мистецтві: при використанні технології доповненої реальності, картини оживають, експонати рухаються, кімнати змінюють вигляд, і це далеко не кінець дивного списку. Внаслідок таких нововведень, інтерес до відвідування музеїв у населення зростає, що призводить до культурного розвитку країни.

Технологія доповненої реальності в поліграфії. За допомогою браузерів доповненої реальності можна візуалізувати цифрові об'єкти через маркери (певні зображення) поміщені в друковану продукцію, таким чином, поліпшується сприйняття тексту.

Існує безліч програмних продуктів для мобільних пристроїв, які дозволяють за допомогою доповненої реальності отримати необхідні відомості про оточення: браузери доповненої реальності і спеціалізовані програми для окремих сервісів, компаній або моделей. Саме поширення доповненої реальності і зростаюча популярність технології серед споживачів пов'язані з тим, що обчислювальна потужність і набір датчиків в апаратних платформах смартфонів і планшетів дозволяють виробляти накладення будь-яких цифрових даних на одержуване в реальному часі зображення, яке формується камерами пристрою.

З ростом ступеня проникнення комп'ютерних технологій у повсякденне життя класичні інтерфейси починають не справлятися з покладеним на них завданням і стають вузьким місцем у взаємодії людини з комп'ютерною технікою. У зв'язку з цим в даний час поширення отримують нові, революційні способи взаємодії людино з віртуальним світом. Одним з видів таких нових інтерфейсів є інтерфейси, що базуються на технологіях змішаної реальності.

І віртуальна, і доповнена реальність відносяться до технологій змішаної реальності.

Технологія доповненої реальності є однією з найбільш ефективних технологій, що відносяться до змішаної реальності, так як вона має найбільшу зв'язок з навколишнім світом.

Крім названих плюсів і ефекту від використання доповненої реальності, варто згадати про матеріальні плюси: відпаде необхідність у виробництві і використанні будь-якого виду поліграфії, друкованої продукції. Камера, двовимірний маркер, який розміщується перед камерою

для зчитування та аналізу з нього інформації, – це все, що необхідно для отримання ефекту доповненої реальності.

В даний час великі інформаційно-технічні кампанії працюють в напрямку впровадження доповненої реальності в повсякденне життя. Цей факт підтверджує те, що доповнена реальність, так чи інакше, проникла в усі сфери життєдіяльності людини, створюючи особливу інноваційну комунікацію за допомогою гаджетів і програмного забезпечення.

Використання технологій додаткової реальності відкрило нову сторінку в маркетингу, вчинила прорив в комерційній сфері. В першу чергу, це певне програмне (апаратне) забезпечення, яке саме по собі є інноваційним продуктом, в другу чергу – використання технологій доповненої і віртуальної реальності дозволяє безпосередньо впливати на збільшення продажів практично будь-якого продукту.

Технологія доповненої реальності значно впливає на всі комунікації соціальних суб'єктів. Так як сьогодні все частіше традиційні види комунікації зі споживачем замінюються на більш сучасні – інтерактивні, то саме технологія доповненої реальності дозволяє здійснювати максимально інтерактивну комунікацію. Інтерактивність передбачає двосторонній взаємозв'язок між об'єктами і суб'єктами комунікації. А що стосується інтерактивної реклами, вона дає клієнтам нові можливості для взаємодії з брендом.

Величезний перелік способів, методів, технологій продажів, величезна кількість видів реклами, способів залучення уваги до продукту і стимулювання попиту привело до того, що в комерційній сфері все складніше донести інформаційний посыл до цільової аудиторії, все складніше акцентувати увагу саме на своєму товарі або послугі. Саме жорстка ринкова конкуренція, боротьба за клієнта, необхідність виділитися серед величезного асортименту товарів і послуг вимагає нових високотехнічних мультимедійних інструментів для посилення ефекту інформаційних

суспільств. Дана ситуація в комерційній сфері вимагає інноваційного підходу, а, значить, засобів інноваційних комунікацій.

Розглянемо докладніше кілька додатків для створення доповненої реальності. Наприклад, додаток «Layar» дозволяє швидко прив'язати контент до довільних міток (зображеннях). При цьому не вимагається ніяких специфічних навичок. Необхідно нанести за допомогою даної програми мітку на будь-який носій і завантажити додаток на смартфон. За допомогою цього додатка можна здійснювати покупки будь-якого продукту або послуги в рекламному каталозі, прослуховувати музику, наводячи екран смартфона на фото виконавця, замовляти їжу з рекламної брошури тощо. Ще одним подібним інструментом є додаток «Junao». За допомогою цього додатка користувач може отримувати інформацію про своє місцезнаходження та будь-який об'єкт реальності на вулицях міста. Програма зв'язується з базою даних і надає інформацію про всі події, які відбуваються всередині того чи іншого будинку.

Існує багато способів створювати доповнену реальність. Всі вони серйозно відрізняються за своїм функціоналом. Найбільш поширеним з яких є QR-код, за допомогою якого здійснюється швидкий перехід до інформації або джерел інформації. Наступним способом є спосіб шарів – спосіб, коли до об'єктів місцевості прив'язуються інформаційні шари і за допомогою них відбувається перехід до джерел інформації. Третім способом є аура. При наведенні смартфона на будь-який носій інформації, розпізнається картинка і поверх картини з'являється аура – це може бути відео, інформаційне повідомлення або фотографія, а також будь-який тривимірний об'єкт.

Зупинимося докладніше на технічних засобах інноваційних комунікацій, що базуються на технології доповненої реальності. Наведемо короткий огляд існуючих технічних рішень.

Окуляри доповненої реальності «GOOGLE Glass» розробляються компанією Google і є автономним пристроєм, керувати яким можливо за допомогою голосу і жестів.

Принцип роботи окулярів доповненої реальності набагато відрізняється від пристроїв віртуальної реальності. Якщо окуляри віртуальної реальності лише умовно носять свою назву «окуляри», то даний пристрій дійсно є окулярами.

Пристрій доповненої реальності створює зображення, орієнтуючись на ті речі, які воно спостерігає перед собою. Отже, від того, куди спрямований погляд користувача, залежить робота обладнання. Перед очима користувача знаходиться невеликий інтерактивний дисплей, який відображає і виводить всю необхідну для вас інформацію. У режимі активного користування батареї пристрою вистачає на цілий день. Крім цього, в Google Glass є жорсткий диск на 12Гб і відеокамера. Google Glass є найдорожчими окулярами доповненої реальності. На даний момент компанією продається третє покоління девайса. Щороку гарнітура поліпшується з урахуванням побажань користувачів.

«Microsoft HoloLens» – шолом доповненої реальності. Користувач даного гаджета бачить весь навколишній світ, але при цьому на навколишню дійсність за допомогою скляних лінз накладаються об'ємні голограми, графіки, різні зображення. Особливістю є те, що тільки користувач бачить дані зображення за допомогою шолома. Але при цьому, якщо кілька користувачів надінуть шоломи, то одночасно зможуть бачити загальний потік інформації, образів і зображень [3].

Дана розробка має на увазі використання голограм, які інтегруються з об'єктами навколишнього середовища. Користувач може не тільки спостерігати за тим, що відбувається, а й взаємодіяти з голограмами інтерфейсу за допомогою рук.

Також за допомогою даного пристрою доповненої реальності можливо проектувати відео або будь-який інший презентаційний матеріал на будь-яку поверхню. Пристрій доповненої реальності є повністю бездротовим, легким і зручним в експлуатації. Також окуляри відтворюють об'ємний звук і мають вбудовані камери. Найбільш перспективною сферою застосування є

архітектура і будівництво. Для презентацій масштабних будівельних об'єктів даний шолом представляє особливу цінність, так як дозволяє розглянути всі особливості архітектурних рішень, найменші об'єкти споруд будь-якої складності, донести інформацію про користь і необхідність тих чи інших рішень в різних будівельних конструкціях.

Smart Glasses M100. Єдиний WQVGA-дисплей окулярів виконаний з співвідношенням сторін 16: 9 і сприймається користувачем як 4-дюймовий дисплей на відстані 35 см. Дисплей досить яскравий, що дозволяє використовувати його при досить яскравому світлі. Камера знімає в розрішенні 1280 x 720. Процесор – OMAP4430 частотою 1 ГГц, об'єм оперативної пам'яті – 1 ГБ, флеш-пам'яті – 4 ГБ. На пристрої є кілька фізичних кнопок: «включення», «вибір» і управління гучністю. В теперішній момент цей гаджет не може похвалитися можливістю доповнення реальності, однак специфіка роботи Vuzix дає можливість припустити, що цей комп'ютер еволюціонує в AR-гарнітуру разом з аналогом Google [4].

Окуляри доповненої реальності SpaceGlasses META.01. Цей гаджет комплектується інфрачервоними камерами для розпізнавання рухів. Плюс Google Cardbox, фактично є картонним (в деяких варіантах – пластиковим) чохлом для смартфона. Визначення положення і формування доповненої реальності здійснюється засобами програмно-апаратних можливостей смартфона.

Magic Leap One Creator Edition – окуляри доповненої реальності. Цей продукт один з найбільш очікуваних продуктів серед технічних рішень, що реалізують функцію доповненої реальності. За попередніми даними генерального директора «Magic Leap» Роні Абовіц це буде гаджет з найвищими технічними характеристиками, за продуктивністю не поступаються «MacBook Pro».

Таким чином, сучасний світ вимагає інтерактивності в комунікації, динаміки і, як наслідок, особливого просторового бачення. Саме ці вимоги

дозволяє реалізувати технологія доповненої реальності як інструмент інноваційних комунікацій.

Ще однією особливістю доповненої реальності є простота використання. Для того, щоб бачити віртуальні шари, що доповнюють реальність, не обов'язково мати спеціальні окуляри або шолом, досить звичайного смартфона і підключення до мережі Інтернет.

Підводячи підсумки першого розділу, варто відзначити, що останнім часом увага дослідників в усьому світі прикута до інновацій. Поняття «інновація», «інноваційні процеси», «нововведення» використовуються практично в кожному тексті. Питання впровадження інновацій в усі сфери життєдіяльності людини стоїть дуже гостро. Не залишається осторонь і комунікаційна сфера.

Таким чином, з'являються нові концепції, що інтегрують усі суспільні комунікаційні сфери. Однією з таких концепцій є концепція інтегрованих маркетингових комунікацій.

Зміни в суспільстві, зміни в інформаційному просторі роблять сильний вплив на комунікації. Зниження ефективності традиційних технологій і методів PR, реклами, маркетингу веде до необхідності вироблення нових рішень і застосування нових технологій для досягнення більшої ефективності комунікацій. Саме тому у соціальних суб'єктів виникає потреба використання інноваційних комунікацій, тобто інтегрованих маркетингових комунікацій з використанням інноваційних технологій, за допомогою яких на практиці реалізується і застосовується нова ідея, якісно відмінна від існуючих форм.

Технології доповненої і віртуальної реальності це інтерактивні інноваційні технічні рішення, які відкривають світ нових можливостей для соціальних суб'єктів. Використання даних технологій як засобів інноваційних комунікацій безмежно. Ефективність використання даних технологій досягається за рахунок «ефекту занурення» і максимального залучення користувача в процес. Поняття «віртуальна реальність» визначається

вченими як «комп'ютерна система, що застосовується для створення штучного світу, користувач якої відчуває себе в цьому світі, може бути керованим в ньому і маніпулювати його об'єктами». Поняття «доповнена реальність» визначається як «технологія інтерактивної візуалізації, яка доповнює зображення реального світу віртуальними елементами».

1.2 Маркетинговий аналіз

Хоча існує безліч інноваційних способів використання доповненої реальності, можна виділити чотири типи додатків, в яких найчастіше використовуються AR технології: рекламні та комерційні, розважальні та освітні, спеціалізовані та мобільні програми.

Доповнена реальність в основному використовується маркетологами для просування нових продуктів. Більшість додатків використовують маркери, які користувач розміщує перед камерою, спеціальне програмне забезпечення розпізнає їх і доповнює будь-якою інформацією. Так, наприклад, проводилася досить інноваційна реклама телеканалу National Geographic в одному з супермаркетів. Для такої ефектної реклами, посеред холу магазину був розміщений екран, до якого була приєднана камера і комп'ютер. Перед монітором знаходився маркер, поряд з яким мали знаходитися люди. Після того, як люди підходили до маркеру, на ньому починали з'являтися різні тварини (звісно, відображення цього дива йшло на екрані). Ці тварини починали бігати серед людей, взаємодіяти з ними і навіть грати.

Доповнена реальність також є вирішенням проблеми в побудові та презентації макетів. Дійсно, виробники стикаються з необхідністю, під час дорогого виробництва продукту, виявляти, до комерціалізації, необхідні зміни або просто продемонструвати макет і визначити, чи відповідає продукт очікуванням. Якщо приймається рішення про внесення змін, зазвичай доводиться виготовляти новий прототип, що означає додаткові витрати часу і

грошей. Група при Інституті Промислової Технологій та Автоматики (ІТІА) Національної Ради Досліджень (CNR) Італії в Мілані працює з AR і VR системами в якості інструменту для віртуального прототипування. ІТІА-CNR-бере участь у дослідженнях промислових систем і додатків, які за допомогою VR і AR, в реальному часі, використовують 3D-моделювання для тестування продуктів, їх розвитку та оцінки.

Розважальна та освітня галузь включає в себе «культурні» додатки для огляду визначних пам'яток і музейні довідники. Ігрові програми являють собою традиційні ігри з додаванням AR інтерфейсів. Так само в цю групу можна включити мобільні додатки, які використовують AR для розважальних або освітніх цілей. Серед «культурних» додатків, існує кілька систем, в яких AR використовуються для реконструкції стародавніх руїн або для надання віртуальної історичної довідки.

AR може також використовуватися для навчання. У галузі освіти системи доповненої реальності можуть мати широке застосування в багатьох областях, таких як історія, математика тощо. Наприклад, Mark Billingham розробив Magic Book, книга, сторінки якої включені прості технології AR, для того, щоб зробити читання більш захоплюючим.

Мобільні AR додатки є одними з небагатьох програм з доповненою реальністю, які можуть дійти до широкої публіки. Тим не менш, навіть ці додатки стикаються з деякими проблемами. Існують, проблеми, пов'язані з GPS датчиками, які можуть бути не досить точними, коли необхідно точно розмістити віртуальні моделі доповненої реальності. Є проблеми, пов'язані з обмеженими можливостями апаратної частини, так як потрібна висока потужність для обробки зображень. Що стосується програмної сторони, існують проблеми з доступом до відео API, пов'язані із закритістю. Тим не менш, сучасні тенденції призводять до все більшої популяризації відкритих систем, що спрощує розробку додатків з доповненою реальністю, так як є повний доступ до пристрою. А також кроки, з якими виробники підвищують потужності своїх пристроїв, наближають до часу, коли зникне потреба у

хвилюванні на рахунок обмеженості, які виникають за рахунок характеристик смартфонів.

Є також кілька систем, які використовують AR для музейних довідників. Автори додатку використовують переваги доповненої реальності і в якості інтерфейсу надає ефективну взаємодію з користувачем через мультимедійні презентації, природну і інтуїтивно зрозумілу техніку і низькі експлуатаційні витрати на придбання технології для музею у разі використання смартфона в якості інтерфейсу. І дійсно, використання смартфона або іншого мобільного пристрою – більш функціональний і досконалий метод, ніж пошук експоната за номером в путівнику, особливо, коли майже у кожного відвідувача музею є свій власний мобільний пристрій.

Мабуть, найпростіший і найбільш затребуваний варіант використання можливостей сучасних технологій візуалізації – це навігація. Сьогодні багато серйозних музеїв та галерей вже встановили сенсорні термінали і мультимедійні екрани, розмістили в залах QR-коди і маячки-мітки, які працюють по протоколу iBeacon, організували прокат багатозадачних гаджетів або можливість скачування власних мобільних додатків для більш детального знайомства з колекціями.

Так, Лувр пропонує аудіовізуальний гід від розробників Nintendo з 3D-зображеннями, розповіддю про експонати та історії музею, а також інтерактивною картою, яка допоможе не заблукати в лабіринті залів і прокласти найцікавіший маршрут від Венери Мілоської до Мони Лізи. Причому програма налаштовується під індивідуальні смаки і переваги гостей. На 12-метровій сенсорній стіні Gallery One Клівлендського художнього музею можна попередньо ознайомитися з тисячами експонатів колекції, вибравши найбільш цікаві та склавши персональний маршрут. А інтерактивний екран, розміщений в залі петербурзького Російського музею, де демонструється полотно І.Ю. Рєпіна «Засідання Державної ради», дозволяє ближче познайомитися з будь-яким з героїв картини.

Дійсно незабутнє враження справляє знайомство з імерсивними технологіями, які створюють ефект присутності і занурення. Якщо говорити про художні збори, то, як і слід було очікувати, найбільш охоче цей інструмент взяли на озброєння галереї сучасного мистецтва. В кінці 2017 року лондонська Tate Modern представила VR-виставку, присвячену творчості Амедео Модільяні. Її творцям знадобилися місяці кропіткої роботи, щоб відвідувачі змогли перенестися в Париж вікової давнини, відвідати відтворену в віртуальній реальності майстерню художника і навіть «доторкнутися» до предметів, які знаходяться в ній. На початку 2018 московський музей сучасного мистецтва «Гараж» організував виставку «Ставки на гласність», яка розповідала про єдиний в СРСР аукціон «Сотбіс», який відбувся в 1988 році. Крім окремих робіт, проданих на цих торгах, архівних фотографій і відеозаписів, важливою складовою експозиції стала VR-реконструкція, яка дозволила відправитися на 30 років назад і особисто побувати в аукціонному залі під час знаменної події.

Втім, сьогодні засоби віртуальної і доповненої реальності починають застосовувати навіть дуже консервативні художні музеї. Так, на подібний досвід недавно зважилася Третьяковська галерея: тепер за допомогою VR-гарнітури можна відправитися в майстерні Наталії Гончарової та Казимира Малевича, відтворені за фотографіями і архівними описами. Віртуальні простори інтерактивні, тому відвідувачі можуть не просто побачити інтер'єр робочих кімнат, але і пересунути речі на столі або вибрати VR-пензель в майстерні Гончарової або відчути себе співавтором Малевича, «кидаючи» на полотно різні предмети. Для технічних і природничо-наукових зібрань імерсивність стала гарним тоном. Музей природних наук MUSE в італійському Тренто пропонує «політати» над Альпами і «спуститися» по гірськолижній трасі. Панорамна проекція в Датському національному морському музеї в Хельсінгері дозволяє «пірнути» в океанські глибини. У стокгольмському музеї Середземномор'я за допомогою 3D-технологій

пошарово відсканували саркофаг з давньої мумією, а в Музеї природи і науки Perot в Далласі цього літа «оживили» рідкісних динозаврів.

