

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



ВИКОРИСТАННЯ AR В ІНТЕРАКТИВНИХ БАГАТОКОРИСТУВАЦЬКИХ РОЗВАЖАЛЬНИХ ДОДАТКАХ

**Текстова частина
магістерської роботи
за спеціальністю „Інженерія програмного забезпечення” 121**

Керівник магістерської роботи
к.ф.-м.н., доц. Нагірна А.М.

_____ (підпис)

“ ____ ” _____ 2023 р.

Виконав студент Олійник М.В.

“ ____ ” _____ 2023 р.

Київ 2023

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики, к.ф.-м.н.

_____ С. С. Гороховський

(підпис)

„_____” _____ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на магістерську роботу

студенту 2 р.н. магістерської програми Інженерія програмного забезпечення

Олійнику Максиму Вадимовичу

Розробити Розважальний багатокористувацький ігровий додаток з використанням доповненої реальності

Зміст текстової частини до магістерської роботи:

Зміст

Анотація

Вступ

1 Теоретичні відомості

2 Огляд засобів реалізації

3 Розробка прототипу багатокористувацького ігрового додатку з доповненою реальністю

Висновки

Список літератури

Дата видачі „_____” _____ 2022 р.

Керівник

Нагірна А.М., к.ф.-м.н., доц.

(підпис)

Завдання отримав

Олійник М.В.

(підпис)

Тема: використання AR в інтерактивних багатокористувацьких розважальних додатках

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу	01.11.2022	
2.	Огляд технічної літератури за темою роботи	15.11.2022	
3.	Виконання аналізу сучасних рішень	29.11.2022	
4.	Реалізація програмного застосунку з використанням доповненої реальності	17.01.2023	
5.	Інтеграція програмного застосунку з бекенд-сервісним рішенням	14.02.2023	
6.	Написання пояснювальної записки	24.04.2023	
7.	Створення слайдів для доповіді та написання доповіді	27.04.2023	
8.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист магістерської роботи	30.04.2023	
9.	Корегування роботи за результатами попереднього захисту	5.05.2023	
10.	Остаточне оформлення пояснювальної записки та слайдів	10.05.2023	
11.	Захист магістерської роботи (проекту)	13.06.2023	

Студент _____

Керівник _____

“ ”

ЗМІСТ

Анотація	5
ВСТУП.....	6
РОЗДІЛ 1 Теоретичні відомості.....	9
1.1 Що таке доповнена реальність.....	9
1.1.1 Як працює доповнена реальність	10
1.1.2 Сфери використання доповненої реальності	14
1.2 Порівняння доповненої та віртуальної реальності	19
1.3 Доповнена реальність та мобільні пристрої.....	20
1.3.1 ARKit	21
1.3.2 ARCore	23
1.4 Вплив доповненої реальності на багатокористувацькі додатки	25
1.5 Висновки за розділом 1	26
РОЗДІЛ 2 Огляд засобів реалізації	27
2.1 Ігрові рушії з підтримкою доповненої реальності.....	31
2.1.1 Unity.....	32
2.1.2 Unreal Engine	36
2.2 Бекенд-сервіс рішення для ігрових додатків.....	40
2.2.1 GameSparks	41
2.2.2 Photon Engine	43
2.3 Висновки за розділом 2	45
РОЗДІЛ 3 Розробка прототипу багатокористувацького ігрового додатку з доповненою реальністю	46
3.1 Вимоги до прототипу проекту	46
3.2 Технічна реалізація	51
3.3 Демонстрація роботи додатку.....	57
3.4 Висновки за розділом 3	68
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70

Анотація

У магістерській роботі надається огляд технологій доповненої реальності сучасних мобільних платформ. Також проаналізовано ігрових рушіїв, що можуть використовувати дані технології. Розглядаються бекенд-сервісні рішення, що можуть бути використані для створення багатокористувацьких ігрових додатків з доповненою реальністю. В роботі детально описано створення багатокористувацького ігрового додатку з використанням доповненої реальності.

Ключові слова: доповнена реальність (англ. augmented reality або AR), багатокористувацькі додатки, ігрові додатки.

ВСТУП

Актуальність. Індустрія мобільних розважальних додатків вважається однією з найперспективніших галузей розваг та електронної комерції.

Широке поширення смартфонів, що за деякими оцінками становить 5.48 мільярдів пристроїв, що покриває 68.3% населення землі, з яких 5.18 мільярдів, або 64.6% населення землі використовують смартфони разом з доступом до мережі інтернет [1]. Саме завдяки великому відсотку користувачів з доступом до смартфонів, ринок мобільних розважальних додатків є надзвичайно перспективним, а те, що майже всі користувачі смартфонів мають доступ до мережі інтернет дає змогу ефективної монетизації додатків, стимулюючи подальший розвиток галузі. Саме ринок мобільних ігрових додатків займає 50% усього ринку відеоігор [2], та оцінювався в 2021 році у 93.1638\$ мільярдів доларів, а прогнозний ріст галузі у 2021 - 2030 оцінюється в 12.2% в рік. Прогнозний розмір ринку в 2030 році - 261.5863\$ мільярдів [3]. Згідно зі статистикою [31], переважна кількість популярних мобільних ігор, що мають найбільшу кількість гравців в місяць, є саме багатокористувацькі ігри.

Перспективність ринку є вагомим рушієм у розвитку технологій, що потенційно можуть використовуватись у створенні мобільних розважальних додатків, забезпечуючи сталий розвиток індустрії в цілому.

Однією з найбільш перспективних технологій у галузі мобільних ігрових додатків є доповнена реальність (AR). Ця технологія докорінно змінює взаємодію між гравцем та додатком, приносячи в ігровий процес новий рівень інтерактивності, завдяки можливості лаконічного поєднання віртуального світу з реальним.

Поєднання доповненої реальності разом з можливістю взаємодії користувачів через мережу інтернет забезпечує сильніше занурення, порівняно зі звичайними ігровими додатками, стираючи межі між реальністю та ігровим світом, та розширюючи соціальні аспекти ігрового процесу в багатокористувацьких ігрових додатків.

Зважаючи на популярність саме багатокористувацьких мобільних ігор, використання доповненої реальності в мультиплеєр іграх потенційно створює надзвичайно перспективний ринок.

Мета дослідження. Демонстрація можливості використання доповненої реальності в багатокористувацьких мобільних ігрових застосунках.

Завдання дослідження. Зробити огляд та аналіз технологій доповненої реальності на сучасних мобільних платформах. Проаналізувати ігрові рушії, що підтримують даних технологій. Зробити огляд бекенд-сервісних рішень, що можуть бути використані для створення багатокористувацького додатку з підтримкою доповненої реальності.

Об'єкт дослідження. Створення прототипу мобільного багатокористувацького додатку з використанням доповненої реальності (AR).

Предмет дослідження. Використання технологій доповненої реальності для створення багатокористувацьких ігрових додатків.

Джерела дослідження. Друкована література, електронні ресурси (у тому числі офіційні сайти технологій, профільні ресурси, тощо), вихідні коди бібліотек.

Наукова новизна дослідження. Демонстрація можливостей поєднання технологій доповненої реальності з мережевим функціоналом для створення розважальних ігрових додатків.

Робота складається зі вступу, трьох розділів, висновків та списку використаних джерел.

Перший розділ надає загальний огляд доповненої реальності, сфери використання, порівняння з віртуальною реальністю, стан технології на сучасних мобільних платформах, та можливий вплив на багатокористувацькі додатки.

Другий розділ присвячено огляду технологій, що можуть бути використані для створення багатокористувацького ігрового додатку з доповненою реальністю, у тому числі ігрові рушії, та бекенд-сервісні рішення для організації багатокористувацького функціоналу.

Третій розділ описує вимоги, та розробку прототипу багатокористувацького ігрового додатку з доповненою реальністю. Також, надається демонстрація роботи даного прототипу.

РОЗДІЛ 1 Теоретичні відомості

1.1 Що таке доповнена реальність

Доповнена реальність (AR) - це набір технологій, які поєднують контент з віртуального світу з реальним, доповнюючи його в реальному часі. Це покращує сприйняття та взаємодію користувача з додатком, даючи абсолютно новий досвід розширення реального світу.

Технології доповненої реальності використовують комбінацію апаратних і програмних компонентів для інтеграції віртуальних 3D об'єктів у реальний світ, та відстежуючи реальні об'єкти у реальному часі. Доповнена реальність може оперувати кількома видами сенсорної інформації, включаючи візуальну, слухову, тактильну, соматосенсорну, та, навіть, нюхову [4]. Дані, необхідні для інтеграції віртуальної інформації у реальний світ, можуть збиратись за допомогою пристроїв, таких як смартфони, планшети, розумні окуляри, гарнітури, і т.д. Ці пристрої оснащені необхідними сенсорами, камерами, мікрофонами, та дисплеями і динаміками, що можуть бути використані для відображення згенерованого контенту. Пристрої фіксують реальне середовище та накладають на нього віртуальну інформацію, графіку, 3D моделі.