Інші музеї також йдуть в ногу з часом. Наприклад, в Дарвінівському музеї в Москві можна зробити селфі з віртуальними звірами. А в петербурзькому Музеї залізниць Росії, що відкрився в минулому році, технології доповненої реальності демонструють процес заправки паровоза водою і пожвавлюють макет старовинного депо, переносючи глядачів у минуле. До речі, крім розважальної функції, інновації цілком можуть виконувати і соціальну місію, роблячи експонати доступними для людей з обмеженими можливостями. Показовий приклад – весняна виставка «Торкаючись до шедеврів» в Національній галереї в Празі. Тут сліпі і слабозорі люди могли «обмацати» 3D-моделі Венери Мілоської і бюста Нефертіті за допомогою спеціальних рукавичок.

Ще один тренд, пов'язаний з розвитком комп'ютерної думки, – збереження шедеврів мистецтва у вигляді цифрових об'єктів і перенесення їх у віртуальний світ, доступний будь-якій людині. Таким чином ті, хто фізично не може потрапити в той чи інший музей, відправляються в віртуальний тур, щоб у всіх подробицях розглянути на сайті оцифровані експонати колекцій. Наймасштабнішим проектом подібного роду став Google Arts & Culture, який об'єднав понад тисячу музеїв і галерей з усього світу.

Віртуальне життя знаходять не тільки музейні зібрання, а й пам'ятники архітектури, в тому числі втрачені або важкодоступні, а також знакові події людської історії. Наприклад, організація The Arc / k Project запустила проект по створенню цифрової копії сирійської Пальміри, знищеної ІГІЛ. Будівництво VR-версії стародавнього міста може зайняти до двох років. Вчені з Пенсільванії реконструюють інше зруйноване поселення, на цей раз відноситься до цивілізації майя, – місто Кахаль-Печ. А ось гробницю давньоєгипетської цариці Нефертарі можна віртуально відвідати вже сьогодні завдяки проекту Nefertari: Journey to Eternity від Experius VR. Для повного занурення крізь час потрібен шолом HTC Vive. Доступні для

віртуальних мандрівників і археологічні об'єкти додатка MasterWorks: Journey Through History. Спільний продукт компаній CyArk і Oculus детально відтворює чотири історичні пам'ятки трьох континентів: печерне поселення в Колорадо і монументальну різьбу по каменю на горі Рашмор в Південній Дакоті (США), храм стародавньої індіанської цивілізації Чавин-де-Уантар (Перу) і столицю стародавнього Сіаму Аюттхаї (Таїланд). Перспективи тренда по перенесенню реальних об'єктів і подій в цифровий світ практично не обмежені, так як реалізовувати подібні проекти може найширше коло людей. Наочним прикладом служить досвід японських школярів, які створили детальну реконструкцію одного з найтрагічніших подій ХХ століття – скидання атомної бомби на Хіросіму.

Сама захоплююча новація останнього року, яка може перевернути світ мистецтва, – це можливість створювати абсолютно нові цифрові твори і навіть цілі музеї, щоб розміщувати їх реальному світі за допомогою технології доповненої реальності. І люди, прогулюючись, наприклад, по історичному центру Риму або Парижа, зможуть не тільки милуватися красотою архітектури, а й заглядати в віртуальні галереї, просто скачавши додаток або відсканувавши маркер.

Далі буде наведений аналіз існуючих музейних додатків, що на даний момент мають високі оцінки на цифрових торгівельних площадках.

«Музей Ермітаж» – додаток, який дозволяє гуляти по Ермітажу і розглядати його експозиції. Для фанатів Ермітажу і живопису в цілому, використовують Windows 8, є хороша альтернатива – «Галерея Ермітажу». У цьому додатку немає ніяких функцій, крім перегляду картин. Віртуальний візит містить докладні описи кожного приміщення.

На відміну від більшості іноземних музейних програм, «Музей Ермітаж» адаптований під великі екрани планшетів і виглядає на них набагато красивіше. При віртуальному відвідуванні можна дивитися тільки по горизонталі, а між тим більшу частину відвідувачів зачаровують саме чудові стелі Ермітажу. В безкоштовний пакет входить 302 експоната – за все

інше треба платити від 1 до 3 доларів. Втім, відвідування музею теж коштує грошей. Уважні користувачі помітили помилки в описах розташування картин на мапі музею. При переустановленні або установці програми на інший гаджет відновити покупки може бути проблематично. Візуалізація додатку представлена на рисунку 1.1.

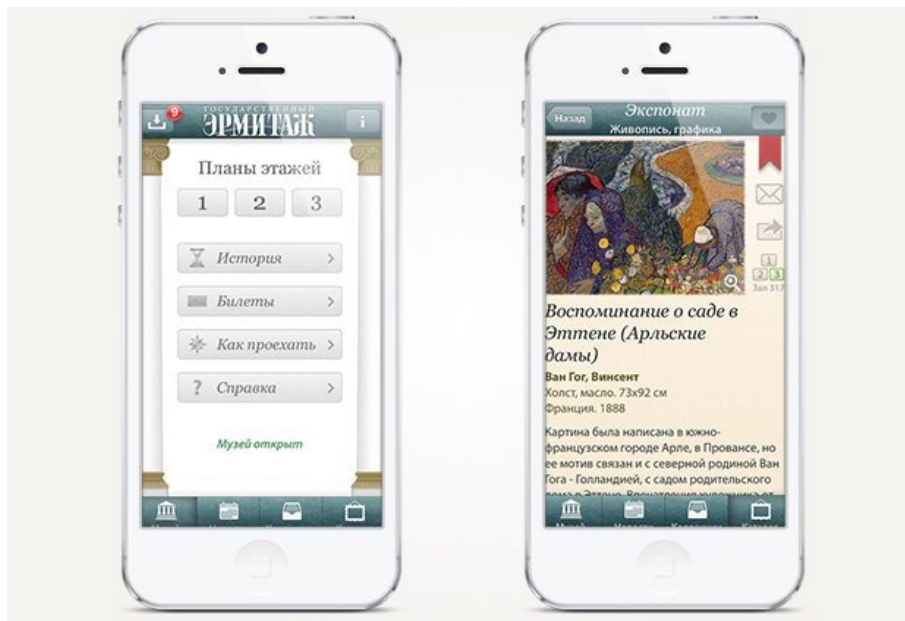


Рисунок 1.1 – Додаток «Музей Ермітаж»

Second Canvas Museo del Prado – додаток для перегляду картин художників іспанської школи живопису, представлених в музеї Прадо. Завдяки цьому додатку колекцію фламандської, німецької, англійської та інших шкіл живопису можна розглядати в своєму телефоні або планшеті. Побачити саму будівлю музею і внутрішній антураж теж можна через додаток. 14 головних шедеврів музею можна розглянути в роздільній здатності 1 гігапксель. Крім власне картин, в додатку очікується поява їх рентгенівських знімків, що дозволяють більш детально вивчити процес створення картин. Аудіодескриптор є тільки у 14 найвидатніших полотен – інформацію про інші 60 полотен доведеться читати очима.

Щоб розглянути у високій роздільній здатності копію «Мони Лізи», представлену в Прадо, доведеться заплатити 1 долар. Візуалізація додатку представлена на рисунку 1.2.

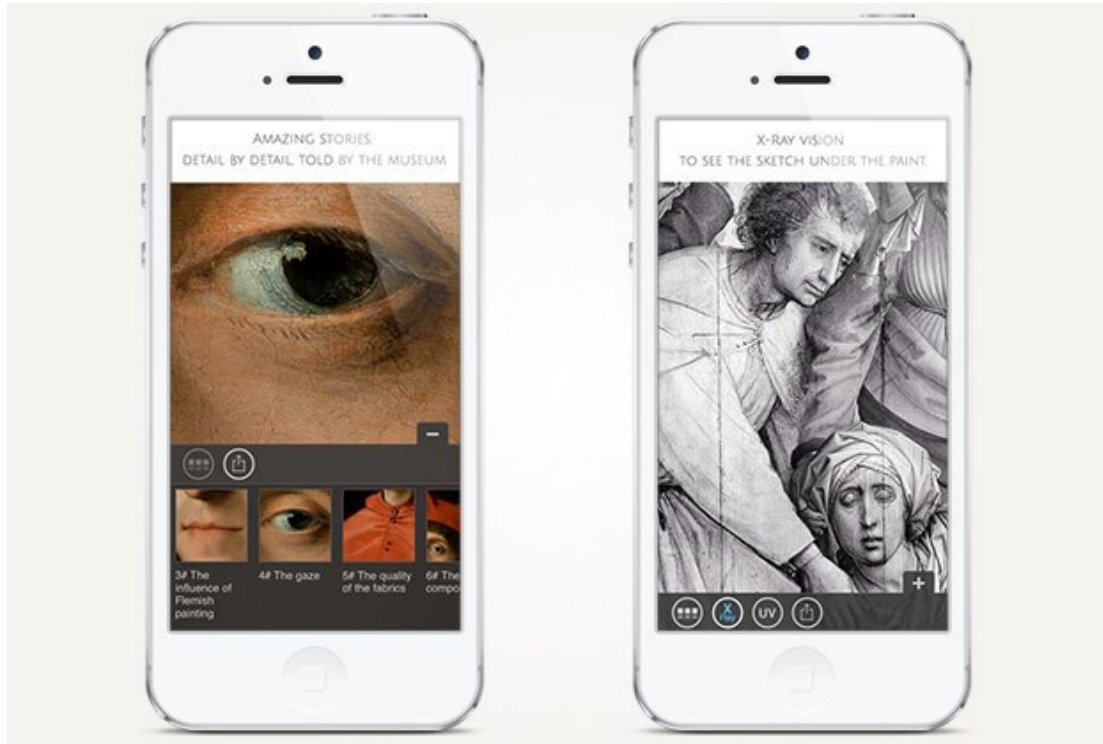


Рисунок 1.2 – Додаток «Second Canvas Museo del Prado»

MoMA – офіційний додаток музею сучасного мистецтва в Нью-Йорку. Фактично всю колекцію музею можна вивчити дистанційно. Додаток можна використовувати як аудіогід. На інтерактивній карті позначені майстерні головних діячів того часу.

У розділі Tours зібрана велика колекція аудіодескрипторів. Самі захоплені користувачі програми знайдуть в розділі More посилання на безліч підкастів і роликів музею. Через завантаження всієї інформації через інтернет додаток працює вкрай повільно. Аудіогід включається тільки в окремому вікні, через що не можна одночасно слухати і розглядати описуваний експонат. Візуалізація додатку представлена на рисунку 1.3.

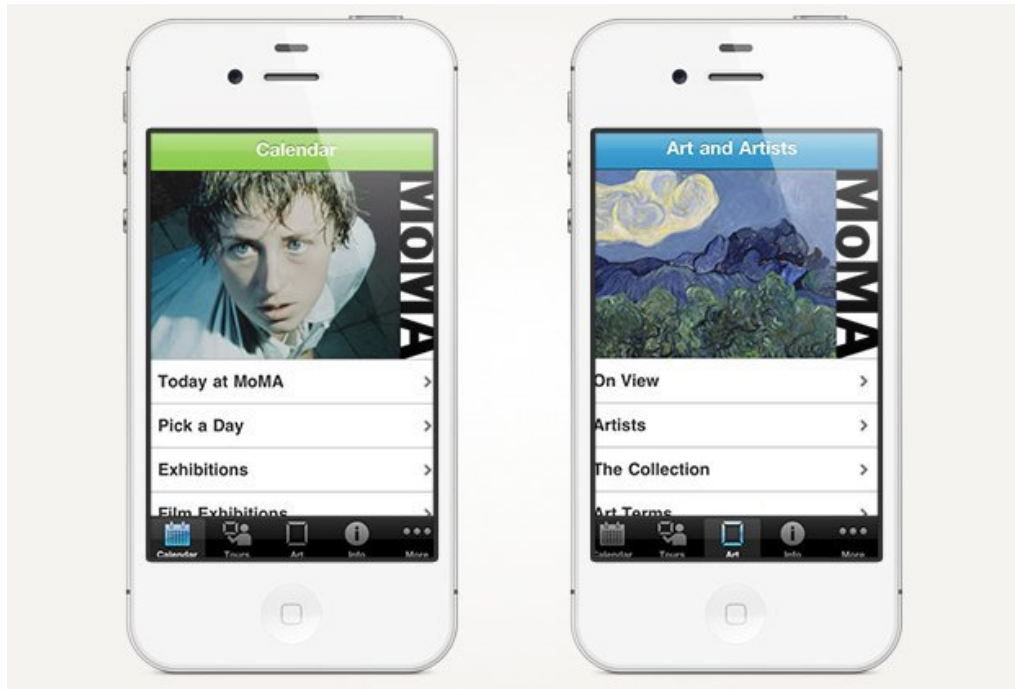


Рисунок 1.3 – Додаток «MoMA»

Guggenheim – додаток максимально близько підводить своїх користувачів до відсутності необхідності відвідувати справжній музей Соломона Гуггенхайма. З усіх нині існуючих додатків Guggenheim найбільш вдало розширює музейне життя людей, які цікавляться сучасним мистецтвом. Виставки можна без проблем скачати з архівів і детально вивчити разом з аудіогідом для кожного експоната. Прямо з карти одним кліком з будь-якого рівню музею можна моментально перенестися на відповідну експозицію. Тут можна подивитися минулі виставки. У додатку є дуже зручні розділи з докладним описом змісту та годин роботи як самого музею, так і магазинів і кафе. При кліці користувача залишають у вбудованому браузері. Після скачування програми з магазину і перед використанням обов'язково потрібно буде завантажити ще 154,5 мегабайта. Багато експонатів надані не в самому високому дозволі. Додаток не оптимізовано для планшетів. Візуалізація додатку представлена на рисунку 1.4.



Рисунок 1.4 – Додаток «Guggenheim»

Musée du Louvre HD – додаток вагою 804 мегабайта, дає можливість вивчення найважливіших експонатів музею, а заодно і розглядати його зали. Втім, тут колекція експонатів неповна. Занадто великий час відгуку при натисканні на кнопки і довге завантаження розділів. Дуже великий розмір додатку. Візуалізація додатку представлена на рисунку 1.5.



Рисунок 1.5 – Додаток «Musée du Louvre HD»

Your Art – цифрова версія Національної галереї мистецтва, розташованої у Вашингтоні, США. У цьому додатку не можна побачити абсолютно всю колекцію музею, але зате можна отримати уявлення про найважливіші експонати. У додатку представлений всього 21 художник. У музеї представлені скульптури, фотографії, графіка і багато іншого, але в додатку можна побачити тільки живопис. Після скачування програми з магазину при його запуску буде необхідно завантажити ще близько 70 мегабайт. Візуалізація додатку представлена на рисунку 1.6.

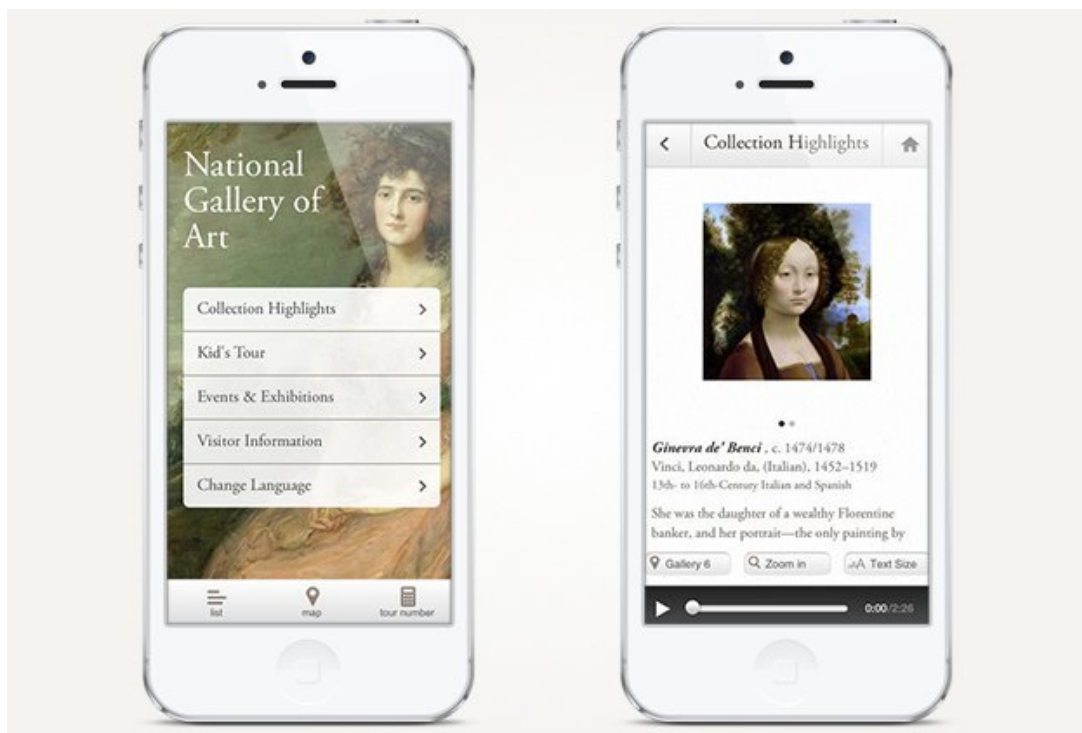


Рисунок 1.6 – Додаток «Your Art»

Love Art: National Gallery, London – даний додаток є збіркою західноєвропейського живопису. При перемиканні між аудіогідами попередній не вимикається автоматично, в результаті чого всі голоси оповідачів накладаються один на одного. Щоб уникнути цього, потрібно вимикати кожен з них вручну. Більшість картин представлено в дуже низькій роздільній здатності. Жодну з них не можна збільшити. Інформації про сам

музей вкрай мало, а при кліці на карту відкривається сторонній браузер. Візуалізація додатку представлена на рисунку 1.7.

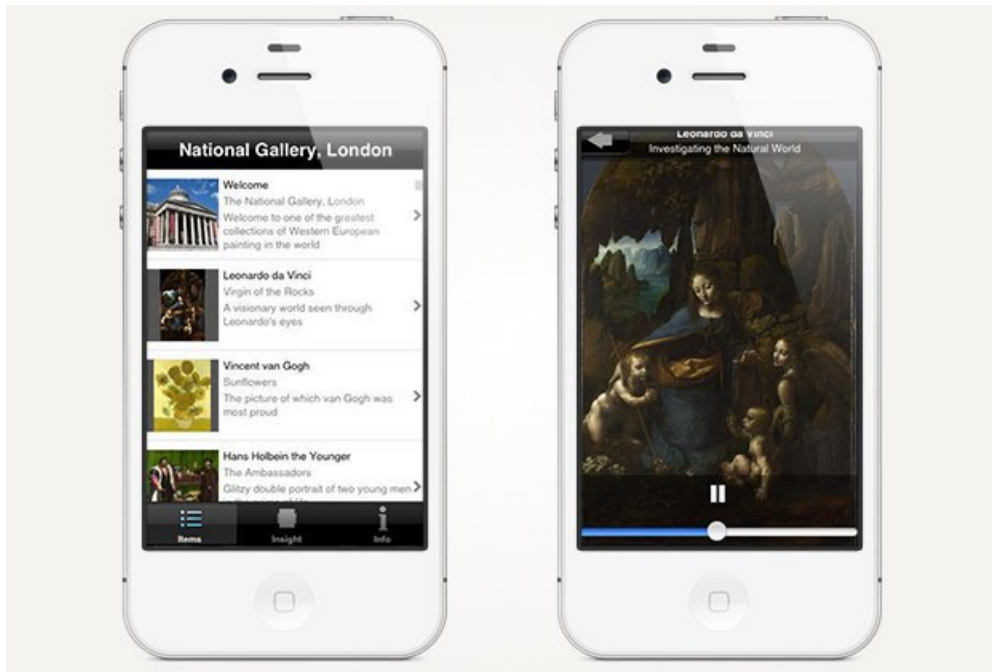


Рисунок 1.7 – Додаток «Love Art: National Gallery, London»

Таким чином можна зробити висновок, що на даний момент торгівельні площадки не можуть надати універсальних додатків до музеїв, що не будуть спеціалізованими до конкретних виставок або митців. Основними проблемами досліджених додатків стали:

- велика «вага» додатку;
- стороння реклама, що заважає перегляду;
- ціна платного додатку;
- невелика кількість контенту;
- неуніверсальність контенту.

1.3 Постановка задачі

Метою даної атестаційної роботи є дослідження технології доповненої реальності та розробка музейного AR-додатку. Згідно з метою у роботі виконані наступні задачі:

- аналіз літератури за темою роботи;
- дослідження методу Feature Detection;
- програмна реалізація технології QCAR;
- розробка додатку для автоматичної ідентифікації зображень в відеопотоці з можливістю подальшої прив'язки до них інтерактивних об'єктів.

Абстрактна візуалізація роботи програми представлена на рисунку 1.8.

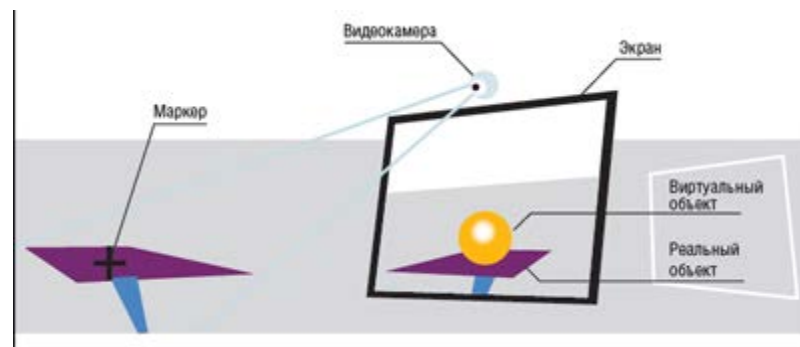


Рисунок 1.8 – Візуалізація роботи «Augmented Reality»

В даному випадку проект розділений на три частини: AR – все, що пов'язано з доповненою реальністю (їм буде головний об'єкт системи, в якому будуть здійснюватися всі розрахунки); Core – графічний рушій для реалізації програми; App – власне сам додаток, що використовує систему доповненої реальності і графічний рушій.

Оскільки додаток призначений для мобільних пристроїв, необхідно буде використовувати графічне ядро, що підтримує нормальний функціонал

Вхідними даними буде відеопотік, який буде надходити через камеру смартфона. Під відеопотоком слід розуміти в даному випадку послідовність кадрів, де інформація від кадру до кадру змінюється не дуже сильно, що дозволяє визначити відповідність між кадрами.

Для здійснення прикладного застосування додаток реалізовуватиме ідентифікацію серії художніх творів. Таким чином необхідно підготувати базу даних зображень, які згодом будуть служити маркерами. У разі ідентифікації в реальному часі на екрані смартфона будуть з'являтися віртуальні об'єкти у вигляді текстового назви картин, при натисканні на які буде виводитися інформація про картину. Приблизний mock-up представлений на рисунку 1.9.

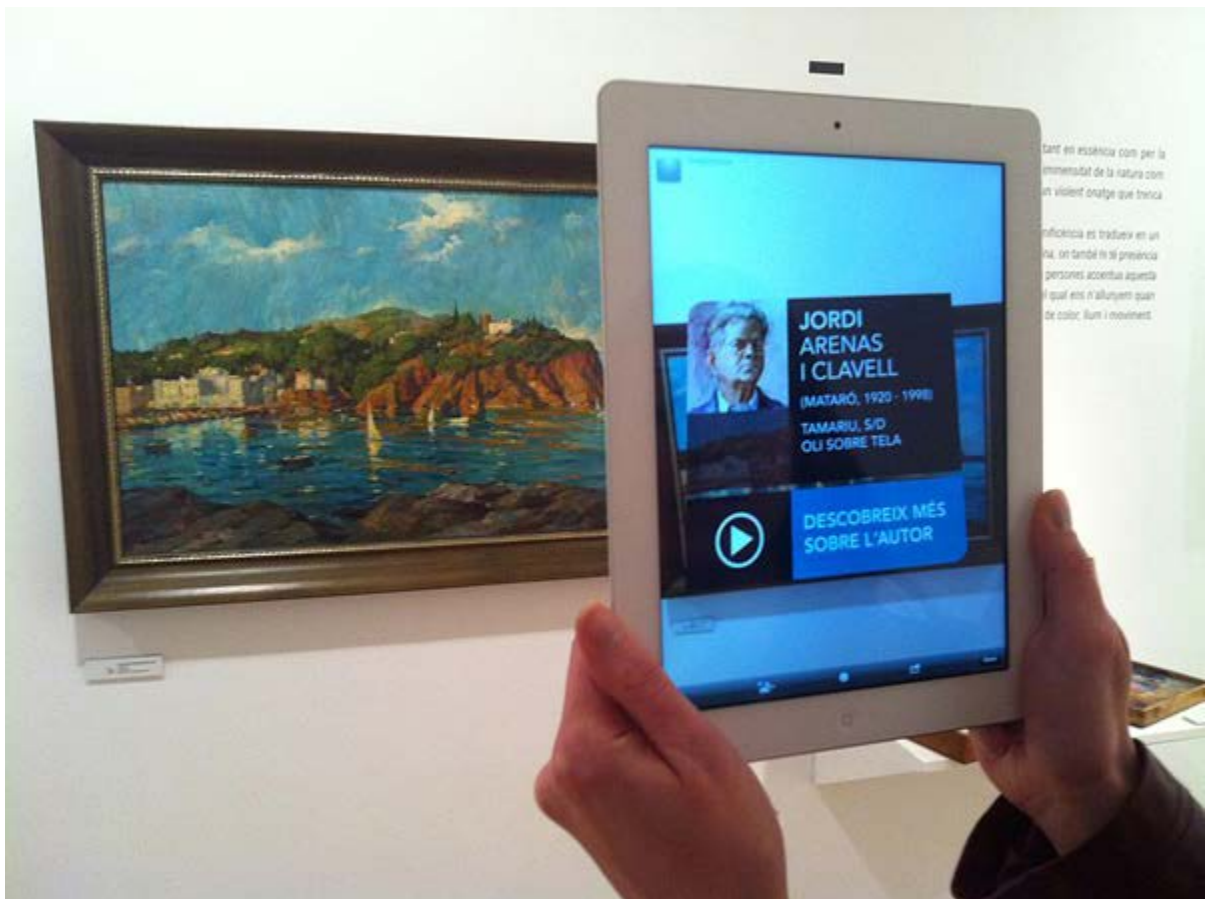


Рисунок 1.9 – Mock-up розроблюємого додатку

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Принципи побудови доповненої реальності

Можна виділити два головних принципи побудови доповненої реальності: на основі маркера і на основі координат місця розташування користувача.

Безмаркерні технології часто застосовуються в мобільних пристроях і будуються за допомогою спеціальних датчиків: акселерометр, гіроскоп, магнетометр, GPS-приймач. В ході даної атестаційної роботи розглянуто побудову доповненої реальності за допомогою маркерів і алгоритмів комп'ютерного зору.

Під маркером розуміється об'єкт, розташований в навколишньому просторі, який визначається та аналізується спеціальним програмним забезпеченням для подальшого відтворення віртуальних об'єктів. На основі інформації про стан маркера в просторі, програма може досить точно спроектувати на нього віртуальний об'єкт, від чого буде досягнуто ефект його фізичної присутності в навколишньому просторі. Використовуючи додаткові графічні фільтри і високоякісні моделі, віртуальний об'єкт може стати практично реальним і важко відмінним від інших елементів інтер'єру або екстер'єру.

Найчастіше в ролі маркера виступає аркуш паперу з деяким спеціальним зображенням. Безліч маркерів досить широко, крім вищевказаного листа маркером можуть бути і геометричні фігури, тривимірні об'єкти, і навіть очі та обличчя людей.

Технології комп'ютерного зору є основоположними для розвитку технологій доповненої реальності і перш за все в галузі використання маркерів. Основний напрямок даної дисципліни – це аналіз і обробка зображень (в тому числі і відеопотоку). Алгоритми комп'ютерного зору дозволяють виділяти ключові особливості на зображенні (кути, межі області),

проводити пошук фігур і об'єктів в реальному часі, виконувати 3D реконструкцію за кількома фотографіями і багато іншого. Приклади об'єктів, які можуть виступати в ролі маркерів представлені на рисунку 2.1.

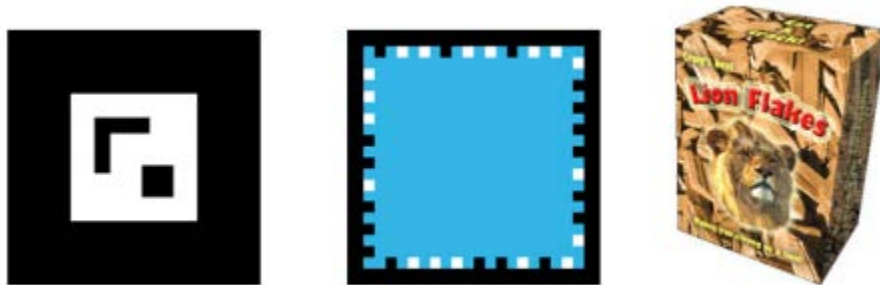


Рисунок 2.1 – Приклади маркерів

В області доповненої реальності алгоритми комп'ютерного зору використовуються для пошуку в відеопотоці спеціальних маркерів. Залежно від завдання в якості маркера можуть виступати як спеціально сформовані зображення, так і особи людей. Після знаходження маркера в відеопотоці і обчислення його розташування, з'являється можливість побудови матриці проєкції і позиціонування віртуальних моделей. За допомогою них можна накласти віртуальний об'єкт таким чином, що буде досягнутий ефект присутності. Основна складність полягає в тому, щоб знайти маркер, визначити його місце розташування в кадрі і спроектувати відповідним чином віртуальну модель.

За останнє десятиліття була створена велика теоретична база в сфері обробки зображень та пошуку в ньому різних об'єктів [5]. Перш за все це стосується методів *template matching*, генетичних алгоритмів і *feature detection*. З точки зору побудови доповненої реальності часто використовується *feature detection*.

Концепція *feature detection* в комп'ютерному зорі належить до методів, які націлені на обчислення абстракції зображення і виділення на ньому

ключових особливостей. Дані особливості можуть бути як у вигляді ізольованих точок, так і кривих або пов'язаних областей. Не існує строгого визначення того, що таке ключова особливість зображення. Кожен алгоритм розуміє під цим своє (кути, межі, області тощо).

Найчастіше для пошуку маркерів виконується пошук і порівняння зображень по ключових точках. Ключова точка – це певна ділянка картинки, яка є характерною для заданого зображення. Як виглядають ключові точки на зображенні вказано на рисунку 2.2.

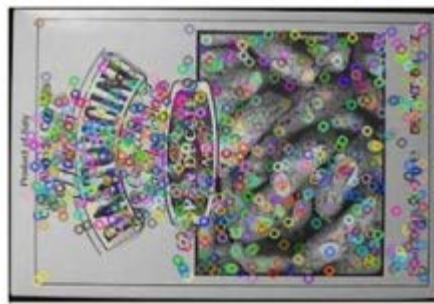


Рисунок 2.2 – Приклад ключових точок на зображенні

Для знаходження ключових точок для ідентифікації зображення в відеопотоці використовуються три складові:

- детектор – здійснює пошук ключових точок на зображенні, безпосередньо feature detector;
- дескриптор – виробляє опис знайдених ключових точок, оцінюючи їх позиції через опис навколишніх областей;
- матчер – здійснює побудову відповідностей між двома наборами точок.

Спочатку за допомогою детектора проводиться пошук ключових точок шаблонного (шуканого) зображення. Отримані точки потім описуються за допомогою дескриптора. Дана інформація зберігається в окремий файл (або базу даних), щоб не виконувати цей процес повторно. При обробці відеопотоку з метою пошуку заданого шаблону описаний процес

виконується для кожного кадру (за винятком збереження даних). Для встановлення відповідності між ключовими точками і дескрипторами застосовується матчер. Приклад відповідності точок надано на рисунку 2.3.

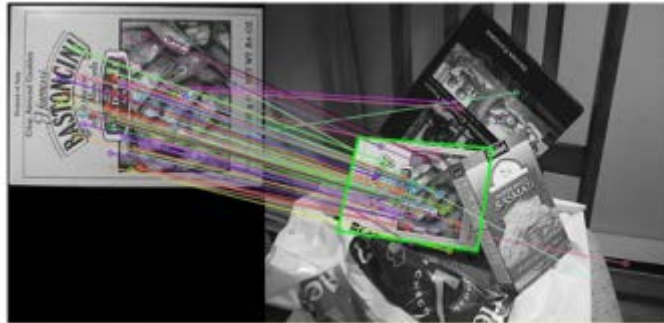


Рисунок 2.3 – Відповідності між точками шаблону та тестового зображення

Природно припустити, що різні алгоритми працюють з різною швидкістю та ефективністю. В умовах їх застосування для побудови доповненої реальності необхідно використовувати тільки ті, які показують високу швидкість роботи при досить хорошій якості відстеження позицій ключових точок. В іншому випадку ми можемо отримати помітні відставання у відеоданих, що знімаються. Для підвищення швидкості роботи алгоритмів роботи feature points detection застосовуються різні способи фільтрації точок, щоб мінімізувати їх число і відсіяти зовсім погані поєднання. Таким чином можна домогтися не тільки підвищення швидкості роботи, але і якості трекінгу маркерів.

2.2 Типи систем доповненої реальності

Людина отримує уявлення про навколишній простір за допомогою великого набору органів сприйняття навколишньої інформації. Система доповненої реальності, будучи посередником між людиною і реальністю, повинна створювати сигнал для одного з таких органів.

Візуальні. В їх основі лежить зорове сприйняття людини. Завдання таких систем – створити зображення, яке буде використано людиною. Оскільки зображення для людини є більш інформативним і зрозумілим, такий вид систем є більш поширеним.

Аудіо. Такі системи орієнтовані на слухове сприйняття [6]. Найчастіше такі системи використовуються в навігації. Наприклад, вони видають спеціальні сигнали, коли людина досягає певного місця. Можливе використання стереоскопічного ефекту, що дозволяє людині йти в потрібному напрямку, орієнтуючись на джерело звуку. Прикладом такої системи є Hear & There [4].

Аудіовізуальні. Це комбінація двох попередніх типів, однак, аудіоінформація в них має лише допоміжний характер. Системи доповненої реальності завжди потребують інформації, одержуваної з навколишнього середовища. Саме на основі цих даних будуються віртуальні об'єкти. Кожна з таких систем має певний набір сенсорів – пристроїв, що дозволяють збирати інформацію з навколишнього середовища: звукові і електромагнітні коливання, прискорення тощо. Для класифікації має сенс розділяти сенсори не за типами реєстрованих фізичних величин, а за їх призначенням, оскільки подібні за своєю природою сигнали можуть нести різну інформацію. За типом сенсорів можна виділити наступні системи:

Геопозиційні. Орієнтуються, перш за все, на сигнали системи позиціонування GPS. На додаток до приймачів таких сигналів геопозиційні системи можуть використовувати компас і акселерометр для визначення кута повороту відносно вертикалі і азимута.

Оптичні. Такі системи обробляють зображення, отримане з камери, яка переміщається разом з системою або незалежно від неї.

Системи доповненої реальності можна розрізнити за ступенем взаємодії з користувачем. У деяких системах користувач грає пасивну роль, він лише спостерігає за реакцією системи на зміни в навколишньому середовищі. Інші ж системи вимагають активного втручання користувача –

він може управляти як роботою самої системи, для досягнення результатів, так і змінювати віртуальні об'єкти.

Автономні. Вони не вимагають втручання користувача. Завдання таких систем зводиться до надання інформації про об'єкти. Наприклад, подібні системи можуть аналізувати об'єкти, що знаходяться в полі зору людини і видавати довідкову інформацію про них. Також системи такого типу використовуються в медицині. Наприклад, система Gait Aid [7] для людей з порушеннями опорно-рухового апарату. Вона шляхом використання віртуальних об'єктів надає мозку додаткову інформацію, яка допомагає координувати рухи [8][9].

Інтерактивні. Такі системи засновані на взаємодії з користувачем. На різні дії користувач отримує різну відповідь. У подібних системах необхідно мати пристрій введення інформації. В якості такого пристрою може застосовуватися сенсорний екран мобільного телефону, планшет або спеціальний маніпулятор. Вибір пристроїв введення залежить від специфіки системи. У разі простих дій з віртуальним об'єктом, достатньо простого вказівного пристрою. Якщо ж необхідна імітація будь-яких реальних процесів і виконання складних маніпуляцій з об'єктами використовуються спеціальні маніпулятори, які мають різну кількість ступенів свободи. Прикладом можуть служити пристрої PHANTOM [6].