Віртуальні дані можуть як доповнювати реальне середовище, так і маскувати його. Використовуючи штучний інтелект, методи комп'ютерного зору, розпізнавання зображень та інформацію з додаткових сенсорів, доповнена реальність відстежує об'єкти, поверхні або маркери в навколишньому середовищі, органічно взаємодіючи з реальним світом. Лаконічна інтеграція та можливість маніпуляції створює ефект сприйняття віртуального світу як частини реального, а не просто відображення даних, що і є основною цінністю технології.

Піонером доповненої реальності є Іван Сазерленд, та його пристрій під назвою Дамоклів меч [44]. Цей пристрій був розроблений у 1966 році в університеті Солт-Лейк-Сіті, США. Пристрій представляє з себе шолом, що був оснащений лінзами для доповненої реальності. Він давав змогу переглядати зображення накладені на реальний світ. Ідеєю розробки було створити девайс для допомоги пілотам

вертольотів з приземленням вночі. Цей пристрій кріпився до стелі, через свою надмірну вагу, та мав обертовий механізм (рисунок 1.1).

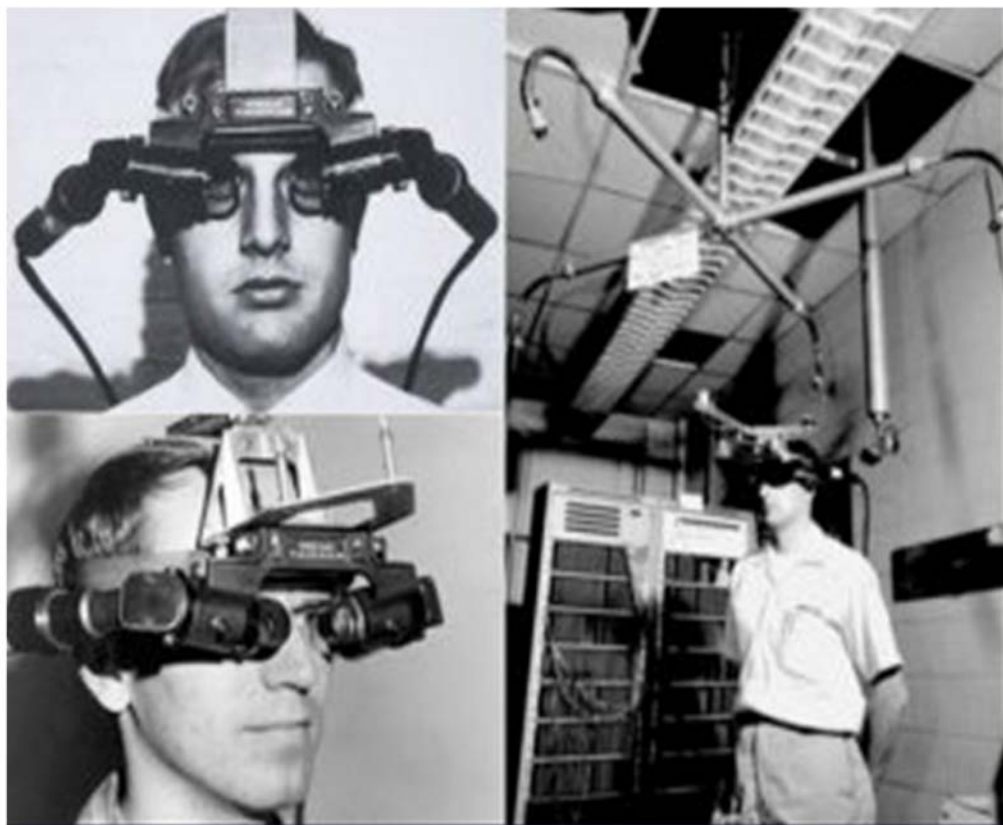


Рисунок 1.1 – Пристрій Дамоклів меч

Термін доповнена реальність був вперше використаний дослідником компанії Boeing, Boeing Томасом Престоном Кауделлом, який розробив пристрій з доповненою реальністю для промислового виробництва, щоб спростити перегляд схем літаків. Цей пристрій представляв з себе дисплей, що кріпився на голові.

1.1.1 Як працює доповнена реальність

Існує декілька технологій доповненої реальності. Доповнена реальність працює за допомогою додаткового обладнання. Прикладами девайсів, що можуть бути використані для доповненої реальності є окуляри доповненої реальності, такі як Microsoft HoloLens. Це оптичний пристрій, що дозволяє бачити наскрізь, проектуючи зображення на прозорі дисплеї, які знаходяться близько до очей.

Також, найпопулярнішими, та найдоступнішими девайсами для відтворення доповненої реальності є звичайні смартфони, у яких є камера, дисплей, та додаткові сенсори.

За принципом роботи можна виділити технології доповненої реальності такі як оптичні окуляри AR, маркерна технологія AR, 3D просторове картографування.

Оптичні окуляри AR

Як випливає з назви, це технологія що базується на оптичних девайсах, таких як окуляри Microsoft HoloLens (рисунок 1.2), або Metavision Meta. Основною відмінністю від мобільних телефонів є те, що ці девайси є оптично прозорими. Ці девайси не відображають відео для відтворення реального світу, а використовують основну особливість - прозорість. Завдяки прозорості, для створення доповненої реальності потрібно відобразити тільки віртуальні об'єкти. Завдяки цьому графіка лаконічно зливається з реальним світом.



Рисунок 1.2 – Microsoft HoloLens

Ці девайси використовують комбіновану лінзу [12], схожу на розгалужувач променю з кутовою поверхнею, яка перенаправляє зображення, що надходить з боку, на око. Оптичний прозорий дисплей змішує світло реального світу з віртуальними об'єктами. Це створює певні обмеження, так темні ділянки можуть загубитись, тому що чорні пікселі прозорі. Це також обмежує використання пристроїв у яскраво освітлених приміщеннях. Також, ці девайси є біноклярними, а тому враховують паралакс і покращують сприйняття світу.

Потрібно звернути увагу на вартість таких девайсів. Так, для прикладу HoloLens від компанії Microsoft коштує починаючи від 3500\$ [32]. Зважаючи на ціну таких

девайсів, їх використання для розробки ігор не є комерційно цікавим, ці девайси більше підходять для професійного використання на виробництвах, тощо.

Маркерна технологія AR

Ця технологія є більш традиційною технологією AR. Камера пристрою знімає кадр відео реального світу. Програмне забезпечення аналізує кадр, шукаючи маркер. У процесі пошуку маркеру відбувається обрахунок розміру, нахилу, щоб визначити відстань і положення, орієнтації відносно камери в тривимірному просторі. Ці дані використовуються для розміщення, візуалізації графіки. У результаті графіка об'єднується з відео та відображається користувачу (рисунок 1.3).

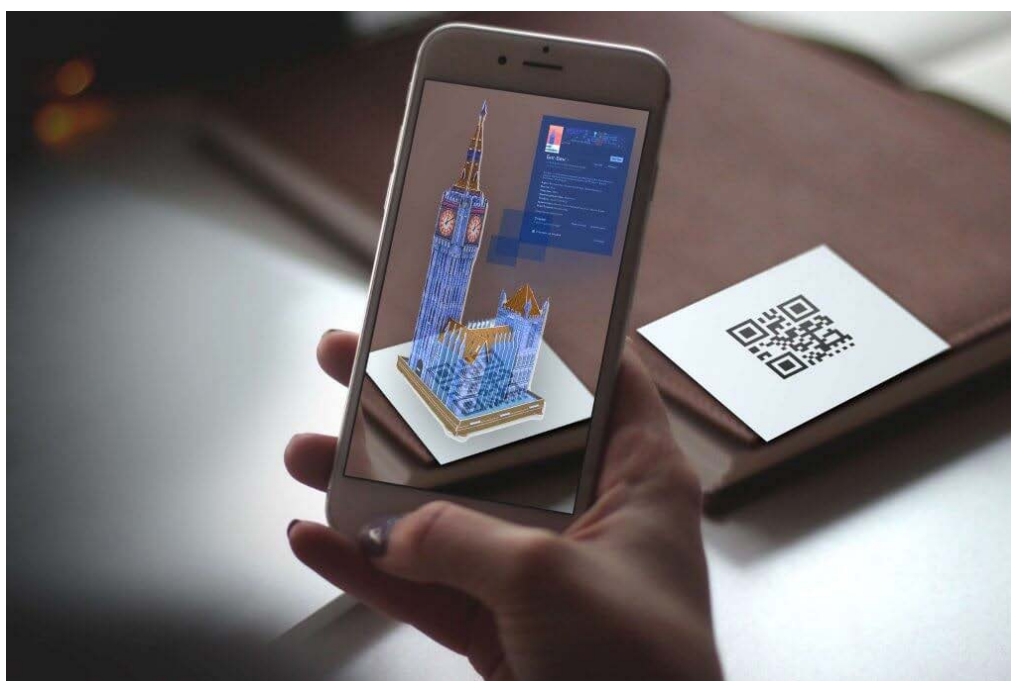


Рисунок 1.3 – Приклад маркерного AR

Ця технологія зазвичай працює на базі смартфонів, що є більш доступним, ніж спеціальні оптичні пристрої. Для плавної роботи на телефонах, ця технологія є добре оптимізованою, та враховує переміщення девайсу, замість обрахунку з нуля у кожному кадрі.