Інтерактивність виражається в різному ступені. Бувають системи, що дозволяють користувачеві активно змінювати віртуальне середовище [7]. Зазвичай це системи-симулятори будь-яких реальних дій. Вони використовуються у разі, коли використання реальних об'єктів неможливо, наприклад, спеціалізовані медичні тренажери, що дозволяють початківцям лікарям відпрацьовувати необхідні навички [8].

Існують інші системи, де користувачеві не потрібно змінювати віртуальне середовище. Замість цього користувач обирає, які віртуальні об'єкти він хоче побачити. Користувач також має можливість маніпулювати віртуальними об'єктами, але не на рівні структури, а на рівні відображення,

тобто застосовувати, наприклад афінні перетворення типу повороту, переміщення тощо. До даної групи можна віднести різні архітектурні системи, що дозволяють побачити, як впишеться в реально існуючу обстановку нова споруда або його частина, а також навігаційні та геоінформаційні системи. Подібні системи можуть показувати частини об'єктів інтересу, приховані іншими будівлями, додаткову інформацію про обрані об'єктах тощо. За ступенем мобільності системи доповненої реальності можна класифікувати як:

Стационарні. Системи цього типу призначені для роботи в фіксованому місці; переміщення таких систем означає часткове або повне припинення їх працездатності.

Мобільні. Системи цього типу можуть без зусиль переміщатися; часто таке переміщення і лежить в основі виконуваної ними функції [8].

Належність до того чи іншого типу визначається функціями системи. Так, симулятор хірургічного столу не повинен бути мобільним, оскільки його завдання – відтворити для людини спеціальні умови, максимально наближених до реальних. У той же час навігаційна система повинна бути якомога більш мобільною, щоб вона могла переміщатися разом з транспортним засобом або людиною, не створюючи додаткових витрат на її переміщення.

2.3 Технологія QCAR

Для роботи доповненої реальності використовуються маркери, які допомагають визначати положення камери в просторі. Це обмежує її використання, так як по-перше, маркери повинні бути постійно в кадрі, а по-друге, їх необхідно спочатку підготувати.

Працювати відразу з усіма точками зображення досить складно, але для мобільних платформ потрібно спрощувати. Тому просто будемо виділяти

окремі особливі точки на зображенні і стежити за їх переміщеннями. Для початку буде необхідно пройти процес препроцесінга маркера.

Метод QCAR вирішує два завдання – пошук особливих точок зображення і створення їх дескрипторів, інваріантних до масштабу і обертання. Це означає, що опис ключової точки буде однаково, навіть якщо зразок змінить розмір і буде повернений. Крім того, сам пошук ключових точок повинен бути інваріантним. Так, щоб повернений об'єкт сцени мав той же набір ключових точок, що і зразок.

В алгоритмі розглядається коло з 16 пікселів навколо точки кандидата P , намальовані алгоритмом Брезенхема. За методом побудови він схожий на малювання лінії. У цьому алгоритмі будується дуга кола для першого квадранта, а координати точок кола для інших квадрантів виходять симетрично. На кожному кроці алгоритму розглядаються три пікселя, і з них вибирається найбільш придатний шляхом порівняння відстаней від центру до обраного пікселя з радіусом кола. Далі вибирається точка зображення p , для якої буде вирішуватися, чи є вона особливою чи ні. Нехай I_p – яскравість точки. Вибирається значення порога t . Розглядається коло з 16 пікселів навколо обраної точки, даний крок зображено на рисунку 2.4.

Точка p вважається кутом, якщо серед 16-ти пікселів кола існує n пікселів, кожен з яких яскравіше, ніж $I_p + t$, або n пікселів, кожен з яких темніше, ніж $I_p - t$ (такі пікселі відзначені пунктирною лінією на рисунку 2.4) n зазвичай вибирається рівним 12. Експерименти показали, що найменше значення n , при якому точки починають стабільно виявлятися, дорівнює 9.

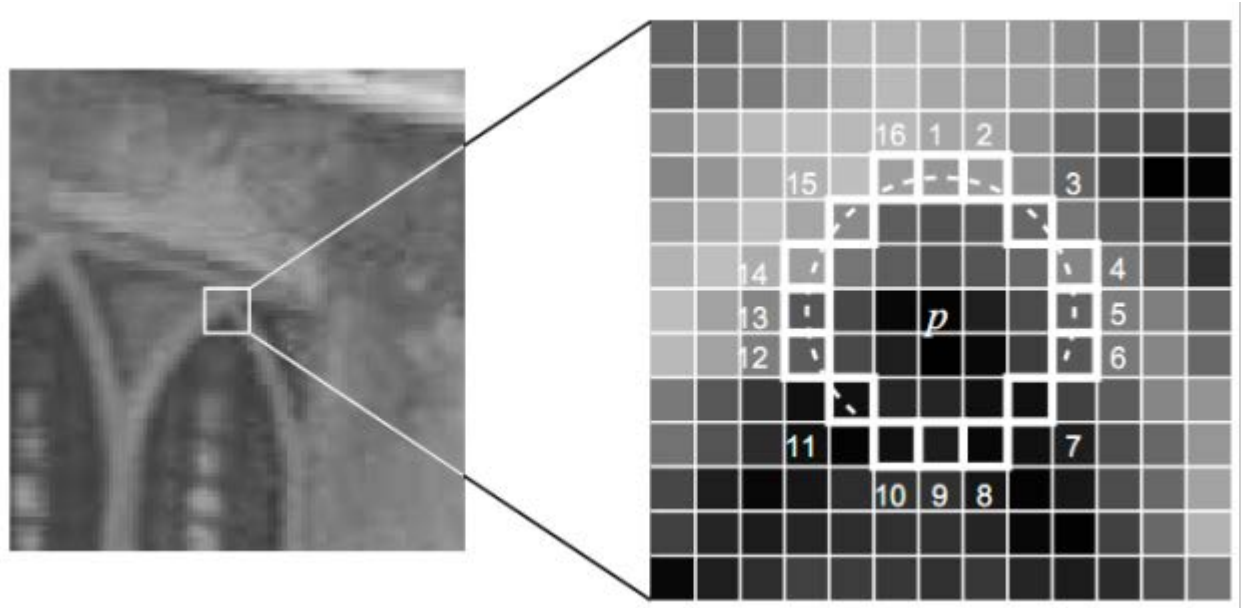


Рисунок 2.4 – Робочий окіл пікселя при використанні FAST-детектора

Висока швидкість роботи даного алгоритму обумовлена тим, що спочатку перевіряється інтенсивність лише чотирьох точок з кола, під номерами 1,5, 9 і 13. Якщо хоча б для трьох з них виконується умова нерівності, яка виглядає наступним чином: $I_{p_i} > I_p + t$ або $I_{p_i} < I_p - t$, $i = 1..4$, тоді проводиться перевірка решти 12-ти пікселів. В іншому випадку вибирається наступна точка і алгоритм повторюється для неї.

Однак даний алгоритм має ряд недоліків, наприклад його ефективність залежить від порядку обробки зображення і розподілу пікселів, поблизу деякого околу може виявитися кілька особливих точок. В [8] було запропоновано використання машинного навчання для виправлення цих недоліків, на основі даних досліджень був виведений алгоритм технології QCAR:

- вибирається набір зображень для навчання;
- алгоритм FAST застосовується для пошуку особливих точок на кожному із зображень;

- для кожної особливої точки зберігається 16 пікселів навколо неї у вигляді вектора. Після виконання даного кроку для кожного зображення всі отримані вектори об'єднуються в один вектор P ;

- кожен піксель з 16-ти може знаходитися в одному з трьох станів, які вказані на формулі 2.1:

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \text{ (темніше)} \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t \text{ (схожий)} \\ b, & I_p + t \leq I_{p \rightarrow x} \text{ (світліше)} \end{cases} \quad (2.1)$$

- в залежності від станів, наведених вище, P розділяється на три підмножини P_d, P_s, P_b ;

- в розгляд вводиться булева змінна K_p , яка дорівнює 1, якщо p – кут та 0 у зворотному випадку;

- використовується алгоритм ID3 (класифікатор дерев рішень) [9]. Він вибирає x , який дає більше інформації про те, чи є піксель кутом чи ні, обчисленої за допомогою ентропії K_p ;

- ця процедура повторюється рекурсивно, до тих пір, поки ентропія K_p не дорівнюватиме нулю;

- створене таким чином дерево рішень використовується для більш швидкого детектування в інших зображеннях.

Для того, щоб уникнути знаходження декількох особливих точок поблизу деякого околу, використовується підхід, званий non-maximal suppression, тобто пошук локальних максимумів. Обчислюється значення функції V для кожної зі знайдених особливих точок. V – це сума різниць інтенсивностей p і кожного з 16-ти навколишніх пікселів. Розглядаються дві сусідніх особливих точки і порівнюються їх значення V . Точка з меншим значенням V виключається з розгляду.

2.4. Алгоритм трекінга

Графічна інтерпретація методу визначення координат показаний на рисунку 2.5. Ідея полягає в тому, що при зміні кута з якого дивиться камера, змінюється розмір проекції, як зазначено на рисунку 2.6.

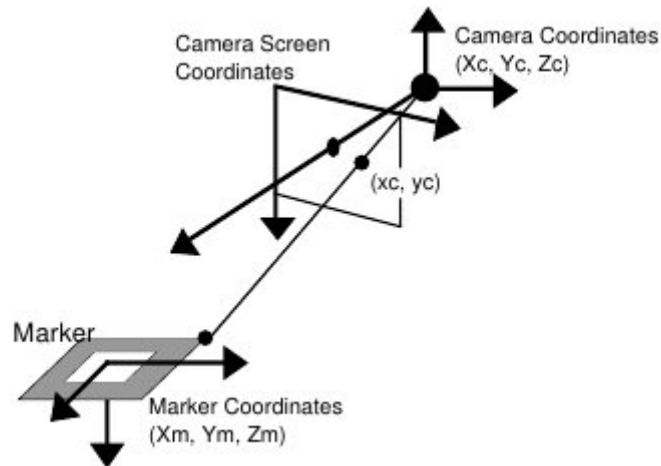


Рисунок 2.5 – Принцип визначення координат

Маючи координати особливих точок, необхідно визначити положення камери в просторі. Для цього спочатку виконується ініціалізація по двом сусіднім кадрах, в цей момент відбувається обчислення координат камери в цих кадрах і положення особливих точок. Далі, по відомим тривимірним координатам точок можна вже обчислити положення камери в кожному наступному кадрі відеопотоку.

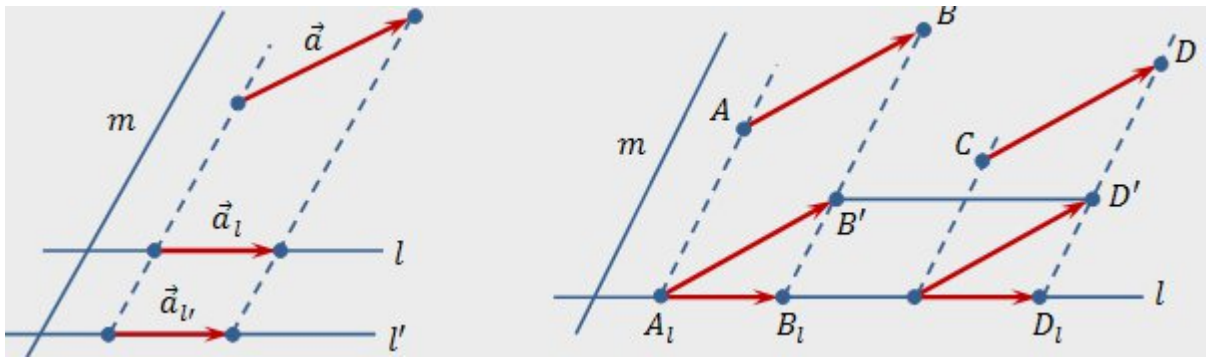


Рисунок 2.6 – Зміна розміру проекції при зміні кута камери

Для більш стабільної роботи додаються нові точки в процес спостереження, складаючи карту простору, який бачить камера. Для початку переведемо тривимірні координати особливих точок в екранні. Для цього використовується формула 2.2.

$$(\text{screen}[i]_x * \text{screen}[i]_z, \text{screen}[i]_y * \text{screen}[i]_z, \text{screen}[i]_z) = mProj * \\ * mWorld * (\text{world}[i], 1). \quad (2.2)$$

World [i] – це особливі точки в світових координатах. До ініціалізації ці координати не відомі. Screen [i] – компоненти x та y – це координати особливої точки на зображенні, z – глибина щодо камери. Всі ці координати вже будуть своїми на кожному кадрі відеопотоку, mProj – матриця проекції

розміром 3 на 3 та має вигляд $\begin{bmatrix} pf_x & 0 & -pc_x \\ 0 & pf_y & -pc_y \\ 0 & 0 & 1 \end{bmatrix}$, тут pf – фокальна відстань

камери в пікселях, pc – оптичний центр камери в пікселях (центр зображення). mWorld – матриця, що описує трансформацію точок, розміром 3 на 4 (світова матриця з якої прибрали останній рядок (0 0 0 1)). У цій матриці міститься інформація про переміщення і повороти камери. І це те, що власне ми в першу чергу шукаємо на кожному кадрі. Спростимо формулу, позбувшись від матриці mProj, шляхом введення s [i], зазначеного у формулі 2.3. Результат показаний на формулі 2.4.

$$c[i] = mProj^{-1} * (screen[i]_x, screen[i]_y, 1) \quad (2.3)$$

$$c[i] * screen[i]_z = mWorld * (world[i], 1) \quad (2.4)$$

Ініціалізація відбувається за двома кадрами. Позначимо їх як А і В. Таким чином з'являються матриці $mWorldA$ – і $mWorldB$. Точки $c[i]$ на кадрах А і В позначимо як $ca[i]$ і $cb[i]$. Матриця $mWorldA$ – одинична матриця. Матриця $mWorldB$ обчислюється за допомогою точок, розташованих на одній площині. Для них буде справедлива формула 2.5.

$$\begin{pmatrix} cb_x * screenB_z \\ cb_y * screenB_z \\ screenB_z \end{pmatrix} = \begin{pmatrix} H_{00} * ca_x + H_{01} * ca_y + H_{02} \\ H_{10} * ca_x + H_{11} * ca_y + H_{12} \\ H_{20} * ca_x + H_{21} * ca_y + H_{22} \end{pmatrix} \quad (2.5)$$

Далі слід позбутися $screenB_z$ і перейти до вигляду, представленою на формулі 2.6. Представивши матрицю Н у вигляді вектора $H' = (H_{00}, H_{01}, \dots, H_{22})$ і представити ці рівняння в матричному вигляді, зазначеному на формулі 2.7.

$$\begin{aligned} H_{20} * ca_x * cb_x + H_{21} * ca_y * cb_x + H_{22} * cb_x \\ = H_{00} * ca_y + H_{01} * ca_y + H_{02} \\ H_{20} * ca_x * cb_y + H_{21} * ca_y * cb_y + H_{22} * cb_y \\ = H_{10} * ca_x + H_{11} * ca_y + H_{12} \end{aligned} \quad (2.6)$$

$$\begin{bmatrix} ca_x, ca_y, 1, 0, 0, 0, & -ca_x * cb_x, -ca_y * cb_x, -cb_x \\ 0, 0, 0, ca_x, ca_y, 1 & -ca_x * cb_y, -ca_y * cb_y, -cb_y \end{bmatrix} * H' = 0. \quad (2.7)$$

Нова матриця позначається М. Виходячи з цього – $M * H' = 0$. Дані операції розписувалися тільки для однієї точки, тому в матриці М тільки 2 рядки. Для того, щоб знайти матрицю Н', необхідно, щоб в матриці М було кількість рядків більше або дорівнює кількості стовпців. Якщо маємо тільки чотири точки, то можна просто додати ще один рядок з нулів, вийде матриця

розміром 9 на 9. Вектор H' - це векторне подання матриці H . Все це справедливо тільки для точок, розташованих на одній площині. А які з них розташовані, а які ні, заздалегідь невідомо, але можна припустити за допомогою методу Ransac. У циклі, з заздалегідь заданою кількістю ітерацій виконуються наступні дії:

- випадковим чином вибираються чотири пари точок кадрів A і B ;
- знаходиться матриця H ;
- рахується, скільки точок дають помилку менше заданого значення;
- вибирається H на тій ітерації, в якій вийшло більше точок.

З матриці H необхідно отримати матрицю переходу положення кадру A до кадру B – матриця $mMotion$. Виконується сингулярне розкладання матриці H . Дані перетворення представлені на формулах 2.8 – 2.11

$$s = [U] * [V] \quad (2.8)$$

$$w1 = |W_{00}|; w2 = |W_{11}|; w3 = |W_{22}| \quad (2.9)$$

$$x1 = \sqrt{(w1^2 - w2^2)/(w1^2 - w3^2)} \quad (2.10)$$

$$x2 = 0; x3 = \sqrt{(w2^2 - w3^2)/(w1^2 - w3^2)} \quad (2.11)$$

Вектори, представлені у вигляді масивів, що вказують на потрібний знак представлені на формулі 2.12.

$$\begin{aligned} e1 &= [1, -1, 1, -1, 1, -1, 1, -1] \\ e2 &= [1, 1, -1, -1, 1, 1, -1, -1] \\ e3 &= [1, 1, 1, 1, -1, -1, -1, -1] \end{aligned} \quad (2.12)$$

Далі використовуються $R [i]$ – матриця повороту, $t [i]$ – вектор зміщення, i , використовуючи попередні обчислення отримуємо 8 можливих варіантів $mMotion$. Алгоритм роботи представлений на формулах 2.13 – 2.17.

$$\sin A = (w1 - w3 * e3[i]) * x1 * x3 * e1[i] * \frac{e2[i]}{w2} \quad (2.13)$$

$$\cos A = (w3 * x1^2 + w1 * x3^2 * e3[i]) * \frac{1}{w2} \quad (2.14)$$

$$R[i] = U * \begin{bmatrix} \cos A & 0 & -\sin A * e3[i] \\ 0 & 1 & 0 \\ \sin A & 0 & \cos A * e3[i] \end{bmatrix} * V^T * e3[i] \quad (2.15)$$

$$t[i] = U * ((w1 - w3 * e3[i]) * x1 * e1[i]), 0, (w3 - w1 * e3[i]) * x3 * e2[i]) \quad (2.16)$$

$$mMotion[i] = \begin{bmatrix} R[i]_{00} & R[i]_{01} & R[i]_{02} & t[i]_x \\ R[i]_{10} & R[i]_{11} & R[i]_{12} & t[i]_y \\ R[i]_{20} & R[i]_{21} & R[i]_{22} & t[i]_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

Далі виходячи зі співвідношення, зазначеного у формулі 2.18, можна зробити висновок про те, що $mWorld_B = mMotion [i]$. Залишається вибрати одну матрицю з 8-ми $mMotion [i]$. Якщо випустити промені з точок першого і другого кадру, то вони повинні перетинатися, причому перед камерою, як в першому, так і в другому випадку. Тому слід підрахувати кількість точок перетину перед камерою в першому і в другому кадрі, використовуючи отримані $mMotion [i]$, відкидаються варіанти, при яких кількість точок буде меншим.

$$mWorld_B = mMotion[i] * mWorld_A \quad (2.18)$$

Залишивши пару матриць в результаті, вибирається та, яка дає менше помилок. Таким чином використовуючи отримані матриці $mWorldA$ і $mWorldB$ можна знайти світові координати точок по проєкціях.

Для обчислення світових координат точки за кількома проєкціями необхідно знайти точку $world$, яку позначимо як a . Припустимо у нас є кадри, в яких відомі матриці $mWorld$ (позначаються як $mW [0], mW [1], \dots$) і відомі координати проєкцій точки a ($z [0], z [1], \dots$).

Система рівнянь, отримана таким чином представлена у формулі 2.19. Однак можна уявити їх у вигляді формули 2.20, позбувшись $screen[i]_z$.

$$\begin{cases} c[0] * screen[0]_z = mW[0] * a; \\ c[1] * screen[1]_z = mW[1] * a; \\ \dots \end{cases} \quad (2.19)$$

T

$$= \begin{bmatrix} mW[0]_{00} + c[0]_x * mW[0]_{20} & mW[0]_{01} + c[0]_x * mW[0]_{21} & \dots \\ mW[0]_{10} + c[0]_y * mW[0]_{20} & mW[0]_{11} + c[0]_y * mW[0]_{21} & \dots \\ \dots & \dots & \dots \end{bmatrix} \quad (2.20)$$

$$\begin{bmatrix} mW[0]_{02} + c[0]_x * mW[0]_{22} & mW[0]_{03} + c[0]_x * mW[0]_{23} \\ mW[0]_{12} + c[0]_y * mW[0]_{22} & mW[0]_{13} + c[0]_y * mW[0]_{23} \\ \dots & \dots \end{bmatrix} f = (a_x * s, a_y * s, a_z * s, s) \quad (2.21)$$

$$T * f = 0 \quad (2.22)$$

У формулі 2.21 видно, що була введена s – будь-яке число, не рівне нулю, формула 2.22 – система рівнянь в матричному вигляді. T – відомо, f – необхідно обчислити, щоб знайти a . Вирішивши дане рівняння за допомогою сингулярного розкладання, отримується вектор f . І відповідно точка a , яка описана формулою 2.23.

$$a = \left(\frac{f_x}{f_w}, \frac{f_y}{f_w}, \frac{f_z}{f_w} \right) \quad (2.23)$$

Після проведених дій можна обчислити положення камери використовуючи відомі світові координати точок. Використовується ітеративний алгоритм. Початкове наближення – попередній результат визначення. На кожній ітерації. Вводимо точки $b[i]$, зазначені у формулі 2.24, в ідеалі точки $c[i]$ мають бути рівними точкам $b[i]$, але так як поточний наближення $mWorld$ – це лише поки наближення, вони будуть відрізнятися. Підраховується вектор помилки за формулою 2.25 і вирішуємо систему рівнянь методом найменших квадратів за формулою 2.26.

$$b[i] = mWorld[i] * (world[i], 1) \quad (2.24)$$

$$e[i] = \left(c[i]_x - \frac{b[i]_x}{b[i]_z}, c[i]_y - \frac{b[i]_y}{b[i]_z} \right) \quad (2.25)$$

$$\begin{cases} \mu_0 * \left(-\frac{1}{b[0]_z} \right) + \mu_2 * \frac{b[0]_x}{(b[0]_z)^2} + \mu_3 * \frac{b[0]_x * b[0]_y}{(b[0]_z)^2} - \mu_4 * \left(1 + \left(\frac{b[0]_x}{b[0]_z} \right)^2 \right) + \\ \mu_1 * \left(-\frac{1}{b[0]_z} \right) + \mu_2 * \frac{b[0]_y}{(b[0]_z)^2} + \mu_3 * \left(1 + \frac{b[0]_y^2}{(b[0]_z)^2} \right) - \mu_4 * \frac{b[0]_x * b[0]_y}{(b[0]_z)^2} - \\ + \mu_5 * \frac{b[0]_y}{b[0]_z} = e[0]_x \\ - \mu_5 * \frac{b[0]_x}{b[0]_z} = e[0]_y \end{cases} \quad (2.26)$$

В результаті проведених операцій необхідно знайти шестивимірний вектор μ – вектор експоненти. Далі, слідує ітерації, знаходиться матриця зсувів з вектора μ і оновлюємо матрицю виходячи з формули 2.27.

$$mWorld = dT * mWorld \quad (2.27)$$

Теоретичні відомості вказують, що достатньо 10-15 ітерацій. Проте можна вказати додаткову умову, яке виводить з циклу, якщо значення $mWorld$ вже досить близько до згаданого значення.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Для експериментального дослідження було потрібно вибрати відповідне специфічне, виходячи з предметної області, програмне забезпечення. Таким чином, в якості модуля реалізації, який підтримує технології Augmented reality для забезпечення функції трекінгу був обраний модуль Vuforia. Оскільки Vuforia підтримує різні 2D і 3D типи мішеней, включаючи маркерні Image Target, що дозволить в подальшому використовувати вкладені функції для реалізації плавного треку маркерів, які будуть реалізовані у вигляді деяких реальних графічних зображень. Наступним кроком став вибір графічного рушія, який дозволив би перенести додаток на мобільну платформу, оскільки відеопотік буде транслюватися безпосередньо через відеокамеру мобільного пристрою. Оптимальним ядром став інструмент для розробки додатків, що працює під операційною системою Windows дозволяє компілювати проекти під мобільні платформи, а, зокрема, Android. Для коректної роботи зі структурою Android-платформи так само необхідно було використовувати Android Studio SDK. Також необхідно було вибрати безпосередньо мову програмування. Мова програмування повинна мати досить широку алгоритмічну базу, можливість модулювання, а також широку підтримку платформ. Інструментом розробки стала мова C # 7.0, оскільки він задовольняє всім перерахованим вимогам і легко інтегрується в графічне ядро Unity.

Для початку необхідно було реалізувати модуль алгоритму FAST технології QCAR. Реалізація на мові C # представлена на рисунку 3.1.

```

public static Image<Bgr, Byte> Draw(Image<Gray, Byte> modelImage, Image<Gray, byte> observedImage)
{
    HomographyMatrix homography = null;

    FastDetector fastCPU = new FastDetector(10, true);
    VectorOfKeyPoint modelKeyPoints;
    VectorOfKeyPoint observedKeyPoints;
    Matrix<int> indices;

    BriefDescriptorExtractor descriptor = new BriefDescriptorExtractor();

    Matrix<byte> mask;
    int k = 2;
    double uniquenessThreshold = 0.8;

    //extract features from the object image
    modelKeyPoints = fastCPU.DetectKeyPointsRaw(modelImage, null);
    Matrix<Byte> modelDescriptors = descriptor.ComputeDescriptorsRaw(modelImage, null, modelKeyPoints);

    // extract features from the observed image
    observedKeyPoints = fastCPU.DetectKeyPointsRaw(observedImage, null);
    Matrix<Byte> observedDescriptors = descriptor.ComputeDescriptorsRaw(observedImage, null, observedKeyPoints);
    BruteForceMatcher<Byte> matcher = new BruteForceMatcher<Byte>(DistanceType.L2);
    matcher.Add(modelDescriptors);

    indices = new Matrix<int>(observedDescriptors.Rows, k);
    using (Matrix<float> dist = new Matrix<float>(observedDescriptors.Rows, k))
    {
        matcher.ID3Match(observedDescriptors, indices, dist, k, null);
        mask = new Matrix<byte>(dist.Rows, 1);
        mask.SetValue(255);
        Features2DToolbox.VoteForUniqueness(dist, uniquenessThreshold, mask);
    }

    int nonZeroCount = CvInvoke.cvCountNonZero(mask);
    if (nonZeroCount >= 4)
    {
        nonZeroCount = Features2DToolbox.VoteForSizeAndOrientation(modelKeyPoints, observedKeyPoints, indices,
        mask, 1.5, 20);
        if (nonZeroCount >= 4)
            homography = Features2DToolbox.GetHomographyMatrixFromMatchedFeatures(
            modelKeyPoints, observedKeyPoints, indices, mask, 2);
    }
}

```

Рисунок 3.1 – Метод «FAST»

У даній реалізації алгоритм ID3 використовується для оптимізації порядку, в якому випробовуються пікселі, в результаті чого виходить найбільш обчислювально ефективний детектор особливостей. Для отримання проміжних результатів додано допоміжний алгоритм, який буде виводити початкові зображення і зображення, що пройшли препроцесінг – з візуалізованими на них знайденими точками. Код допоміжного алгоритму представлений на рисунку 3.2, проміжні результати – на рисунку 3.3.

```

//Draw the matched keypoints
Image<Bgr, Byte> result = Features2DToolbox.DrawMatches(modelImage, modelKeyPoints, observedImage,
observedKeyPoints,
    indices, new Bgr(255, 255, 255), new Bgr(255, 255, 255), mask,
Features2DToolbox.KeypointDrawType.DEFAULT);

#region draw the projected region on the image
if (homography != null)
{ //draw a rectangle along the projected model
Rectangle rect = modelImage.ROI;
PointF[] pts = new PointF[] {
    new PointF(rect.Left, rect.Bottom),
    new PointF(rect.Right, rect.Bottom),
    new PointF(rect.Right, rect.Top),
    new PointF(rect.Left, rect.Top)};
homography.ProjectPoints(pts);

    result.DrawPolyline(Array.ConvertAll<PointF, Point>(pts, Point.Round), true, new Bgr(Color.Red), 5);
}
#endregion

return result;
}

```

Рисунок 3.2 – Алгоритм виведення проміжних результатів



Рисунок 3.3 – Візуалізація особливих точок методу FAST.

В результаті експериментів стало очевидно, що якість дескриптора залежить від розміру і роздільної здатності зображення. Наприклад, використання в якості вихідного зображення кола з чорною крапкою

всередині при роздільній здатності 329x333 дало мінімальну кількість особливих точок і на практиці такий маркер використовувати недоцільно. Початкове зображення і зображення з дескриптором зображені на рисунку 3.4.

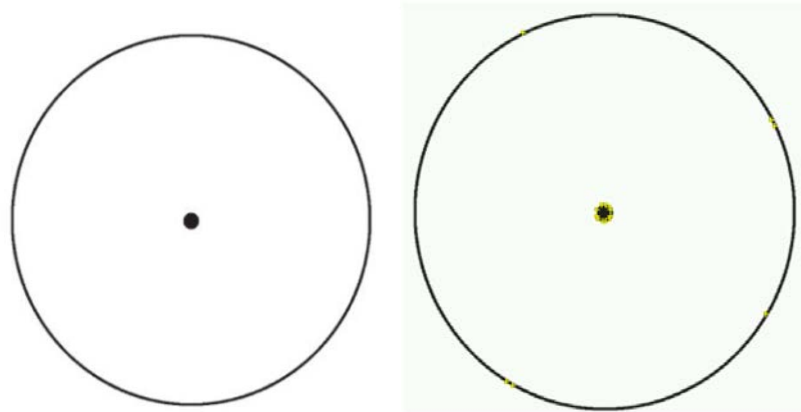


Рисунок 3.4 – Зразок маркера с дескриптором низької якості

Після успішної інтеграції бази даних в проект Unity необхідно, використовуючи модуль Vuforia, співвіднести маркери і віртуальні об'єкти, відповідні їм. Як віртуальний об'єкт буде виступати тривимірний текст, з координатами, які є відповідними локальним координатам відповідного маркера. Так само, як демонстрація можливостей інтерактивного характеру для кожного віртуального об'єкта був написаний свій активний скрипт для виведення додаткової інформації при натисканні на ідентифікований в відеопотоці маркер.

4 РОЗРОБКА ДОДАТКУ

4.1 Засоби розробки

4.1.1 Unity

Ігровий рушій – це середовище розробки програмного забезпечення, призначена для людей, що створюють відеоігри. Розробники використовують ігрові рушії для створення ігор для консолей, мобільних пристроїв і персональних комп'ютерів. Основні функціональні можливості, що зазвичай надаються ігровим рушієм, включають в себе механізм рендеринга («рендерер» або генерація / обробка графічного зображення) для 2D або 3D-графіки, фізичний рушій або виявлення зіткнень (і реагування на зіткнення), звук, сценарії, анімацію, штучний інтелект, мережеве взаємодія, потокову передачу, пам'ять управління, багатопоточність, підтримка локалізації, графічні сцени і може включати в себе підтримку відео для кінематографії.

У багатьох випадках ігрові рушії надають набір інструментів візуальної розробки на додаток до повторно використовуваних програмних компонентів. Ці інструменти можуть надаватися в інтегрованому середовищі розробки, що дозволяє спростити і прискорити розробку ігор на основі даних. Більшість ігрових рушіїв надають засоби, які полегшують розробку, такі як графіка, звук, фізика та функції штучного інтелекту. Ці ігрові рушії іноді називають «проміжним програмним забезпеченням», тому що, як і в діловому сенсі цього слова, вони надають гнучку і багаторазово використовувану програмну платформу, яка надає всі основні функціональні можливості, необхідні для розробки ігрової програми, при одночасному зниженні витрат.

Як і інші типи проміжного програмного забезпечення, ігрові рушії зазвичай надають абстракцію платформи, дозволяючи запускати одну і ту ж гру на різних платформах, включаючи ігрові приставки і персональні

комп'ютери, з невеликими змінами в вихідному коді гри, якщо такі є. Найчастіше ігрові рушії розробляються на основі компонентної архітектури, яка дозволяє замінювати або розширювати конкретні системи в рушії більш спеціалізованими (і часто більш дорогими) компонентами ігрового проміжного програмного забезпечення. Деякі ігрові рушії спроектовані як набір слабо пов'язаних компонентів проміжного програмного забезпечення, які можна вибірково комбінувати для створення власного рушія замість більш поширеного підходу до розширення або налаштування гнучкого інтегрованого продукту. Незважаючи на те, що розширюваність досягається, вона залишається високим пріоритетом для ігрових рушіїв через великого розмаїття областей застосування, для яких вони застосовуються. Незважаючи на специфіку назви, ігрові рушії часто використовуються для інших видів інтерактивних додатків з графічними потребами в реальному часі, такими як маркетингові демонстрації, архітектурні візуалізації, симуляції навчання і середовища моделювання.

Деякі ігрові рушії надають тільки можливості 3D-рендеринга в реальному часі замість широкого спектра функцій, необхідних для ігор. Ці двигуни покладаються на розробника ігор, щоб реалізувати решту цієї функціональності або зібрати її з інших компонентів проміжного програмного забезпечення гри. Ці типи двигунів зазвичай згадуються як «графічний рушій», «рушій рендеринга» або «тривимірний рушій» замість більш всеосяжного терміна «ігровий рушій». Ця термінологія використовується непослідовно, оскільки багато повнофункціональні ігрові рушії називаються просто «тривимірними двигунами». Сучасні ігрові або графічні рушії зазвичай надають графічні сцени, який є об'єктно-орієнтованим поданням ігрового світу 3D, який часто спрощує ігровий дизайн і може використовуватися для більш ефективного рендеринга великих віртуальних світів.

Гральний рушій – це центральний програмний компонент комп'ютерних та відео ігор або інших інтерактивних додатків з графікою,

оброблюваної в реальному часі. Він забезпечує основні технології, спрощує розробку і часто дає грі можливість запускатися на декількох платформах, таких як ігрові консолі та 16 настільні операційні системи, наприклад, GNU / Linux, Mac OS X і Microsoft Windows. Основну функціональність зазвичай забезпечує гральний рушій, що включає рушій рендеринга («відображувач»), фізичний рушій, звук, систему скриптів, анімацію, штучний інтелект, мережевий код, управління пам'яттю і багатопоточність [10]. Часто на процесі розробки можна заощадити за рахунок повторного використання одного грального рушія для створення безлічі різних ігор. Перший час через невисоку швидкість і відсутність будь-якої стандартизації апаратної частини навіть порти однієї гри серйозно переписувалися – а з однієї гри в іншу взагалі переносили мінімум коду. Втім, в іграх жанру квест існували Infocom'овська Z-Machine і SCI компанії Sierra – саме їх можна вважати першими закінченими гральними рушіями. Сам же термін «гральний рушій» з'явився в середині 1990-х років, головним чином у зв'язку з 3D-іграми, такими як шутери від першої особи. Ігри Doom і Quake від id Software виявилися настільки популярними, що інші розробники замість того, щоб працювати з чистого аркуша, ліцензували основні частини програмного забезпечення і створювали свою власну графіку, персонажів, зброю і рівні – «ігровий контент» або «ігрові ресурси». Рушій Quake був використаний в більш ніж десяти проектах і дав серйозний поштовх розвитку middleware-індустрії.