У ролі цілей для даної технології можуть виступати маркери, закодовані маркери та зображення.

Маркери. Це найпростіша ціль. Маркери з широкою рамкою легко розпізнаються програмним забезпеченням. При цьому витрати обчислювальних

потужностей є досить невеликими. Також, цей тип цілей мінімізує ризики некоректної роботи через погане освітлення (рисунок 1.4).



Рисунок 1.4 – Приклад маркеру з широкою рамкою

Закодовані маркери. Більш просунутим варіантом простих маркерів є закодовані маркери, що можуть відповідати за різні дані. Змінюючи маркер будуть відображатись різні дані [13] (рисунок 1.5).



Рисунок 1.5 – Приклад закодованих маркерів

Зображення. Новим рівнем інтерактивності стала можливість розпізнавання зображень. Це дало змогу відстежувати будь які природні об'єкти без необхідності розповсюджувати власні маркери [14]. Зображення є гарною ціллю для відстежування якщо воно має чітко виражені межі, нерегулярні патерни, хорошу контрастність. У випадку використання зображень у якості цілей, будується та зберігається карта особливостей. Ця карта співставляється з оригіналами, для пошуку співпадінь.

3D просторове картографування

Цей тип технологій базується на скануванні навколишнього середовища та побудові просторової 3D сітки. Ця технологія використовує датчики глибини. Завдяки цьому можна реєструвати об'єкти та поверхні без необхідності використання спеціальних маркерів, або бази з цільовими зображеннями для відстеження (рисунок 1.6).



Рисунок 1.6 – Приклад 3D просторової карти

Датчик глибини вимірює відстань до твердих поверхонь за допомогою інфрачервоної камери, та проєктора. Проєктор проєктує інфрачервоні точки в навколишнє середовище, які потім зчитуються інфрачервоною камерою і аналізуються програмно-апаратним забезпеченням. Глибина обчислюється за зміщенням точок. Аналіз відбувається за допомогою декількох кадрів, для збільшення точності. У доповнення до датчика глибини може використовуватись звичайна камера видимого світла. Використовуючи методи фотограмметрії, видимі об'єкти ідентифікуються як набір точок. 3D позиція кожного вузла обчислюється за допомогою тріангуляції. Таким чином, ця технологія дає змогу отримати якісне зображення доповненої реальності.

1.1.2 Сфери використання доповненої реальності

Доповнена реальність може використовуватись у широкому спектрі сфер реального життя, у тому числі освіти, охорони здоров'я, реклами, архітектури і декору, промисловості, та, звісно, розважальних ігрових додатках.

Освіта. Згідно з мультимедійно-когнітивною теорією, люди навчаються краще, якщо інформація подається за допомогою декількох органів відчуття одночасно. У контексті цієї теорії, технології доповненої реальності мають величезний потенціал, покращуючи сприйняття навчальної інформації [5], та потенційно збільшуючи

зацікавленість студентів за допомогою інтерактивної подачі [43]. Інтерактивність подачі інформації, що може бути представлена у ігровій формі, також може бути досить ефективним методом навчання для дітей. Це створює нові можливості у сфері освіти. Цікавим прикладом використання доповненої реальності у навчанні є додаток SkyView [33] (рисунок 1.7).



Рисунок 1.7 – Додаток SkyView

Основною ідеєю цього додатку є надати користувачам легкий для застосування інструмент для спостереження за зоряним небом. Цей додаток використовує доповнену реальність для пошуку зірок, планет, та туманностей на зоряному небі.

Медицина. Доповнена реальність також є ефективним засобом для відображення медичної інформації. Так, для прикладу, ця технологія може бути використана для відображення 3D-US зображень, сегментованих та рендерингових зображень на основі СТ у реальному часі, що неможливо візуалізувати без додаткових

засобів, що зазвичай є досить дорогими [6]. Використання додатків, що працюють з доповненою реальністю, зменшує вірогідність людських помилок, та здешевлює необхідне обладнання для візуалізації медичної інформації, при цьому збільшуючи розуміння інформації (рисунок 1.8).

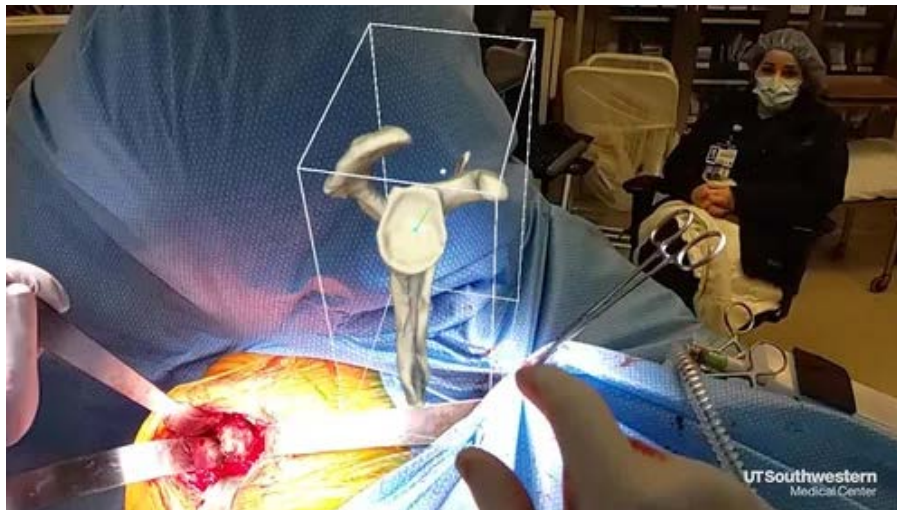


Рисунок 1.8 – AR зображення медичного плану пацієнту, що транслюється на оптичну гарнітуру вряча під час операції [15]

Маркетинг. За допомогою доповненої реальності можливо досить ефективно привертати увагу людей, що є безумовно основною задачею реклами. Гарним прикладом використання доповненої реальності у рекламі є реклама Pepsi у 2014 році. Компанія встановила екран на стіні автобусної зупинки в Лондоні. Цей екран відігравав роль фальшивого вікна, на якому проектувати зображення з доповненою реальністю, що зображало ілюзію атаки літаючими тарілками, роботами, та тигром [7]. Ця рекламна акція була безумовно успішною, загальний охоп аудиторії що переглянули новини про цю рекламну кампанію становить 385 мільйони, а відео з цією стіною набрало 3 мільйони переглядів за 5 днів [8] (рисунок 1.9).



Рисунок 1.9 – Рекламна кампанія Pepsi

Архітектура та дизайн. Ще одним прикладом поєднання рекламної кампанії та розробки у сфері архітектури та дизайну є додаток від компанії ІКЕА з використанням доповненої реальності [9]. Цей додаток дає можливість користувачам тестувати продукти ІКЕА в реальному часі, за допомогою доповненої реальності. Додаток пропонує функціонал з розміщення деталізованих 3D моделей продуктів компанії в інтер'єрі житла користувача, представляючи клієнтам новий рівень сервісу. Додаток автоматично масштабує 3D моделі продуктів компанії у простір реального світу, забезпечуючи точність розмірів 98% (рисунок 1.10).

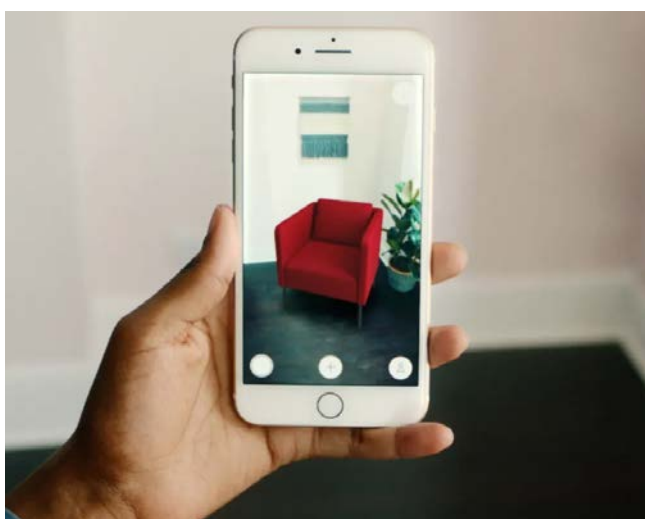


Рисунок 1.10 – Мобільний додаток ІКЕА

Промисловість. Прикладом використання доповненої реальності у промисловості є компанія Bosch. Ця компанія використовує доповнену реальність у автомайстернях, для підвищення ефективності виконання завдань з сервісного

ремонту. Компанія використовує навчальні додатки з доповненою реальністю для технічного персоналу майстерні. Завдяки цим додаткам, компанія навчає персонал, надаючи дані про деталі автомобілів, їх розміщення, вказівки по роботі з цими деталями, тощо. Завдяки цим програмам вдалось зменшити кількість помилок на 15% (рисунок 1.11).

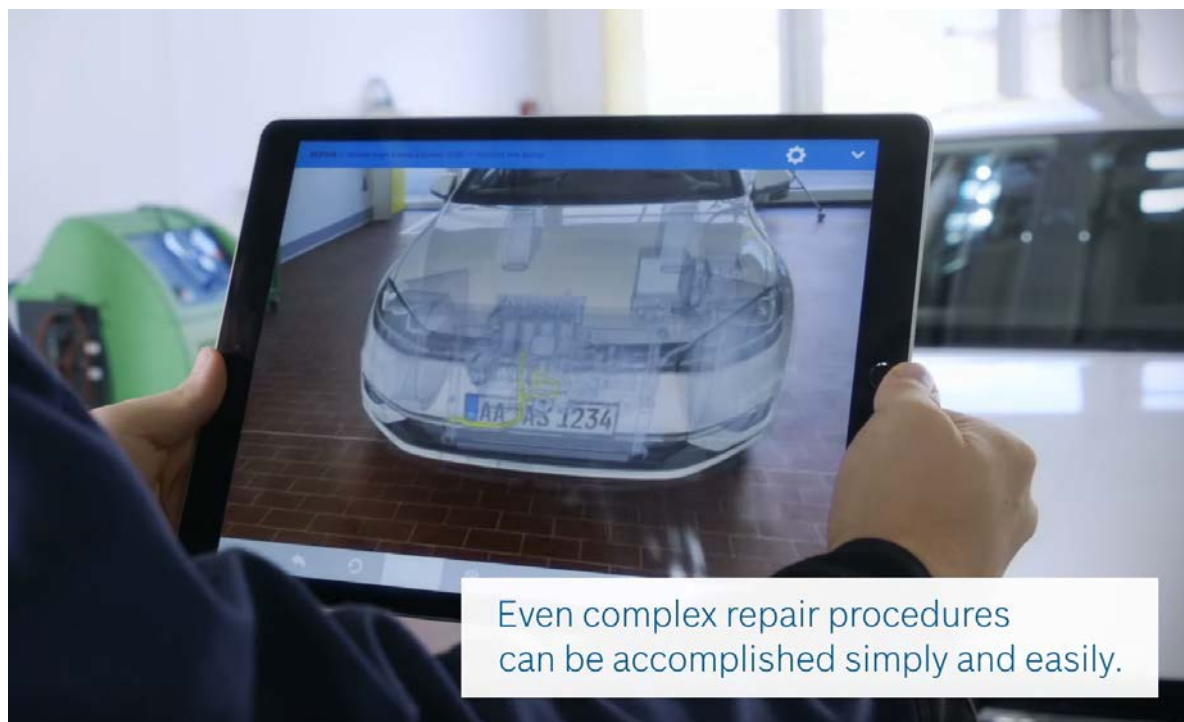


Рисунок 1.11 – Використання доповненої реальності компанією Bosch

Ігри. Прикладом надзвичайно вдалого використання доповненої реальності в мобільних ігрових застосунках є гра Pokemon Go, що розроблена компанією Niantic спільно з The Pokemon Company. Гра поєднує франшизу Pokemon з можливістю дослідження реального світу, використовуючи технології доповненої реальності. У грі гравці беруть на себе роль тренера покемонів, ловлячи та тренеруючи різноманітних покемонів в різних публічних місцях, таких як парки, торгові центри. За допомогою технології доповненої реальності гравці можуть бачити покемонів, що накладаються на реальне оточення, на екранах своїх телефонів (рисунок 1.12).



Рисунок 1.12 – Мобільна гра Pokemon Go

Гра стимулює соціальну взаємодію, та спонукає гравців на фізичну активність, досліджуючи околиці. Гра є надзвичайно успішною, загальна кількість скачувань гри становить більше 500 мільярдів разів, а оборот від гри оцінюється в 1 мільярд доларів на рік [10].

1.2 Порівняння доповненої та віртуальної реальності

Доповнена реальність (AR) та віртуальна реальність (VR) є двома різними технологіями, що мають досить подібні принципи роботи. Ці дві технології мають за мету дати користувачам абсолютно новий досвід у взаємодії з додатками. Віртуальна і доповнена реальності дають можливість користувачам взаємодіяти з віртуальними об'єктами, що знаходяться в просторі реального світу, хоча і у різний спосіб. Доповнена реальність інтегрує віртуальні об'єкти беручи до уваги дані про реальний світ, що збираються за допомогою камери та додаткових сенсорів, змішуючи реальний та віртуальні світи, покращуючи сприйняття реального світу [11]. Віртуальна реальність майже повністю базується на даних про віртуальний простір, ігноруючи всі дані реального світу, окрім просторового положення гравця, що збираються за допомогою акселерометру, та гіроскопу, створюючи абсолютно новий світ для користувача [42]. Саме факт інтеграції у реальний світ збільшує кількість можливих сфер використання доповненої реальності у порівнянні з віртуальною.

Також, важливою відмінністю між технологіями є вимоги до обладнання. Віртуальна реальність потребує спеціальної гарнітури з дисплеями, та якісними

лінзами для передачі зображення користувачу, та компютера для запуску додатків. Така гарнітура має обмежену віртуальною реальністю сферу використання, та коштує як середньо бюджетний смартфон. У той же час для роботи доповненої реальності достатньо одного тільки смартфона, який має всі необхідні для роботи AR сенсори, та дисплей.

Окрім цього у деяких випадках ці дві технології можуть поєднуватись. Гарним прикладом можливого поєднання віртуальної реальності з доповненою реальністю є їх застосування в додатках пов'язаних з архітектурою. Віртуальна реальність може бути використана для відображення симуляції інтер'єру майбутнього будинку на етапі планування. У той же час доповнена реальність може бути використана для відображення зовнішнього вигляду майбутнього будинку, беручи до уваги ландшафт. Також, можливо відобразити комунікації будинку, спроектовані на реальний будинок.

Різниця між технологіями показує, що доповнена реальність є більш перспективною з комерційної точки зору, так як не потребує ніякого специфічного обладнання, що робить її більш доступною для звичайного користувача, збільшуючи кількість потенційних користувачів додатків, та має більш широкий спектр можливого застосування у порівнянні з віртуальною реальністю. Також це робить доповнену реальність більш перспективною з точки зору маркетингу, та монетизації.

1.3 Доповнена реальність та мобільні пристрої

Найбільш цікавим є використання доповненої реальності в додатках для смартфонів. Смартфони мають необхідні датчики, для знаходження просторового положення девайсу у реальному світі, камеру для збору відео з реального світу, та екран, для відображення результуючого контенту.

Найпоширенішими застосуваннями доповненої реальності у додатках для смартфонів є фільтри, різного роду маски для лица, що можуть використовувати нейромережі та спеціальні алгоритми для створення різних ефектів, відтворення тексту, графіки.

Ще одним популярним застосуванням доповненої реальності для смартфонів є ігри, в яких ігрові об'єкти можуть бути розміщені в реальному просторі. Даючи можливість для взаємодії з цими об'єктами, додаток створює ефект присутності.

Доповнена реальність для смартфонів реалізується за допомогою двох фреймворків, ARKit у випадку смартфонів iPhone від компанії Apple, та фреймворк ARCore, що дає доступ до можливостей доповненої реальності для розробки додатків для смартфонів на базі операційної системи від компанії Google, Android.

1.3.1 ARKit

ARKit це набір для розробки додатків доповненої реальності. ARKit створений компанією Apple, та представлений на конференції для розробників WWDC у червні 2017. Фреймворк був частиною великого оновлення iOS 11. Цей набір для розробки створений спеціально для телефонів(iPhone), та планшетів(iPad) на базі операційної системи iOS, використовуючи особливості телефонів, та планшетів від компанії Apple, включаючи наявність сенсору глибини, для оптимізації роботи фреймворку. ARKit підтримують усі девайси з iOS 11, або вище.

Цей фреймворк використовує ряд сенсорів, та камеру на пристроях компанії Apple для створення точної тривимірної карти оточення реального простору.

За допомогою ARKit розробники можуть створювати різноманітні додатки з використанням доповненої реальності. Компанія Apple активно підтримує та покращує цей фреймворк.