Пізніші ігри, такі як Unreal 1998 (рушій Unreal Engine) і Quake III Arena (на рушії id Tech 3) 1999 року, були спроектовані із застосуванням даного підходу, з окремо розробленими рушієм і наповненням. Практика ліцензування такої технології виявилася корисним допоміжним доходом для деяких розробників ігор. Так, вартість однієї ліцензії на комерційний ігровий рушій класу high-end може варіюватися від 10 тис. До 3,75 млн. \$ (в разі Warcraft III), а число ліцензіатів може досягати кілька десятків компаній (як для Unreal Engine). Принаймні, багаторазово використовувані рушії

прискорюють і спрощують розробку гри, що є цінною перевагою в конкуруючій індустрії комп'ютерних ігор. Подальше вдосконалення гральних рушіїв призвело до сильного поділу між рендерингом, скриптіном, художнім дизайном і дизайном рівнів. Зараз для типової команди розробників ігор є цілком звичайним мати у складі стільки ж художників, скільки і програмістів. Шутери від першої особи залишаються переважаючими користувачами сторонніх гральних рушіїв, але зараз такі рушії також використовуються в інших жанрах. Наприклад, RPG Morrowind і MMORPG Dark Age of Camelot засновані на рушії NetImmerse, в той час, як Oblivion і Fallout 3 використовують нову версію даної технології – Gamebryo. Відома MMORPG Lineage II побудована на рушії Unreal Engine 2 (незважаючи на те, що даний рушії спочатку призначався для використання в шутерах). Гральні рушії також використовуються в іграх, спочатку розроблених для ігрових консолей; наприклад, рушії RenderWare використовується у франчайзі Grand Theft Auto і Burnout.

Marmalade. Демо-версія: доступна 90 денна тестова копія для некомерційного використання. Цей ігровий рушії являє собою досить професійний інструмент, для серйозних ігор. Marmalade має досить складний інтерфейс, що підкреслює необхідність у володінні високим рівнем підготовки розробника. Серед створених на його базі ігор можна виділити такі: Pro Evolution Soccer (PES). Такі приклади є досить популярними і якісними, від відомих компаній, які використовували Marmalade, спираючись на його професійні якості. Однак Marmalade швидше є не рушії, а середовище для створення своїх рушіїв (приклад нижче). Крім того він вам дозволяє використовувати різні наявні у розробника (або у відкритому доступі) C / C ++ бібліотеки.

Corona – це 2D рушії для створення простих ігор на мобільні платформи у дусі Angry Birds. Corona має простий інтерфейс що значно полегшує процес розробки гри. Як приклад можна привести Bubble Ball, яку написав 14 річний хлопець з Америки. У Corona досить обширне API на всі

випадки життя, що дозволить вам з легкістю реалізувати всі потреби розробника. Однак, все в API передбачити неможливо і цілком імовірно, що рано чи пізно розробнику захочеться скористатися, якоюсь нативною (звичною для конкретної платформи) можливістю Android, або IOS. Тут на розробника буде чекати розчарування – Corona не має таких можливостей. Проте для флеш-розробників є приємна новина. Творці рушія стверджують, що тим, хто пише гри на флеш, не важко перейти на Corona, тому вони дуже схожі.

ShiVa3D – це рушій з вбудованим візуальним редактором (як в Unity3d). Для реалізації мультиплатформеності рушій використовує Marmelade (дивимося вище). У рушія багато вбудованих можливостей: є сторонні плагіни і можливість використання бібліотек написаних нативною для платформи мовою (скажімо Java для Android).

Нещодавно з'явилася свіжа версія AIR, в якій є можливість розробки під IOS, Android і BlackBerry PlayBook. Нова версія принесла більше стабільності і продуктивності. Хоча серед демонстрації можливостей рушія були вказані лише звичайні програми (не ігри). Також поки відсутня можливість використання нативних бібліотек. А сам API теж вельми бідний на можливості.

Зробивши огляд деяких відомих ігрових рушіїв (таблиця 4.1) було прийнято рішення використовувати рушій Unity. Таке рішення базується на таких критеріях:

- вартість. Використовуючи Unity у своїх цілях, розробник користується їм безкоштовно;
- мови програмування. C# і JavaScript являють собою досить розповсюджене використання;
- функціонал. Так як Unity підтримує можливість завантаження плагінів, функціонал цього рушія може бути необмеженим;
- інтерфейс. Інтерфейс Unity є досить інтуїтивним, що полегшує роботу з ним;

- мультиплатформеність. Завдяки Unity можна легко і якісно створювати ігри і додатки на велику кількість відомих платформ.

Таблиця 4.1 – Порівняння ігрових рушіїв

Критерії	Платформи, що підтримуються	Інтерфейс (оцінка від 0 до 5 Де 5 – інтуїтивно)	Підтримка різних мов програмування	Підтримка додаткових сервісів (оцінка від 0 до 5 Де 5 – відмінно)	Вартість
Marmelade	IOS Android Symbian, Bada, WebOS	2	C/C ++	4	\$499/ рік
Unity	IOS, Android, персональні комп'ютери, веб, ігрові приставки	4	C #, JavaScript, Boo	5	Free + \$ 400 / рік
Flash	IOS, Android BlackBerry PlayBook	2	Flash	3	Free
ShiVa3D	IOS, Android, Windows, Mac OS, Linux, Palm, Wii, веб	4	Lua	2	Free + €169.00
Corona	IOS, Android	5	Lua	1	\$ 199 / рік

Unity – це кросплатформене середовище розробки. Unity дозволяє створювати додатки, що працюють під більш ніж 20 різними операційними системами, що включають персональні комп'ютери, ігрові консолі, мобільні

пристрої, інтернет-додатки та інші. Випуск Unity відбувся в 2005 році і з того часу йде постійний розвиток.

Основними перевагами Unity є наявність візуальної середовища розробки, міжплатформеної підтримки і модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатоконпонентними схемами і труднощі при підключенні зовнішніх бібліотек, однак оскільки для розробки додатку це не є основним, такий недолік не є критичним.

На Unity написані тисячі ігор, додатків та симуляцій, які охоплюють безліч платформ і жанрів. При цьому Unity використовується як великими розробниками, так незалежними студіями.

Редактор Unity має простий Drag & Drop інтерфейс, представлений на рисунку 4.1, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Рушій підтримує два скриптових мови: C #, JavaScript (модифікація). Раніше була підтримка Воо (діалект Python), але його прибрали в 5-й версії. Розрахунки фізики виробляє фізичний рушій PhysX від NVIDIA.

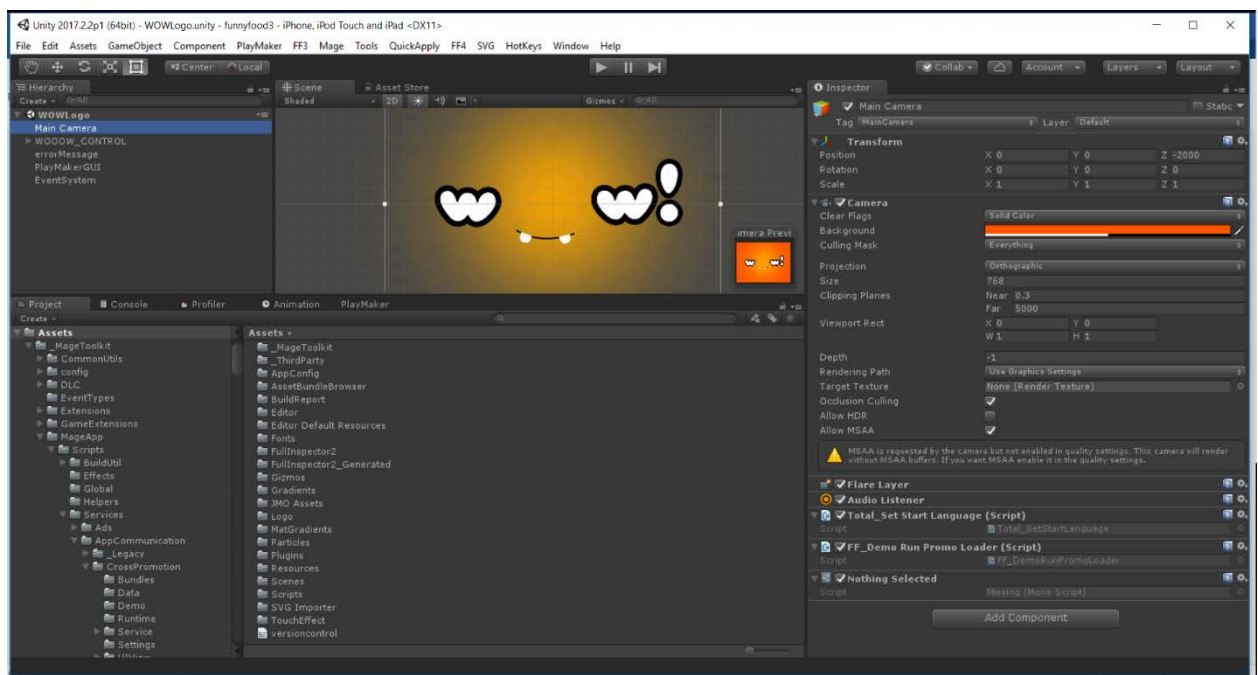


Рисунок 4.1 – Інтерфейс Unity

Інтерфейс Unity включає в себе вікно проекту, вікно сцени, вікно гри, вікно ієрархії, вікно інспектора, панель інструментів, консольне вікно, та опціональні вікна сторонніх плагінів.

До об'єктів можна застосовувати колізії (в Unity так звані Колайдери – collider), яких існує декілька типів.

Також Unity підтримує фізику твердих тіл і тканини, а також фізику типу Ragdoll (тряпична лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

При імпорті текстури в Unity можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна – буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання і редагування шейдерів. Редактор Unity має компонент для створення анімації, але також анімацію можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли.

Unity 3D підтримує систему Level Of Detail, суть якої полягає в тому, що на далекій відстані від гравця високодеталізовані моделі замінюються на менш деталізовані, і навпаки, а також систему Occlusion culling, суть якої в тому, що у об'єктів, які не потрапляють в поле зору камери не візуалізується геометрія і колізія, що знижує навантаження на центральний процесор і дозволяє оптимізувати проект. При компіляції проекту створюється виконуваний (.exe) файл гри (для Windows), а в окремій папці – дані гри (включаючи всі ігрові рівні і спільні бібліотеки).

Рушій підтримує безліч популярних форматів. Моделі, звуки, текстури, матеріали, скрипти можна запаковувати в формат .unityassets і передавати іншим розробникам, або викладати у вільний доступ. Цей же формат

використовується у внутрішньому магазині Unity Asset Store, в якому розробники можуть безкоштовно і за гроші викладати в загальний доступ різні елементи, потрібні при створенні ігор. Щоб використовувати Unity Asset Store, необхідно мати акаунт розробника Unity. Unity має всі потрібні компоненти для створення мультиплеєра. Також можна використовувати відповідний користувачеві спосіб контролю версій. Наприклад, Tortoise SVN або Source Gear.

У Unity входить Unity Asset Server – інструментарій для спільної розробки на базі Unity, є доповненням, що додає контроль версій і ряд інших серверних рішень.

Як правило, ігровий рушій надає безліч функціональних можливостей, що дозволяють їх задіяти в різних іграх, в які входять моделювання фізичних середовищ, карти нормалей, динамічні тіні і багато іншого. На відміну від багатьох ігрових рушіїв, у Unity є дві основні переваги: наявність візуальної середовища розробки та міжплатформенна підтримка. Перший фактор включає не тільки інструментарій візуального моделювання, а й інтегровану середу, лінію складання, що направлено на підвищення продуктивності розробників, зокрема, етапів створення прототипів і тестування. Під міжплатформенною підтримкою надається не тільки місце розгортання (установка на персональному комп'ютері, на мобільному пристрої, консолі і т.д.), а й наявність інструментарію розробки (інтегроване середовище може використовуватися під Windows і Mac OS).

Третьою перевагою є модульна система компонентів Unity, за допомогою якої відбувається конструювання ігрових об'єктів, коли останні є комбінованими пакетами функціональних елементів. На відміну від механізмів успадкування, об'єкти в Unity створюються за допомогою об'єднання функціональних блоків, а не переміщення в вузли дерева успадкування. Такий підхід полегшує створення прототипів, що актуально при розробці ігор.

Як недоліки наводяться обмеження візуального редактора при роботі з

багатокомпонентними схемами, коли в складних сценах візуальна робота ускладнюється. Іншим недоліком є відсутність підтримки Unity посилань на зовнішні бібліотеки, роботу з якими програмістам доводиться налаштовувати самостійно, і це також ускладнює командну роботу. Ще один недолік пов'язаний з використанням шаблонів примірників (англ. Prefabs). З одного боку, ця концепція Unity пропонує гнучкий підхід візуального редагування об'єктів, але з іншого боку, редагування таких шаблонів є складним.

4.1.2 Vuforia

Проведено порівняльний аналіз найбільш відомих бібліотек доповненої реальності: ARCore, ARToolKit, EasyAR, Kudan, Maxst, Vuforia, Xzing, Wikitude, ThingWorx Studio. З цього списку можна виключити ARCore, тому що ця платформа підходить для обмеженої кількості смартфонів.

Всі представлені інструменти повинні мати сумісність з міжплатформним середовищем розробки Unity. Unity дозволяє створювати проекти як в 2D, так і в 3D, підтримує дві скриптових мови: C #, JavaScript. Це середовище розробки містить в собі редактор сцен, редактор об'єктів і редактор скриптів. Для створення реалістичної поведінки об'єктів в Unity вбудований фізичний рушій, що забезпечує компонентами для симуляції фізичної поведінки об'єктів. При розробці використовується компонентно-орієнтований підхід до програмування. Необхідність використання Unity обумовлена тим, що це середовище дозволяє компілювати проекти під операційні системи Android і iOS, спрощує роботу з 3D моделями. Коротко розглянемо кожен платформу [11].

ARToolKit є набором програмних бібліотек, які можуть використовуватися в AR (augmented reality) додатках. Головна перевага бібліотеки – відкритий вихідний код. Відсутня WaterMark. WaterMark або водяний знак – це напівпрозоре зображення, що накладається поверх зображення з камери. Найчастіше водяний знак представлений логотипом

компанії і присутній в безкоштовних ліцензіях бібліотек доповненої реальності. Є можливість розпізнавати 2D об'єкти. Незважаючи на безкоштовний доступ, документація для розробників вельми обмежена. Відсутня можливість розпізнавання об'ємних об'єктів і мітки з хмари [13].

EasyAR – безкоштовна і проста у використанні бібліотека. Має можливість розпізнавання тільки 2D об'єктів. Відсутня WaterMark. Є хорошою альтернативою Vuforia за якістю розпізнавання міток. Підтримується до 1000 міток для розпізнавання. Відсутні хороші навчальні посібники, вивчення можливо тільки за документацією, що не зручно і не продуктивно.

Kudan – до позитивних можливостей можна віднести відсутність обмежень на кількість розпізнаваних зображень і маленький обсяг пам'яті, необхідний для зберігання файлів на пристрої. Існує можливість розпізнавання 3D об'єктів, безмаркерне відстеження об'єктів. Розробники можуть скористатися базовою документацією при зверненні до бібліотеки. Однак керівництво з використання не дуже докладний і вимагає пошуку додаткової інформації. Відображення доповнень реалізується через окремий компонент-обгортку над OpenGL. Є WaterMark. Але безкоштовна версія призначена тільки для тестування додатків. Для публікації додатка потрібна покупка ліцензії [14].

Maxst – оптимізований саме для мобільних платформ. Розпізнає і відстежує зображення, але одночасно максимально тільки 3 мети. Є можливість відстежити плоску поверхню і розмістити на ній потрібний об'єкт. Безкоштовна версія дозволяє вести розробку тільки для ОС Android і iOS. Є підтримка одночасного створення і збереження тривимірної карти простору. Maxst працює тільки з 32-розрядної версії редактора Unity. Maxst володіє не дуже гарною якістю розпізнавання міток.

Xzing – є три продукти для роботи з додатками на основі доповненої реальності: Augmented Face, Augmented Vision і Magic Face. Xzing Augmented Face розпізнає і відстежує особи. Xzing Augmented Vision

розпізнає і відстежує плоскі зображення. Xzing Magic Face призначений для заміни рис обличчя і нанесення макіяжу. Є можливість розробки проекту під HTML5. Наявність WaterMark. Безкоштовна пробна версія доступна тільки для демонстрації (інвертує колір і змінює зображення) [15].

Wikitude має функцію Extended Tracking для збереження положення мітки на мобільному пристрої, навіть якщо вона за межами огляду камери. Підтримується розпізнавання як 2D, так і 3D форматів. Існує можливість сканування реального об'єкта для подальшого розпізнавання. Є хороша документація, але при цьому мала кількість навчального матеріалу. Але безкоштовна версія надається тільки для тестування. Для подальшої розробки та публікації необхідно придбання ліцензії [16].

ThingWorx Studio – універсальний інструмент для створення і публікації продуктів доповненої реальності. Дозволяє розробляти програми для промислових підприємств, використовує зрозумілий графічний редактор. У ThingWorx Studio використовується технологія комп'ютерного зору Vuforia. Мітки для розпізнавання і 3D об'єкти зберігаються і використовуються через хмарну базу даних. ThingWorx Studio єдине з представлених середовищ розробки дозволяє безпосередньо використовувати САПР моделі. Але даний інструментарій має тільки пробну безкоштовну версію, для подальшої розробки та публікації необхідно придбання ліцензії [17].

Vuforia – одна з найпопулярніших в світі платформ доповненої реальності. Існує велика кількість навчального матеріалу як в текстовому, так і в відео форматі. Програмне забезпечення реалізує наступні функції: розпізнавання різних типів візуальних об'єктів (куб, циліндр, площина), розпізнавання тексту і навколишнього середовища, VuMark (комбінація зображення і QR-коду). Процес розпізнавання команд може бути реалізований з використанням локальної або хмарної бази даних. Обмеження безкоштовної версії відносяться тільки до числа VuMark, кількості взаємодій

з хмарної базою даних і наявності WaterMark. Vuforia підтримує Extended Tracking. Є можливості розпізнавання 3D об'єктів і підтримки Virtual Button.

Виходячи з наведеного аналізу, зроблений вибір інструменту розробки програмного забезпечення доповненої реальності Vuforia. Дана платформа найбільш популярна, а тому більш доступна для вивчення. На даний момент власником Vuforia є компанія PTC, що займається впровадженням в роботу підприємств доповненої реальності, інтернету речей, 3D-друку, цифрових двійників, що дає надію на подальшу активну підтримку і розвиток AR SDK Vuforia. Будучи повноцінною SDK для розробників, Vuforia являє собою набір інструментів для створення додатків доповненої реальності. Vuforia підтримує:

- розпізнавання декількох цілей одночасно (включаючи об'єкти, зображення і текст);
 - відстеження цілей;
 - розпізнавання 2D і 3D форматів;
 - сканування реального об'єкта для подальшого розпізнавання;
 - віртуальні кнопки;
 - відображення додаткових елементів через OpenGL;
 - Smart Terrain™ – можливість реконструювати навколишній ландшафт, створюючи його 3D-карту;
 - Extended Tracking – можливість продовжити відображення мети на мобільному пристрої, навіть коли вона знаходиться поза полем зору.

Зокрема, при розпізнаванні зображень Vuforia дозволяє мобільним додаткам використовувати дані, що знаходяться або на пристрої, або в хмарі. До основних переваг бібліотеки можна віднести підтримку пристроїв віртуальної реальності, а також тестовий додаток з супроводжуючими коментарями, в якому показані можливості бібліотеки. Однак відсутність повноцінного керівництва з використання бібліотеки ускладнює перший досвід роботи з Vuforia. Окремі інструкції і короткі поради представлені у

великій кількості, але не впорядковані і тому не замінюють розробнику необхідну документацію [18][19].

Огляд існуючих додатків показав основні тенденції в розвитку технології доповненої реальності. Ця технологія в наш час дуже бурхливо розвивається і може знайти застосування в багатьох областях. Аналіз додатків показав, що зараз вони мають, в основному, або розважальний, або вузькоспрямований характер. Основна увага в таких додатках приділяється їх можливостям при взаємодії з навколишнім середовищем і користувачем. Огляд альтернатив та визначення основних критеріїв аналізу зображень [20] та урахування найбільш важливих критеріїв показали, що найоптимальнішим середовищем для розробки системи є Vuforia. Vuforia – безкоштовна бібліотека, постійно модернізується, дозволяє працювати як з нативними додатками, так і створювати кросплатформені програми за допомогою спеціального ігрового рушія Unity.

4.2 Розробка додатку

Для інсталяції Unity необхідно завантажити інсталяційний файл з офіційного сайту Unity (<http://unity3d.com/unity>). Перед завантаженням розробнику необхідно вирішити: завантажувати безкоштовну версію, чи професійну (треба платити 75\$ щомісяця). Після того, як розробник вирішив, він має обрати необхідне поле на сайті. Найкращим варіантом для студентів і початківців, а також для всіх, хто вже має досвід роботи з ігровими рушіями, але тільки починає працювати з Unity, пропонується завантажити безкоштовну версію. При першому завантаженні розробнику надається місяць безкоштовного використання усіх функцій, за який розробнику надається змога зрозуміти, потрібні йому платні функції, чи ні. Отже, в цій роботі буде використовуватися безкоштовна версія. Поки файл завантажується необхідно зареєструвати акаунт на сайті.

Після завантаження, розробник запускає файл і проходить усі кроки інсталяції, по завершенню якої, буде відкрито вікно. У нього необхідно ввести дані з акаунту, який зареєстрували на сайті. Підсумувавши все, написано вище, інсталяцію можна розбити на такі кроки:

- завантаження інсталяційного файлу;
- реєстрація на сайті Unity;
- інсталяція завантаженого файлу;
- активація програми.

Після виконання усіх цих кроків, інсталяцію програми можна вважати завершеною.

Якщо користувач обирає проект, з яким він працював, він одразу відкривається. Якщо було обрано новий проект, то з'являється нове вікно, у якому треба дати проекту назву, обрати місце його збереження і обрати тип гри 2D чи 3D. Після цього кроку з'являється вікно з головним робочим інтерфейсом.

Головний інтерфейс поділено на 5 зон:

- 1 Зона. У ній знаходяться всі фізичні об'єкти, з якими в подальшому працює розробник;
- 2 Зона. Це Сцена, на якій іде відображення усіх фізичних об'єктів. Саме на сцені вони розміщуються просторово;
- 3 Зона. Це файлова структура гри;
- 4 Зона. Через цю зону можна працювати з файлами, які знаходяться в зоні №3;
- 5 Зона. Це усі характеристики об'єкта, який обрано в зоні №1.

Перше, що необхідно зробити, щоб працювати з Vuforia – зареєструватися на сайті розробників Vuforia. Тільки після цього розробник матиме доступ до всіх файлів, які знадобляться для роботи з Vuforia. Після реєстрації можна завантажувати пакети розробки додатків на Android та iOS, їх можна знайти, у розділі «Downloads» -> «SDK». Та є ще один пакет, який потім імпортується в Unity, який розглянуто вище, і після цього можна

розробляти проекти під Android і iOS одночасно. Завантаживши пакет для Unity, треба його імпортувати в ігровий рушій.

Спочатку було розроблено схему роботи додатку. Додаток повинен мати можливість ідентифікувати зображення у відеопотоці, виводити назву на зображення, та відкривати можливість перегляду додаткової інформації. Вікно додаткової інформації повинно мати можливість виходу та кнопки пошуку інформації по зображенню в інтернеті. Таким чином орієнтовна блок-схема роботи додатку буде мати вигляд, що представлений на рисунку 4.2.

Розробивши структуру роботи необхідно визначитися з необхідною архітектурою проекту. Виходячи з поставленої задачі коренева структура проекту мати 5 основних директорій, представлених на рисунку 4.3.

У директорії Modules знаходяться три основних Third-Party модулі:

- Demigant;
- TextMesh Pro;
- Vuforia.

Demigant – модуль для взаємодії функціонального пересування, контролю на налаштування ротації та змінення розмірів об'єктів у реальному часі. Цей модуль необхідний для полегшення роботи при написанні механізмів візуалізації.

TextMesh Pro – допоміжний модуль для роботи зі шрифтами та UI текстовими компонентами. Має можливість роботи із мультимовністю. У додатку необхідний для візуалізації описів ідентифікованих картин.

Vuforia – безпосередньо допоміжний модуль для взаємодії трекінгу та допоміжними даними.

У директорії StreamingAssets знаходиться база даних маркерів, що дають можливість ідентифікувати зображення.

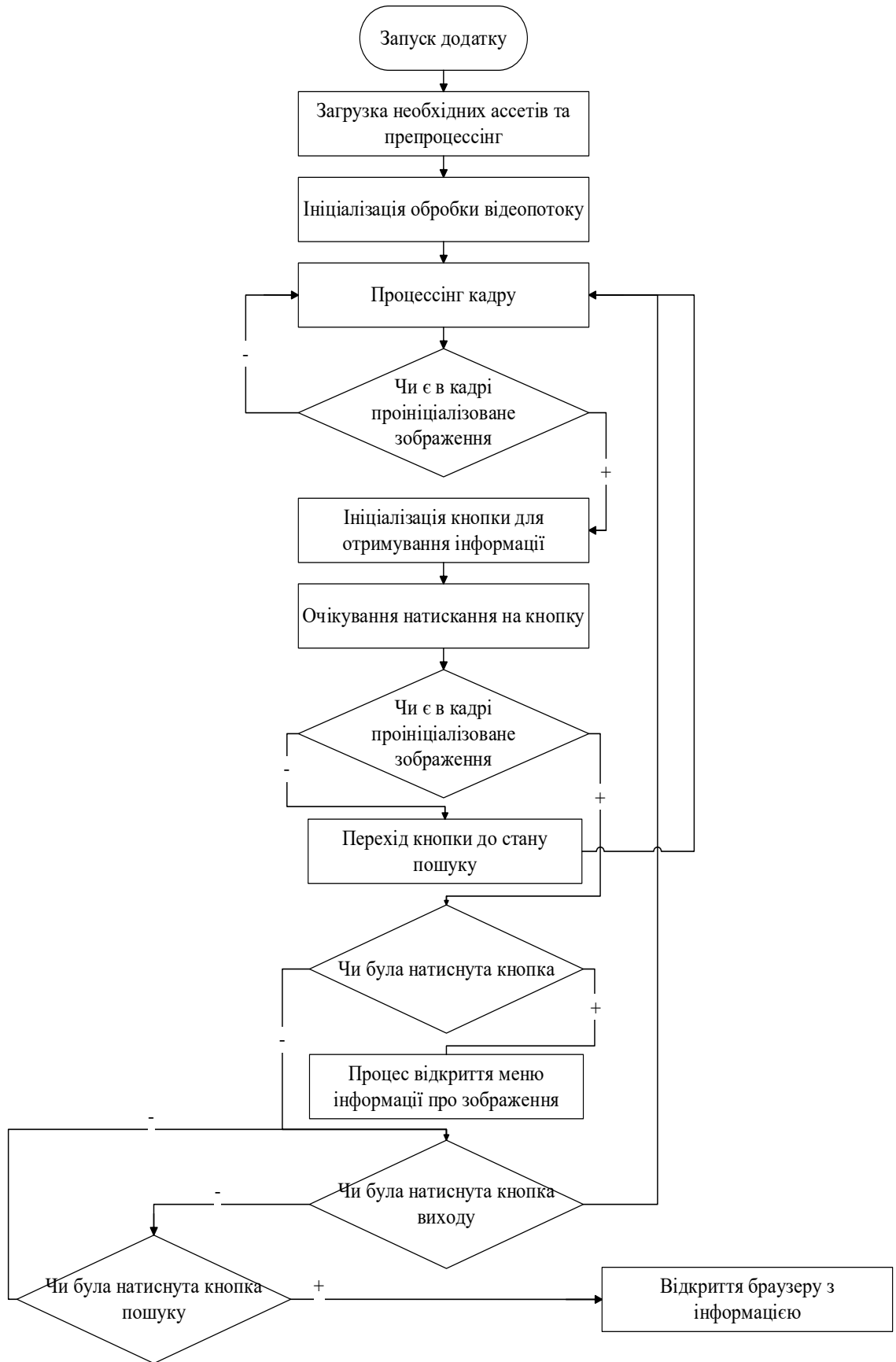


Рисунок 4.2 – Блок-схема роботи додатку

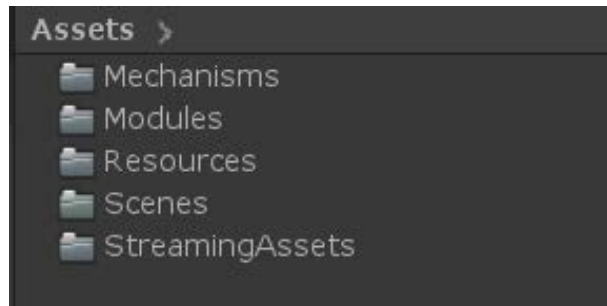


Рисунок 4.3 – Кореневі директорії проекту

Директорія Mechanisms включає у себе 5 основних механізмів:

- Bootstrap;
- DescriptionManager;
- MuseumManager;
- TrackingSystem;
- UIController.

Для початку необхідно розробити базу даних зображень та описів до них. Для цього був реалізований ScriptableObject DescriptionDataStorage, що серіалізує у собі ліст класу ArtDescription, що містить у собі назву зображення, спрайт зображення, роки створення та код трекінгу. Код представлений на рисунку 4.4. Також необхідно було реалізувати механізм пошуку по серіалізованому об'єкту з ціллю знаходження необхідних даних по трекінговому коду. Для цього був реалізований механізм GetArtByCode.

Для доступу до бази даних необхідно було реалізувати менеджер, що представлений статичним класом та має в собі реалізацію методів доступу до бази даних. Реалізація класу DescriptionManager представлена на рисунку 4.5.

Наступним кроком стала реалізація універсального контролеру UI у додатку. Основними блоками роботи є кнопки видачі інформації про ідентифіковане зображення та меню додаткової інформації. До контролеру меню основної інформації потрібно реалізувати методи появи, скриття, візуалізації додаткової інформації та метод відкриття інформації у браузері. Таким чином, основні методи представлені на рисунку 4.6.

```

namespace Assets.Mechanisms.DescriptionManager.Scripts
{
    [CreateAssetMenu(fileName = "DescriptionDataStorage", menuName = "DescriptionDataStorage", order = 51)]
    6 references
    public class DescriptionDataStorage : ScriptableObject
    {
        [SerializeField] private List<ArtDescription> Arts;

        1 reference
        public ArtDescription GetArtByCode(string trackCode)
        {
            foreach (var artDescription in Arts)
            {
                if (artDescription.TrackingCode == trackCode)
                {
                    var result = new ArtDescription(artDescription.Name, artDescription.Years, artDescription.Image, artDescription.TrackingCode);
                    return result;
                }
            }
            return null;
        }

        [System.Serializable]
        7 references
        public class ArtDescription
        {
            public string Name;
            public Sprite Image;
            public string Years;
            public string TrackingCode;

            1 reference
            public ArtDescription(string Name, string Years, Sprite Image, string TrackingCode)
            {
                this.Name = Name;
                this.Years = Years;
                this.Image = Image;
                this.TrackingCode = TrackingCode;
            }
        }
    }
}

```

Рисунок 4.4 – Реалізація бази даних зображень

```

namespace Assets.Mechanisms.DescriptionManager.Scripts
{
    4 references
    public class DescriptionManager
    {
        private DescriptionDataStorage descriptionDataStorage;
        private DescriptionDataStorage.ArtDescription currentArtDescription;

        1 reference
        public DescriptionManager(DescriptionDataStorage descriptionDataStorage)
        {
            this.descriptionDataStorage = descriptionDataStorage;
        }

        2 references
        public DescriptionDataStorage.ArtDescription GetArtDescription(string findCode)
        {
            currentArtDescription = descriptionDataStorage.GetArtByCode(findCode);
            return currentArtDescription;
        }

        2 references
        public DescriptionDataStorage.ArtDescription GetCurrentDescription()
        {
            return currentArtDescription;
        }
    }
}

```

Рисунок 4.5 – Реалізація менеджера доступу до бази даних зображень

```

public void HideInfoPanel()
{
    behaviourCoroutine = HideSequence();
    StartCoroutine(behaviourCoroutine);
}
0 references
public void ShowInfoPanel()
{
    AppUIController.SetBusy(value: true);
    var description = MuseumManager.DescriptionManager.GetCurrentDescription();
    artPanelImage.Initialize(description.Image);
    artPanelDescriptionText.Initialize(description.Name, description.Years);
    HideAllElements();
    this.gameObject.SetActive(value: true);
    behaviourCoroutine = ShowSequence();
    StartCoroutine(behaviourCoroutine);
}
0 references
public void OpenURL()
{
    var artName = MuseumManager.DescriptionManager.GetCurrentDescription().Name;
    artName = artName.Replace(oldValue: " ", newValue: "+");
    var link = "https://www.google.com/search?client=psy-ab&btnG=Search&q=" + artName+"art";
    Application.OpenURL(link);
}
1 reference
private void HideAllElements()
{
    var artImage = artPanelImage.GetComponent<Image>();
    var artImageColor = artImage.color;
    artImageColor.a = 0;
    artImage.color = artImageColor;
    var panelBackgroundImage = this.GetComponent<Image>();
    var panelBackgroundColor = panelBackgroundImage.color;
    panelBackgroundColor.a = 0;
    panelBackgroundImage.color = panelBackgroundColor;
    var descriptionColor = artPanelDescriptionText.GetComponent<TextMeshProUGUI>().color;
    descriptionColor.a = 0;
    artPanelDescriptionText.GetComponent<TextMeshProUGUI>().color = descriptionColor;
    strockout.transform.localScale = new Vector3(x:0, y:1.2f, z:1);
    closeButton.color = new Color(closeButton.color.r, closeButton.color.g, closeButton.color.b, a:0);
    searchButton.color = new Color(searchButton.color.r, searchButton.color.g, searchButton.color.b, a:0);
}

```

Рисунок 4.6 – реалізація методів класу InfoPanel

Наступним кроком стала реалізація кастомного хенлеру для обробки івентів детекції зображень та їх виходу із зони пошуку. Клас CustomTrackableEventHandler є наслідником класів MonoBehaviour та інтерфейсу ITrackableEventHandler. При ініціалізації цього компоненту проходить реєстрація підписок TrackableEventHandler компоненту та ініціалізація TextMesh – генерація об'єкту з компонентом тексту за заданим набором параметрів відображення до кожного із об'єктів трекінгу. Даний алгоритм необхідний для того, щоб при ідентифікації об'єкту у відеопотоці на ідентифікований об'єкт можна було зпроєціювати його назву. При реєстрації івенту трекінгу маркеру викликається метод OnTrackingFound(),

що викликає метод UpdateTextMesh, що виводить на ідентифіковане зображення необхідний текст у якості параметру. Та викликає метод OnItemTracked класу MuseumManager. При загубленні зображення у зоні пошуку викликається метод OnTrackingLost, що відключає текст на зображенні та викликає метод OnItemLost, що відключає кнопку отримання додаткової інформації на екрані. Реалізація реєструючого методу представлена на рисунку 4.7.

```

1 reference
public void OnTrackableStateChanged(
    TrackableBehaviour.Status previousStatus,
    TrackableBehaviour.Status newStatus)
{
    Debug.Log(message: ">>>> " + previousStatus + " " + newStatus);
    m_PreviousStatus = previousStatus;
    m_NewStatus = newStatus;

    if (newStatus == TrackableBehaviour.Status.DETECTED ||
        newStatus == TrackableBehaviour.Status.TRACKED)
    {
        Debug.Log(message: "Trackable " + mTrackableBehaviour.TrackableName + " found");
        OnTrackingFound();
    }
    else if (previousStatus == TrackableBehaviour.Status.TRACKED &&
        (newStatus == TrackableBehaviour.Status.NO_POSE ||
         newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED))
    {
        Debug.Log(message: "Trackable " + mTrackableBehaviour.TrackableName + " lost");
        OnTrackingLost();
    }
    else
    {
        OnTrackingLost();
    }
}
1 reference

```

Рисунок 4.7 – Реалізація реєструючого методу пошуку зображення у відеопотоці.

Після реалізації основних модулів необхідно заповнити базу даних зображеннями та дескрипторами для них. Для цього було необхідно застосувати механізми Vuforia. Була проініціалізована нова база даних та заповнена тестовими зображеннями. Фрагмент контенту бази даних представлений на рисунку 4.8.