Основними функціями ARKit є:

- відслідковування. Цей фреймворк дає можливість відслідковувати просторову позицію смартфона в реальному часі, в реальному просторі. Для реалізації цієї функції, фреймворк комбінує зображення з камери, дані з сенсору руху, та одометру. За допомогою цієї функції можливо модифікувати віртуальний простір так, щоб він ефективно зливався з реальним світом;
- розуміння середовища. Фреймворк надає функціонал розпізнавання горизонтальних та вертикальних площин реального світу. Це дає можливість реалістично інтегрувати віртуальну реальність в реальний світ у реальному

часі, використовуючи розпізнані площини як контрольні позиція для розміщення віртуальних об'єктів;

- розпізнавання статичних 2D зображень. За допомогою цієї функції можливо розпізнавати зображення, що співпадають з деякими шаблонними зображеннями, що надаються додатком для роботи фреймворку. Це дає можливість надавати якусь візуальну інформацію поверх розпізнаних зображень;
- оцінка світла. Ця функція допомагає гармонічно інтегрувати об'єкти віртуального світу у реальний, з урахуванням світла реального світу. Це дає можливість зменшити візуальні границі між віртуальним та реальним світом, покращити якість відображення об'єктів.

ARKit активно розвивається, нові версії фреймворку приносять нові потужні можливості для розробки додатків з доповненою реальністю. Функціонал, що з'явився в ARKit 2 є [16]:

- виявлення статичних 3D об'єктів. Ця функція дає змогу розпізнавати 3D об'єкти, що співпадають з з деяким 3D шаблоном;
- відслідковування пересування зображень. Ця функція дозволяє накладеній на зображення візуальній інформації слідкувати за переміщеннями зображення якоря;
- відслідковування лиця. Використовуючи камеру TrueDepth, що є особливістю телефонів компанії Apple, можливо відслідковувати емоції та рухи обличчя користувача;
- зберігання карти навколишнього середовища. Ця функція дає можливість зробити карту навколишнього середовища при першому запуску додатку. Цю карту можливо використовувати при наступних запусках, оптимізуючи процеси виявлення та відстеження додатку;
- багатокористувацький режим. Використовуючи цю функцію, можливо передати дані про розміщення віртуальних об'єктів на площині через мережу іншим користувачам. Це дає змогу побудови якісної багатокористувацької сесії.

Версія ARKit 3 привнесла такі нові функції [17]:

- відслідковування тіл і перекриття. Ця функція дає можливість відслідковувати позиції та переміщення тіл людей та реалістично розміщати 3D об'єкти перед або позаду людей;
- захоплення рухів. Використовуючи цю функцію можливо побудувати 3D скелет людини, записувати рухи людей;
- відстежування декількох облич. Тепер можливо відстежувати 3 або більше облич одночасно;
- використання передньої та задньої камери одночасно. Це дає можливість швидшої загрузки зображення, покращує якість інтеграції просторової віртуальної інформації.

Версія ARKit 4 додала наступні функції [18]:

- привязка до локації. Ця функція дає можливість прив'язувати 3D інформацію до позиції в реальному світі, використовуючи GPS;
- оцінка глибини. Використовуючи LiDAR, ARKit надає можливість оцінювати відстань від телефону до об'єктів.

Версія ARKit 5 покращила вже існуючий функціонал фреймворку [19]. У тому числі, покращила якість привязки до локації, покращила відслідковування лиця, захоплення рухів.

1.3.2 ARCore

ARCore це фреймворк, що заснований на Tango SDK. Tango SDK це розробка компанії Google, для створення AR додатків для телефонів на базі операційної системи Android. Tango SDK був випущений у 2014 році, для його роботи був потрібен телефон з датчиком глибини. Особливістю цього фреймворку було те, що він надавав можливість картографувати середовище у 2D, та 3D. Цей проект був реалізований у досить малій кількості девайсів.

ARCore представлений у березні 2018 року компанією Google як новий фреймворк для розробки додатків з доповненою реальністю для телефонів на базі операційної системи Android. Основною відмінністю було те, що для роботи нового

фреймворку не потрібна наявність спеціалізованих датчиків, чи камер, що дало змогу значно ширше розповсюдити функціонал доповненої реальності на телефони під управлінням операційної системи Android. Мінімальна версія операційної системи повинна бути не менше Android Nougat (7.0).

Основними функціями фреймворку ARCore є:

- відслідковування. Цей фреймворк використовує звичайну камеру телефону та дані з інерційного датчика, що визначає орієнтацію і позицію девайсу під час руху для побудови якірних точок реального світу. Це дає можливість правильно розташовувати віртуальні об'єкти у реальному просторі;
- розуміння середовища. ARCore використовує різноманітні сенсори смартфона щоб знаходити горизонтальні та вертикальні площини. Окрім цього фреймворк може відслідковувати рухи;
- оцінка яскравості. Ця функція дозволяє смартфону оцінювати навколишнє освітлення щоб ідеально інтегрувати 3D елементи в реальний світ.

Також варто відмітити нововведення, що з'явилися з новими версіями ARCore [20].

Версія 1.0. З'явилось відстежування горизонтальних площин, відслідковування переміщення телефону в просторі.

Версія 1.2. Додалась можливість розпізнавати вертикальні площини. Також, з'явився функціонал розпізнавання зображень.

Версія 1.7. Новий функціонал з розпізнавання та доповнення облич за допомогою фронтальної камери смартфона.

Версія 1.9. Додалась можливість відслідковування рухомих зображень, навіть якщо вони пропали з видимості камери. Фреймворк може відслідковувати до 20 зображень.

Версія 1.10. Функціонал з оцінки освітленості середовища.

Версія 1.11. Покращено продуктивність фреймворку для підтримки обробки 60 кадрів в секунду.

Версія 1.18.0. Додана можливість оцінки глибини зображення.

Версія 1.21.0. Додана підтримка стереокамер.

Версія 1.31.0. Додано API для роботи з даними з Google Earth 3D, Street View, Google Maps, даючи можливість створення доповненої реальності на базі місцезнаходження в реальному світі.

1.4 Вплив доповненої реальності на багатокористувацькі додатки

Соціалізація в багатокористувацьких ігрових додатках відіграє важливу роль. Багатокористувацькі ігри покладають на себе мету надати гравцям простір для гри, та взаємодії один з одним у віртуальному світі, спонукаючи їх до цього різними методами, ставлячи певні завдання, що можуть бути виконані тільки разом, стимулюючи до кооперації. Багатокористувацькі ігри надають можливості для комунікації між гравцями, надаючи доступ до різного роду чатів, як текстових, аудіо, та можливість показувати емоції, що дозволяє гравцям спілкуватись між собою, обговорювати гру, обмінюватись досвідом, планувати стратегію. Також зазвичай присутній такий соціальний функціонал, як системи що мають можливість додавати в друзі, формувати групи, запрошувати до кланів, тощо. Усе це дає можливість створення соціальних зв'язків, незалежно від фактичного місця знаходження гравців.

Ігри, створені з урахуванням можливостей доповненої реальності можуть покращити соціальну взаємодію гравців у багатокористувацьких іграх. Візуальна присутність персонажів у реальному просторі дозволяє гравцям відчути ніби аватари інших гравців знаходяться у реальному просторі, поруч з реальними гравцями. Гравці можуть бачити віртуальних персонажів у реальному часі, спостерігаючи за їхніми переміщеннями, діями, рухами, покращуючи враження від спільного ігрового процесу. Спільні ігрові процеси, можливості для взаємодії між гравцями сприяє покращенню взаємовідносин. Немаловажливу роль відіграє й надання можливості персоналізації аватарів гравців. Відчуття унікальності аватарів дає гравцям можливість розрізняти гравців, персоналізуючи їх, сприяючи покращенню комунікації, творчому процесу між гравцями.

1.5 Висновки за розділом 1

У даному розділі було зроблено огляд технологій доповненої реальності, принципів роботи цих технологій. Було порівняно доповнену реальність з віртуальною реальністю. З цього порівняння можливо зробити висновок, що доповнена реальність є цікавішою з точки зору комерційного застосування, так як не потребує додаткового обладнання зі сторони кінцевого користувача. Також, було розглянуто технології доповненої реальності, що використовуються на сучасних мобільних девайсах. Було досліджено можливий вплив доповненої реальності на багатокористувацькі ігрові додатки.

РОЗДІЛ 2 Огляд засобів реалізації

Ігрові додатки є технологічно складною задачею, що вирішується за допомогою великої кількості різноманітних технологій. Основою будь якої гри є ігровий рушій. Ігрові рушії є повноцінними пакетами розробки, що надають набір візуальних інструментів для розробки. Як правило, у ігрового рушія є спеціальне інтегроване середовище розробки, що має на меті спростити та пришвидшити розробку гри. Більшість ігрових рушіїв надає набір інструментів для полегшення розробки графічної, звукової, фізичної, та інших систем гри. Надаючи великий набір можливостей, ігровий рушій є повноцінною програмною платформою, що допомагає скоротити фінансові та часові витрати на розробку ігрових додатків для компаній, ефективно реалізуючи, та постійно удосконалюючи процес розробки.

Як правило, ігрові рушії забезпечують можливість розробки під різні платформи, майже відокремлюючи розробника від специфіки розробки під конкретну платформу. Як правило, використовуючи ігровий рушій розробники витрачають мінімальні зусилля для адаптації під різні платформи.

Попри основну задачу, а саме написання ігор, ігрові рушії можуть бути використані і для інших задач, включаючи професійні середовища для симуляції певних процесів, маркетингові демонстрації, навчальні додатки, платформи для моделювання, тощо.