KTURE_AI_DIPLOMA_DATABASE_DEVICE Edit Name
Type: Device

Targets (5)

Add Target Download Database (All)

Target Name	Type	Rating	Status	Date Modified
Bogatyri	Single Image	★★★★★	Active	Dec 01, 2019 21:07
The_Kiss	Single Image	★★★★★	Active	Dec 01, 2019 21:03
Vavilon_Tower	Single Image	★★★★☆	Active	Dec 01, 2019 21:02
Adam	Single Image	★★★★☆	Active	Dec 01, 2019 11:39
Nascita_di_Venere	Single Image	★★★★★	Active	Dec 01, 2019 10:53

Рисунок 4.8 – Фрагмент бази даних додатку

Після апробації додатку виникла необхідність мати можливість роботи з хмаровою базою даних. Ця необхідність виникла на підставі взаємозалежності кількості зображень та ваги додатку на девайсі. Vuforia дає можливість роботи з базою даних, що знаходиться не тільки локально, а ще і на сервері. Для адаптації до додатку цієї можливості був реалізований клас CloudRecoBehaviour. Основна реалізація представлена на рисунку 4.9.

```
protected override void Initialize()
{
    this.mTargetFinder = this.mObjectTracker.GetTargetFinder<ImageTargetFinder>();
    if (this.mTargetFinder != null && ((ImageTargetFinder) this.mTargetFinder).StartInit(this.AccessKey, this.SecretKey))
        this.mCurrentlyInitializing = true;
    else
        Debug.LogError((object) "CloudRecoBehaviour: TargetFinder initialization failed!");
}

protected override void Deinitialize()
{
    if (this.mTargetFinder == null)
        return;
    this.mCurrentlyInitializing = !this.mTargetFinder.Deinit();
    if (this.mCurrentlyInitializing)
        Debug.LogError((object) "CloudRecoBehaviour: TargetFinder deinitialization failed!");
    else
        this.mInitSuccess = false;
}

protected override bool IsRecoAvailable()
{
    if (!VuforiaConfiguration.Instance.Vuforia.IsUFOEnabled())
        return true;
    Debug.LogError((object) "Cloud Reco is not available");
    return false;
}

private void OnValidate()
{
    this.AccessKey = this.AccessKey.Trim();
    this.SecretKey = this.SecretKey.Trim();
}
}
```

Рисунок 4.9 – Реалізація можливості роботи з базою даних на сервері

5 АНАЛІЗ РЕАЛІЗОВАНОГО ПРОДУКТУ

Після успішної реалізації був проведений процес апробації додатку. Результати атестаційної роботи також були апробовані на Міжнародній науково-практичній конференції та була видана у друк стаття [21]. Додаток був протестований у низці різноманітних пристроїв на платформах iOS та Android. Результати тестування показали, що додаток працює стабільно і не викликає системних помилок, що призводять до закриття чи збоїв.

Реалізований продукт відповідає усім вимогам, що були поставлені у розрізі аналізу області та маркетингового огляду:

- невелика «вага» додатку;
- стабільна робота на різноманітних пристроях;
- кросплатформеність;
- можливість роботи без підключення до інтернету;
- безперервна робота у режимі відеопотоку;
- можливість отримувати назву зображення при фокусі на ньому;
- можливість отримати базову додаткову інформацію про зображення не виходячи з додатку;
- можливість отримати розширену інформацію про додаток з виходом у браузер.

Базовий інтерфейс додатку є відеопотоком камери на пристрої та інтерактивною кнопкою у активному стані та кнопкою-підказкою у пасивному. У пасивному стані текст на кнопці підказує користувачу, що треба знайти картину. Візуалізація інтерфейсу у цьому стані представлена на рисунку 5.1.



Рисунок 5.1 – Інтерфейс додатку у стані пошуку

У разі того, якщо у рамках відеопотоку з'явиться зображення, яке можливо ідентифікувати, то кнопка змінить свій стан та стане інтерактивною та текст на ній зміниться на Get Info. Також під ідентифікованим зображенням з'явиться його назва. Візуалізація цього стану представлена на рисунку 5.2.

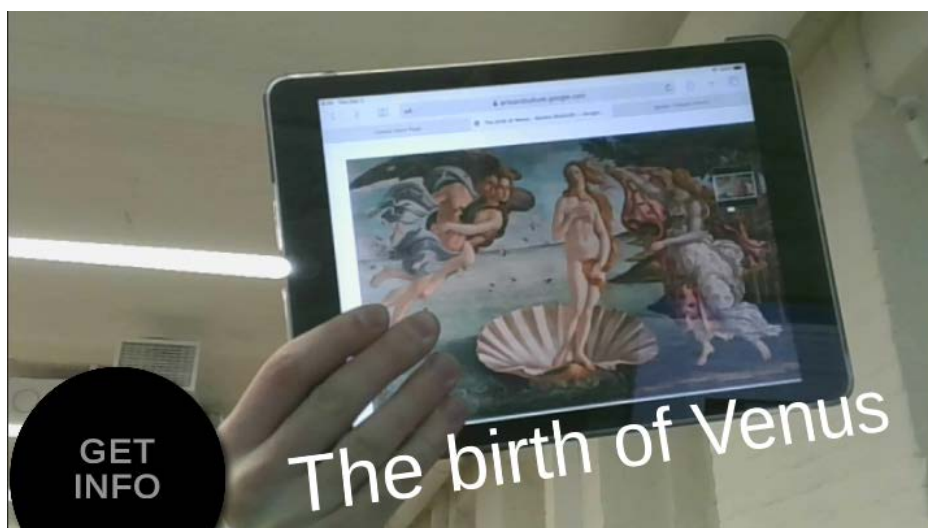


Рисунок 5.2 – Інтерфейс додатку у активному стані

У разі, якщо користувач натисне на кнопку Get Info, то у додатку запуститься вікно додаткової інформації, що містить у собі зображення у кращій якості, назву картини, автора та роки її створення. На цьому меню також є два функціональних юніти – кнопка виходу та кнопка пошуку інформації в інтернеті. При натисканні на кнопку виходу – додаток повернеться до стану обробки відеопотоку. При натисканні на кнопку пошуку інформації в інтернеті – відкриється вікно браузера із запитом про дану картину. Візуалізація меню додаткової інформації представлена на рисунку 5.3.



Рисунок 5.3 – Меню додаткової інформації про картину

Оскільки розробка додатку була зорієнтована на мобільні платформи було важливо дотримуватись високої швидкодії продукту. На рисунку 5.4 представлений графік швидкодії додатку по CPU.

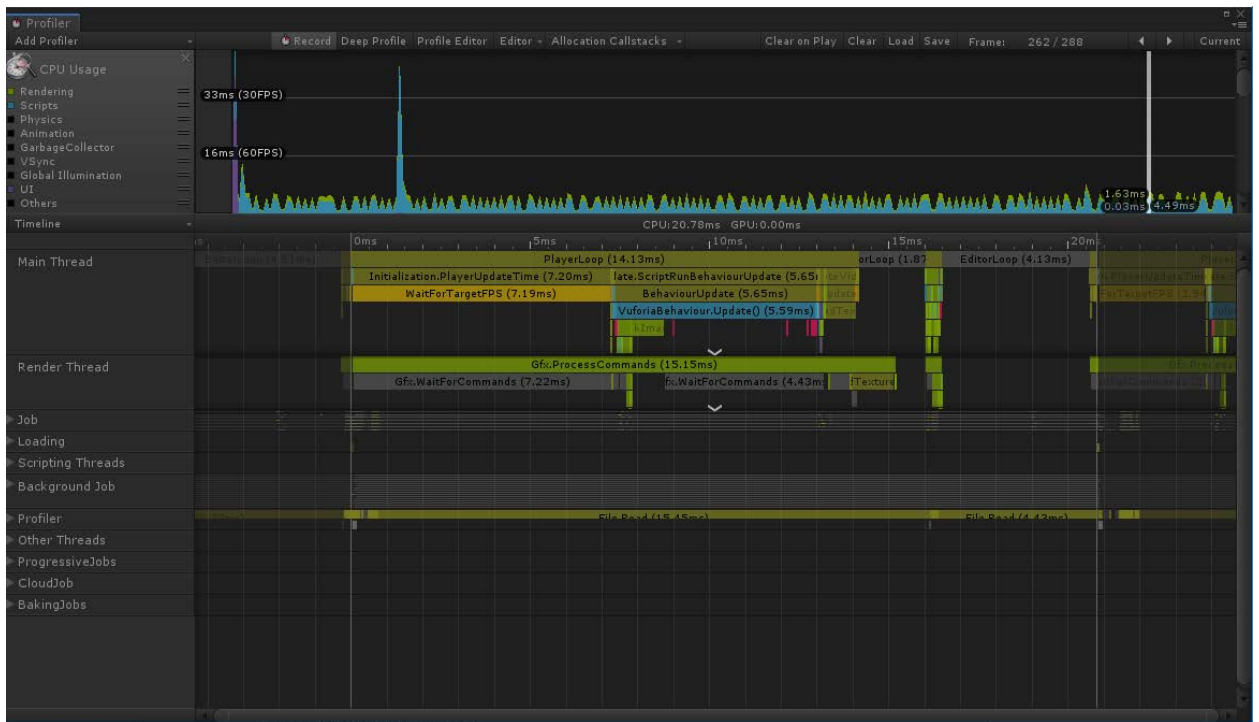


Рисунок 5.4 – Аналіз роботи оперативної пам'яті додатку

Оскільки додаток напряму взаємодіє з відеопотоком та рендерінгом, то потрібно було дотримуватися високої швидкодії роботи із відеопам'ятю. На рисунку 5.5 представлений аналіз навантаження на відеопроцесор.

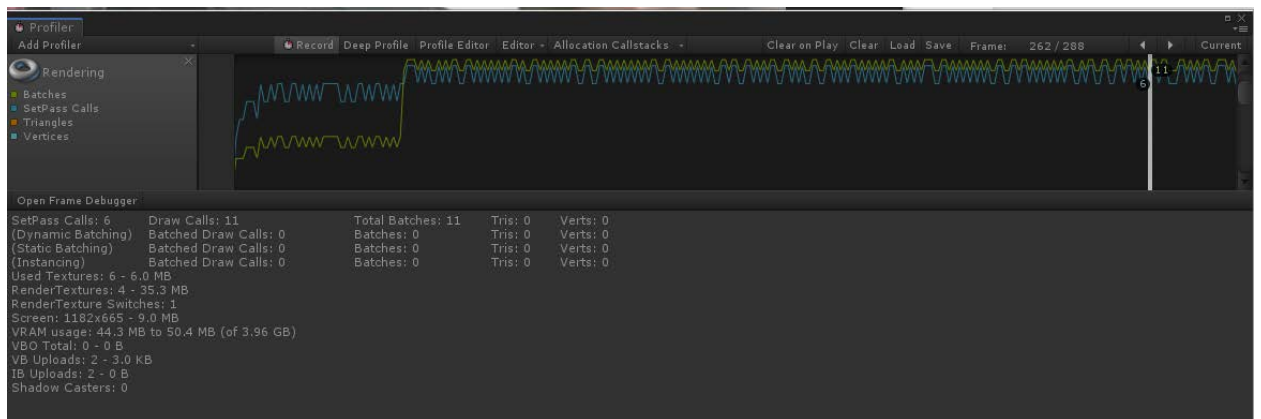


Рисунок 5.5 – Аналіз роботи відеопам'яті додатку

Також, оскільки на мобільних платформах доцільно притримуватись мінімальних затрат фізичної пам'яті, то було необхідно оцінити, чи багато

пам'яті виділяється для стабільної роботи додатку та мінімізувати її. Результат представлений на рисунку 5.6.

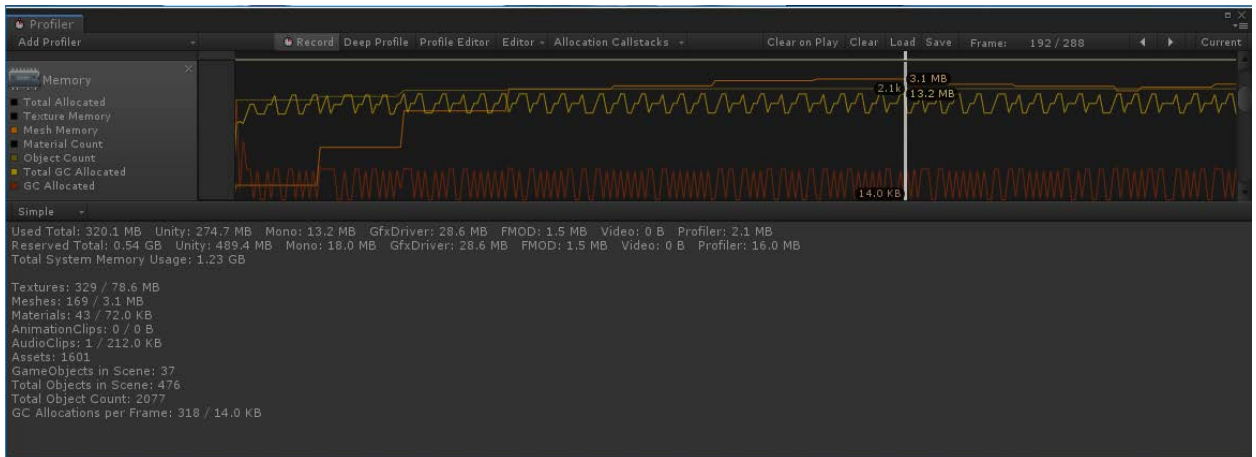


Рисунок 5.6 – Аналіз роботи фізичної пам'яті

ВИСНОВКИ

Під час написання даної атестаційної роботи були розкриті дві досить важливі і великі теми: ігрові рушії і доповнена реальність. В ході розгляду теми про доповнену реальність були розкриті аспекти використання методів доповненої реальності у різних сферах життєдіяльності та, власне, у музейній сфері. У темі ігрові рушії були приведені у приклад п'ять відомих і одночасно різних рушіїв. Вони були порівняні, серед яких найкращим для цієї роботи став Unity, за такими критеріями:

- вартість. Використовуючи Unity у своїх цілях, розробник користується їм безкоштовно;
- мови програмування. C# і JavaScript мають досить розповсюджене використання;
- функціонал. Так як Unity підтримує можливість завантаження плагінів, функціонал цього рушія може бути необмеженим;
- інтерфейс. Інтерфейс Unity є досить інтуїтивним, що полегшує роботу з ним;
- мультиплатформеність. Завдяки Unity можна легко і якісно створювати ігри і додатки на велику кількість відомих платформ.

Іншою ключовою темою стала доповнена реальність. Ця тема була опрацьована з різних боків: були класифіковані системи, які використовують доповнену реальність; були розглянуті і порівняні AR-прилади; оглянуті мобільні системи доповненої реальності, також запропоновані принципи та методи використання доповненої реальності у майбутньому. Доповнена реальність – методика доповнення існуючих візуальних відчуттів віртуальними візуалізаціями за допомогою програмних модулів. У цій роботі були вивчені аспекти доповненої реальності, одним з яскравих прикладів є метод feature detection, були розглянуті основні методи, такі як QCAR і алгоритм FAST, що є його базовою складовою. Було прийнято рішення

реалізувати мобільний додаток за допомогою мови «C#», з використанням модулів «Vuforia», за допомогою графічного ядра Unity.

В результаті розроблений додаток, який дозволяє в відеопотоці ідентифікувати об'єкти і в режимі реального часу отримувати про них додаткову неочевидну інформацію. Доповнена реальність є одним з проривних методів сучасності, що дозволяє користувачеві сприймати більше корисної інформації про навколишній світ і інтерактивно впливати з нею, за допомогою використання різних гаджетів – мобільних пристроїв, лептопів, окулярів віртуальної реальності тощо.

ПЕРЕЛІК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Brian X. Chen. If You're Not Seeing Data, You're Not Seeing. Wired, 2009. 213 p.
2. Azuma R., A Survey of Augmented Reality. Cambridge University Pres, 2005. 254 p.
3. Amit Y. 2D Object Detection and Recognition: Models, Algorithms and Networks. The MIT Press, 2002. 113 p.
4. Bay H., Tuytelaars T., Gool L. Surf: Speed up robust features // European Conference on Computer Vision, 2006. P. 404–417.
- 5 Hartley R., Zisserman A. Multiple view geometry in computer vision. 2nd edition. Cambridge University Press, 2004. 112 p.
6. Lepetit V., Fua P., Pilet J. Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation // Conference on Computer Vision and Pattern Recognition, 2004. P.125–243.
7. Mikolajczyk K., Schmid C. An Affine Invariant Interest Point Point Detector // European Conference on Computer Vision, 2002. P. 128–142.
8. Rosten E. Faster and better: a machine learning approach to corner detection, 2008. 96 p.
9. David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, 2004. 195 p.
10. Ronald T. Azuma A. Survey of Augmented Reality // In Presence: Teleoperators and Virtual Environments. 1997. No 4. P. 355–385.
11. Tikander M. Development and evaluation of augmented reality audio systems: Abstract of dissertation for the degree of Doctor of Science in Technology. Helsinki, 2009. 231 p.
12. Rozier J. Hear&There: An Augmented Reality System of Linked Audio // Online Proceedings of the ICAD, 2009. P 92–123.
13. Karahalios K. GaitAid Virtual Walker for Movement disorder patients Augmented Reality, 2007. 231 p.

14. Benda P., Ulman M. Augmented Reality As a Working Aid for Intellectually Disabled Persons For Work in Horticulture. URL: https://www.researchgate.net/publication/298033889_Augmented_Reality_As_a_Working_Aid_for_Intellectually_Disabled_Persons_For_Work_in_Horticulture (дата доступу – 14.11.2019)
15. Linietsky J. A Small Defense Of GLTF 2.0 On Its Comparison Against OpenGex. URL – <https://godotengine.org/article/small-defense-gltf> (дата доступу – 15.11.2019)
16. Brainville A. Why glTF 2.0 is awesome! URL: <https://dev.to/ybalrid/why-gltf-20-is-awesome-2khp> (дата доступу – 17.11.2019)
17. Understanding World Tracking in ARKit URL: https://developer.apple.com/documentation/arkit/understanding_world_tracking_in_arkit (дата доступу – 28.11.2019)
18. Axon S. How ARKit 2 works, and why Apple is so focused on AR. URL: <https://arstechnica.com/gadgets/2018/06/arkit-2-why-apple-keeps-pushing-ar-and-how-it-works-in-ios-12/> (дата доступу – 29.11.2019)
19. Ronald T. Azuma. A Survey of Augmented Reality. URL: <https://www.cs.unc.edu/~azuma/ARpresence.pdf> (дата доступу – 29.11.2019)
20. Чалая, Л.Э. Поиск неполных дубликатов в системах анализа цифровых изображений [Текст] / Л.Э. Чалая, П.Ю. Попаденко // Вісник КрНУ імені Михайла Остроградського. Вип. 5/2014 (88). – 2014. – С.42 – 47.
21. Берестовий, О.О. Системи когнітивного розвитку з використанням платформи VUFURIA [Текст] // О.О. Берестовий, Д.І. Карпенко, С.Г. Удовенко // Обчислювальний інтелект (результати, проблеми, перспективи) (ComInt-2019): матеріали V Міжнар. науково-практичної конференції, 15-20 квітня. 2019 р. – Ужгород, 2019. – С. 165 – 166.