Окрім цього, рушії зазвичай розробляються з урахуванням можливостей для розширення, що потенційно допомагає розробникам боротися з устаріванням технологій, та розширювати недостатній функціонал. Без цієї можливості адаптація до нових технологій потенційно може призвести до необхідності переходу на інший ігровий рушій, що створює небажані затримки.

Крім функціоналу, ігрові рушії відрізняються і моделями монетизації, та ліцензування. Це може бути як повністю безкоштовні ігрові рушії з відкритим кодом, як наприклад Godot, ігрові рушії з відкритим кодом але моделлю фінансування через виплату певного відсотка від доходу, як у Unreal Engine, так і закриті ігрові рушії, з необхідністю одноразової оплати за користування, як Unity.

Окрім ігрового рушія, для розробки багатокористувацького додатку потрібно налагодити мережеву взаємодію між гравцями. Мережева взаємодія неможлива без використання додаткової інфраструктури для створення та забезпечення роботи багатокористувацької гри.

Є 2 основні моделі для побудови міжмережевої взаємодії для гри, клієнт-сервер, та P2P (Peer-to-Peer). Кожна з моделей має свої переваги та недоліки. Вибір моделі залежить від конкретних вимог та характеристик гри.

Клієнт-сервер. Ця архітектура полягає у розділенні гри на дві складові сторони: клієнт, та сервер.

Клієнтська сторона (Client) представляє собою користувацький ігровий додаток, що виконується на пристрої гравця. Пристроєм може бути персональний комп'ютер, ігрова консоль, мобільний телефон. Клієнт відповідає за відображення графіки, програвання звуку, забезпечує можливості для взаємодії гравця з грою за допомогою інтерфейсів вводу, таких як клавіатура, мишка, або сенсорний дисплей, тощо.

Серверна сторона (Server) представляє собою потужний комп'ютер (сервер), що відповідає за управління ігровою сесією. Це включає обробку ігрових подій, обрахунок фізики, управління даними. Серверна сторона забезпечує зв'язок з клієнтськими сторонами, забезпечує синхронізацію між гравцями. Також, серверна сторона відповідає за забезпечення безпеки та справедливості між усіма гравцями, що беруть участь у ігровій сесії.

Перевагами клієнт-серверної архітектури є:

- безпека. Зважаючи на те, що основна логіка гри обраховується на сервері, а клієнт відповідає тільки за відображення та ввід даних для контролю, клієнт-серверна архітектура збільшує безпеку, ускладнюючи взлом гри. Це дає можливість забезпечити добросовісну ігрову сесію без шахрайства;
- стабільність. Завдяки тому, що зазвичай сервер має високу продуктивність, та гарний зв'язок з мережею інтернет, використання клієнт-серверної архітектури забезпечує стабільну ігрову сесію з порівняно мешними шансами проблем під час гри;

- менший об'єм трафіка. Основною перевагою є те, що клієнт обмінюється пакетами тільки з сервером, без необхідності обмінюватись пакетами з іншими клієнтами. Це дає можливість зменшити кількість пакетів, що отримує клієнт, що дозволяє економити трафік клієнта, зменшуючи ціну за користування зв'язком для гравця;
- можливості для масштабування. Простота масштабування дозволяє відносно легко масштабувати інфраструктуру ігрового додатку. Збільшуючи кількість серверів, можливо легко збільшити обсяг гравців, що можуть одночасно грати в гру. Також, існують спеціальні інструменти для моніторингу та керування ігровими серверами, забезпечуючи ефективне використання інфраструктурних ресурсів.

Недоліками клієнт-серверної архітектури є:

- залежність від серверу. Клієнт серверна архітектура робить гру залежною від доступності та надійності серверів. Якщо кількість серверів не буде достатньою, користувачі можуть перебувати значний період часу у черзі на гру;
- більша завантаженість серверу. Зважаючи на те, що клієнт-серверна архітектура передбачає основні обрахунки на серверній стороні, це створює більші вимоги щодо потужності серверів;
- більші витрати на інфраструктуру. Зважаючи на те, що всі обрахунки ведуться на серверах, та всі ігрові сесії повинні контролюватись на стороні серверу, масштабування гри потребує значних фінансових витрат.

Ця архітектура є надзвичайно популярною. Прикладами ігор, що використовують цю архітектуру є надзвичайно популярна гра Minecraft, та змагальні ігри, такі як Counter-Strike: Global Offensive, League of Legends, тощо. Основною перевагою використання цієї архітектури у цих іграх є захист від шахрайництва.

P2P (Peer-to-Peer). Ця архітектура є альтернативною для створення багатокористувацьких ігор. У Peer-to-Peer архітектурі усі гравці виступають як рівні між собою, та спілкуються без посередництва серверу. У цій архітектурі кожен гравець одночасно виступає як клієнт, так і сервер, що означає що кожен гравець

може отримувати дані безпосередньо від інших гравців, утворюючи деяку павутину зв'язків.

Перевагами P2P (Peer-to-Peer) є:

- відсутність централізованого сервера. Архітектура P2P (Peer-to-Peer) передбачає передачу даних безпосередньо від одного гравця до іншого, та обрахунок усіх подій на рівні клієнтів. Це зменшує залежність гри від інфраструктури;
- легке масштабування. Зважаючи на те, що усі обрахунки подій відбуваються на стороні клієнтів, а дані передаються від клієнта до іншого клієнта, без додаткового серверу, масштабування потребує мінімальних змін, збільшуючи потужність тільки серверу, що відповідає за з'єднання гравців;
- дешевше обслуговування інфраструктури. Зважаючи на відсутність серверного обрахунку подій, тобто відсутності централізованого серверу, ціна за масштабування є мінімальною;
- менша затримка. Так як дані передаються безпосередньо від одного гравця, до іншого, без додаткового етапу у вигляді серверу, затримка передачі даних є мінімальною.

Недоліками P2P (Peer-to-Peer) є:

- вразливість до шахрайства. Основним недоліком P2P (Peer-to-Peer) архітектури є те, що ця архітектура не передбачає сервер, що контролює ігрові події, та фізику. Це дає контроль над власними даними на стороні кожного з клієнтів, що дає більші можливості для шахрайства. У деяких іграх цю проблему можливо частково подолати використовуючи гібридну архітектуру, у якій буде наявний сервер перевірки чесності подій;
- проблеми з стабільністю. У архітектурі P2P (Peer-to-Peer) вся мережа гравців залежить один від одного. Якщо один гравець виходить з гри, або має погану якість з'єднання, це може негативно вплинути на ігрову сесію для всіх гравців;
- більший об'єм трафіку. Так як всі гравці можуть отримувати дані від інших гравців, це може призвести до того, що кожен з гравців буде отримувати більшу кількість пакетів, витрачаючи на це трафік, збільшуючи ціну з'єднання.

Прикладами використання Peer-to-Peer архітектури у популярних іграх є такі проекти як: Animal Crossing: New Horizons, GTA Online.

Архітектури P2P (Peer-to-Peer), та Client-Server мають свої переваги та недоліки. Використання тієї чи іншою архітектури залежить від різних факторів. Клієнт серверна архітектура є більш стабільною, з меншою кількістю можливостей для нечесної гри, за рахунок наявності серверу. Це робить цю архітектуру ідеальною для кіберспортивних ігор, MMORPG, які є вимогливими до відсутності шахрайства. У той же час P2P (Peer-to-Peer) є ідеальною для казуальних, невеликих ігор, за рахунок вартості масштабування. Загалом, для вибору підходящої архітектури мережевого зв'язку для ігор потрібно враховувати наступні фактори:

- допустима затримка в грі;
- максимальна кількість гравців у ігровій сесії;
- масштаб синхронізації;
- вартість створення, підтримки, та експлуатації;
- важливість захисту від шахрайства.

2.1 Ігрові рушії з підтримкою доповненої реальності

Ігрові рушії, що надають можливість розробки додатків з доповненою реальністю зазвичай використовують ARCore для телефонів та планшетів на базі операційної системи Android, ARKit для телефонів та планшетів на базі операційної системи iOS, створюючи платформи-незалежну абстракцію для розробників. Найпопулярнішими ігровими рушіями з підтримкою доповненої реальності є Unity, та Unreal Engine. Хоча доступні і інші рішення, але вони мають значно меншу спільноту, що підтверджується статистикою з використання ігрових рушіїв у реальних проектах [37]. Розмір спільноти є досить вагомим фактором, так як саме спільнота забезпечує доступність інформації про розробку з урахуванням технології. Тому розглянемо саме Unity, та Unreal Engine.

2.1.1 Unity

Unity, також відомий як Unity3D це багатофункціональний ігровий рушій розроблений компанією Unity Technologies, який також має інтегроване середовище розробки. Цей ігровий рушій зазвичай використовується для створення відеоігор, але, також, знайшов широке застосування у інших сферах, таких як кінематографія, автомобілі, архітектура та дизайн, інженерія, та навіть у сфері оборони [22].

Станом на 2021 рік програмне забезпечення створене за допомогою Unity скачувалось 5 мільярдів разів кожного місяця, маючи при цьому 2.8 мільярдів активних користувачів в місяць [39]. За деякими даними Unity займає 38% ринку ігрових застосунків [40].

Перша версія ігрового рушія Unity була випущена 6 червня 2005. Авторами Unity є Девід Хельгасон, Йоахім Анте, та Ніколас Френсіс. Вони мали на меті створити доступний ігровий рушій з професійним набором інструментів для розробників аматорів. Перша версія була доступна лише для MacOS X. Запуск Unity зробив розробку ігор більш доступною, що також підтверджується другим місцем в категорії “краще використання графіки MacOS X” на церемонії нагородження Apple Design Awards від компанії Apple Inc. у 2006 році [38].

Подальші версії ігрового рушія додали безліч нових функцій, та значно розширили кількість платформ, що підтримуються. З останніх оновлень, найцікавішими є підтримка конвеєру скриптового рендерінгу для створення високоякісної графіки, та інструменти для роботи з машинним навчанням (Unity 2018).

Unity є кросс-платформенним ігровим рушієм, що підтримує широкий спектр платформ [21]. Редактор Unity може працювати на платформах Windows, MacOS, Linux, при цьому сам ігровий рушій може працювати на 19 різних платформах включаючи такі мобільні платформи, як iOS, Android, настільні платформи, як Windows, MacOS, Universal Windows Platform, Linux, ігрові консолі, такі як: PS5, PS4, Xbox One, Xbox X|S, Nintendo Switch, встроєні платформи, як QNX, та навіть роботу в браузері завдяки підтримці компіляції в WebGL додаток.

Ігровий рушій Unity має декілька моделей ліцензування

[23], включаючи безкоштовну, та платні. Безкоштовна ліцензія призначена для особистого використання або для невеликих компаній, які отримують менше 200 000 доларів на рік, та не розробляють ігор для закритих ігрових консолей, таких як PlayStation, Nintendo Switch, та Xbox. Платними ж є підписки Plus (399\$/yr per seat), Pro (2040\$/yr per seat), та Enterprise. Підписка Plus майже не має функціональної різниці з безкоштовним використанням, окрім можливості зміни загрузочного екрану. Pro має дещо ширший функціонал, що включає розширену технічну підтримку, та можливість розробки ігор для закритих ігрових консолей, таких як PlayStation, Nintendo Switch, та Xbox.

Ігровий рушій Unity створений для роботи як з 2D-просторовими додатками, так і для 3D-просторових додатків. Цей ігровий рушій надає функціонал для роботи з графікою, звуком, фізикою, мережевою роботою, та взаємодією з користувачем. Цей ігровий рушій використовує мову розробки C# з використанням Mono як для інтегрованого середовища розробки Unity, у вигляді плагінів, так і для створення самих ігор. У минулому Unity підтримував також використання реалізацію мови програмування JavaScript, але цей функціонал був вилучений у серпні 2017 року. Ігровий рушій Unity забезпечує широкі можливості інтеграції з різноманітними сторонніми інструментами. Що дає можливості легко інтегрувати аналітику, соціальні мережі, рекламу, тощо.

Окрім цього Unity дає розробникам потужний візуальний редактор, що дозволяє створювати, та редагувати сцени, об'єкти, матеріали, та інше в інтуїтивно-зрозумілому інтерфейсі. Також цей редактор має функціонал попереднього перегляду коду.

Unity дає розробникам доступ до широкого набору готових компонентів, що можуть бути використані в проекті. Це включає 3D-моделі персонажів, та об'єктів оточення, готові скрипти, графічні ефекти, звукові ефекти, та інше. Це дозволяє значно спростувати процес розробки.

Ще варто відмітити велику, та активну спільноту розробників, що використовують ігровий рушій Unity, які діляться знаннями, ресурсами, та порадами.

Unity є одним з найпопулярніших ігрових рушіїв, що використовується для створення додатків з доповненою реальністю. Розробка додатків з доповненою реальністю в Unity відбувається за допомогою фреймворку AR Foundation.

AR Foundation надає багато бібліотек та SDK для кросс-платформеної розробки додатків з використанням доповненої реальності за допомогою ігрового рушія Unity. AR Foundation пропонує набір інтерфейсів для функцій доповненої реальності, але не реалізує ці функції. Для реалізації функцій, цей фреймворк використовує SDK, що пропонується для розробки додатків з доповненою реальністю для цільової платформи [24] (рисунок 2.1). Так для телефонів, та планшетів на базі операційної системи Android це ARCore, для телефонів та планшетів на базі операційної системи iOS це ARKit, а для оптичної гарнітури Microsoft HoloLens 2 це OpenXR.

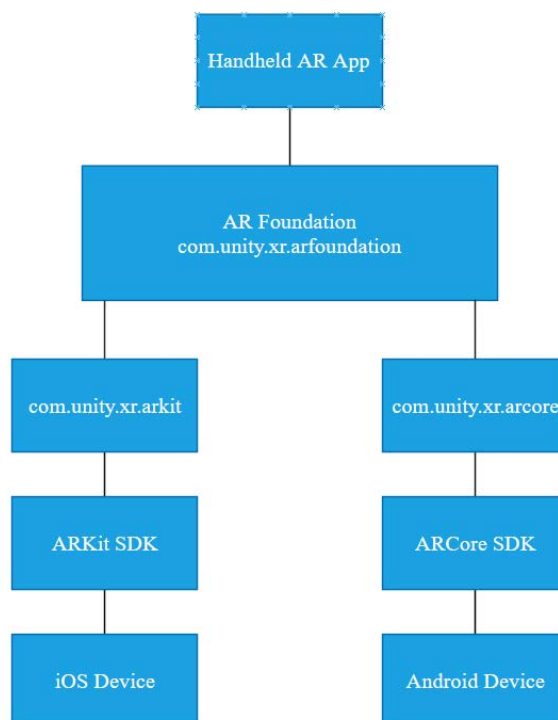


Рисунок 2.1 – Архітектура AR Foundation

Фреймворк AR Foundation пропонує досить широкий спектр функціоналу, що включає наступне:

- управління сесіями AR. Фреймворк надає можливості для включення, відключення, та конфігурування AR сесії на цільовій платформі;

- відслідковування девайсу. Відслідковування девайсу включає знаходження позиції, обертання в фізичному просторі;
- камера. Фреймворк надає можливість отримання зображення з камери, та робити оцінку освітленості;
- знаходження площини. Фреймворк має функціонал для відслідковування вертикальних та горизонтальних поверхонь;
- відслідковування зображення. Фреймворк надає можливість для знаходження та відслідковування заданого зображення в реальному просторі;
- відслідковування об'єктів. Фреймворк надає можливість для знаходження та відслідковування заданого 3D об'єкту в реальному просторі;
- відслідковування лица. Фреймворк має функціонал для знаходження та відслідковування людських лиць;
- відслідковування тіла. Фреймворк має функціонал для знаходження та відслідковування людських тіл;
- відслідковування хмари точок. Фреймворк надає функціонал для відслідковування характерних точок;
- перевірка попаданням променя. Це дає можливість знаходити об'єкти що перетинаються променем, що є необхідною функцією для взаємодії з об'єктом;
- якірні точки. Це функціонал, що дає можливість відслідковувати задані точки. є досить ресурсозатратним;
- побудова кубічної карти середовища. Завдяки цій функції можливо створювати високоякісні відображення на віртуальних об'єктах;
- побудова 3D сітки. Завдяки цьому можливо створити 3D модель реального середовища.

Підтримка функціоналу AR Foundation при роботі з різними SDK доповненої реальності наведено у (таблиця 2.1).

Таблиця 2.1 – Підтримка функціоналу AR Foundation з різними SDK

Функціонал	ARCore	ARKit	OpenXR
Управління сесіями AR	Так	Так	Так
Відслідковування девайсу	Так	Так	Так
Камера	Так	Так	
Знаходження площини	Так	Так	
Відслідковування зображення	Так	Так	
Відслідковування об'єктів		Так	
Відслідковування лиця	Так	Так	
Відслідковування тіла		Так	
Відслідковування хмари точок	Так	Так	
Перевірка попаданням променя	Так	Так	
Якірні точки	Так	Так	Так
Побудова кубічної карти середовища	Так	Так	
Побудова 3D сітки		Так	Так

Наведені дані (таблиця 2.1) демонструють, що не всі платформи підтримують весь функціонал. При розробці додатку, що повинен працювати як на девайсах під управлінням Android, так і під iOS, потрібно відштовхуватись від можливостей набору для розробки ARCore.

2.1.2 Unreal Engine

Unreal Engine - потужний ігровий рушій, розроблений компанією Epic Games. Він є одним з найпопулярніших ігрових рушіїв у світі. Так як і Unity, Unreal Engine має інтегроване середовище розробки. Unreal Engine відомий фотореліастичністю та якістю графічних ефектів в цілому, завдяки чому знайшов використання в ряді різноманітних сфер. Також Unreal Engine має підтримку широкого спектра функцій, та платформ.

Unreal Engine за деякими даними займає 15% ринку ігрових рушіїв [40], будучи при цьому другим за популярністю ігровим рушієм після Unity.

Цей ігровий рушій має багату історію, що починається ще з 1998 року, коли компанія Epic Games продемонструвала гру-шутер Unreal. Перше покоління Unreal Engine було розроблено засновником компанії Epic Games, Тімом Суїні. Створення ігрового рушія Unreal Engine почалось ще у 1995 році. 90 відсотків першої версії Unreal Engine було написано Тімом Суїні. За роки існування рушію було створено безліч ігрових додатків, та інших застосунків з використанням Unreal Engine. Найвідомішими прикладами застосування Unreal Engine є такі ігри як Fortnite, PUBG: Battlegrounds, Fantasy VII Remake. Також варто зауважити, що цей ігровий рушій активно використовується в кінопродакшені. Прикладом є серіал The Mandalorian. Також, цей ігровий рушій використовується у симуляторах для військових потреб [25].

Остання версія ігрового рушія Unreal Engine 5 демонструє новий рівень комп'ютерної графіки. Однією з особливостей оновлення є графічний рушій Nanite, що дає можливість працювати з віртуалізованою геометрією на базі високо деталізованих фотографічних вхідних даних [35]. Ще однією з особливостей є система повністю динамічного освітлення Lumen [36], що миттєво реагує на зміни в ігровій сцені.

Unreal Engine також є кросс-платформеним рішенням, та офіційно підтримує наступні платформи, включно з операційними системами для персональних комп'ютерів, Windows, macOS, Linux, мобільні операційні системи, iOS, Android, Закриті ігрові консолі, Nintendo Switch, PlayStation 4, Xbox One, PlayStation 5, Xbox Series X/S.

Згідно з стандартною ліцензією [26], Unreal Engine є безкоштовним для використання для перших 1 мільйонів доларів США, та має 5 відсотків роялті після цього порогу. Також, можлива розширена підтримка, що коштує 1500 доларів США за людину на рік. Крім цього, будь хто може отримати доступ до сирцевого коду Unreal Engine.

Основною задачею Unreal Engine є робота з 3D-просторовими додатками, але також може бути використаним для 2D-просторових додатків. Unreal Engine надає великий набір потужних інструментів для створення ігрових додатків різної складності. Це включає в себе візуальні редактори світу, матеріалів, штучного інтелекту, ігрової логіки. Цей ігровий рушій має потужні можливості для рендерингу, дозволяючи створювати вражаючі графічні зображення. Unreal Engine підтримує можливості для фотореалістичного рендерингу, високоякісних тіней, води, ефектів, тощо. Крім цього, Unreal Engine має вбудовану систему фізики, що підтримує руйнування, симуляцію твердих тіл, рідини, тканини, тощо. Також присутні потужні та різноманітні мультимедійні можливості для роботи з аудіо, зображеннями, та іншими видами мультимедійної інформації.

Основною мовою програмування для Unreal Engine є C++. Мова C++ використана як для самого ігрового рушія Unreal Engine, так і є основною мовою для створення додатків з використанням Unreal Engine. Крім цього варто відмітити наявність системи візуального програмування Blueprint. Візуальне програмування в Unreal Engine може майже повністю змінювати написання коду на C++. Система візуального програмування Blueprint є потужним інструментом для швидкої розробки ігор, будучи інтуїтивною, та легкою в засвоєнні, хоча і мають дещо гіршу продуктивність порівняно з C++. Крім того, Unreal Engine дозволяє інтегрувати код написаний на C++ в систему візуальної розробки Blueprint, тим самим розширюючи її. Blueprint має можливість роботи з ігровою логікою, штучним інтелектом, матеріалами, та навіть зі звуком. Крім цього Epic Games анонсували нову мову програмування для написання скриптів для Unreal Engine під назвою Verse. Ця мова написання скриптів стане доступна для всіх користувачів ігрового рушія Unreal Engine до 2025 року [34]

Також слід відмітити наявність магазину активів, де розробники можуть продавати, або купувати готові ігрові ресурси, такі як 3D-моделі, матеріали, звукові ефекти, плагіни, та інше. Це дозволяє економити час та зусилля для створення ігор, прискорюючи процес в цілому.

Для роботи з доповненою реальністю Unreal Engine пропонує уніфікований фреймворк [27], який інкапсулює роботу з різними платформами, даючи крос-платформний інструмент. Для роботи з доповненою реальністю на девайсах під управлінням операційної системи Android, фреймворк використовує ARCore SDK, а для роботи доповненої реальності на телефонах, та планшетах компанії Apple, фреймворк використовує ARKit SDK. Уніфікований фреймворк надає наступні функціональні можливості:

- управління сесіями AR. Фреймворк надає можливості для запуску та зупинки сесій AR;
- відслідковування девайсу. Функціонал, що дає можливість відслідковувати просторове положення, обертання девайсу в реальному світі;
- камера. Можливість отримання зображення реального світу. Також, можливість оцінки освітленості;
- знаходження площини. Можливість знаходження та відслідковування горизонтальних та вертикальних площин;
- відслідковування зображення. Функціонал зі знаходження та відслідковування заданих зображень у реальному світі;
- відслідковування об'єктів. Функціонал зі знаходження та відслідковування заданих 3D-об'єктів у реальному світі;
- відслідковування лиця. Знаходження та відслідковування людських лиць;
- якірні точки. Можливість відслідковувати задані точки.

Підтримка функціоналу уніфікованого фреймворку для доповненої реальності Unreal Engine при роботі з різними SDK доповненої реальності наведено у (таблиця 2.2).

Таблиця 2.2 – Підтримка функціоналу уніфікованого фреймворку для доповненої реальності Unreal Engine з різними SDK

Функціонал	ARCore	ARKit
Управління сесіями AR	Так	Так
Відслідковування девайсу	Так	Так
Камера	Так	Так
Знаходження площини	Так	Так
Відслідковування зображення	Так	Так
Відслідковування об'єктів		Так
Відслідковування лиця	Так	Так
Якірні точки	Так	Так

Наведені дані (таблиця 2.1) демонструють, що набір для розробки додатків з доповненою реальністю для платформи Android має дещо меншу підтримку функціоналу уніфікованого фреймворку для доповненої реальності Unreal Engine. При розробці реального додатку потрібно враховувати відсутність підтримки функціоналу відслідковування об'єктів на платформі Android.

2.2 Бекенд-сервіс рішення для ігрових додатків

Бекенд-сервіс є підходом для розробки багатокористувацьких додатків, який має на увазі використання стороннього сервісу, що виконує функції бекенду для гри. Цей сервіс забезпечує управління багатокористувацькими сесіями, зберігання даних.

Основною перевагою використання бекенд-сервісних рішень є швидкість розробки. Сервіс надає готові рішення, що дозволяють швидко реалізувати такі компоненти, як автентифікація користувачів, матчмейкінг, зберігання та обробка даних. Ще однією важливою перевагою є масштабування та надійність, що забезпечується тим, що інфраструктура таких сервісів зазвичай розміщена на розподілених сервісах, забезпечуючи високу надійність навіть при великій кількості

гравців. Крім цього, бекенд-сервіси надають зручні інтерфейси для керування даними користувачів.

Із недоліків даного підходу можна перш за все назвати обмежені можливості, у порівнянні з власним серверним рішенням. Це включає обмежені можливості для налаштування, розширення, та гнучкість функцій загалом. Також, недоліком є те, що додаток стає залежним від стороннього сервісу, що є проблематичним у разі виходу з ладу сервісу, або зміни умов надання послуг. Крім цього, вартість використання такого сервісу може бути значно вищою у разі великого проекту.

Важливо врахувати переваги та недоліки бекенд-сервісу перед використанням його у реальному продукту.

Слід розглянути популярні бекенд-сервісні рішення, що є на ринку, це GameSparks, та Photon.

2.2.1 GameSparks

GameSparks це повністю керований сервіс AWS від Amazon, що представляє з себе платформу Backend-as-a-Service (BaaS), або Бекенд-сервіс платформу. Ця Платформа створена для спрощення процесу розробки ігор, даючи можливість концентруватися розробникам на створенні самої гри, не витрачаючи час на управління та масштабування хмарної інфраструктури, залишивши цю роботу на платформу GameSparks. Прикладами ігор, що використовують цю платформу є такі ігри як: 7 Days To Die, Pac-Man Pop!. Також, платформа GameSparks надає різноманітні мультиплеєрні можливості, включаючи [28]:

- хмарний код. Цей функціонал дає можливість розробникам додавати код, що буде виконуватись в хмарі, завдяки інтеграції з іншими сервісами AWS;
- система повідомлень. Завдяки системі повідомлень можливо налагодити обмін даними між ігровими клієнтами;
- середовище тестування. GameSparks надає можливість тестування функцій в консолі, без необхідності тестування в грі, що дає можливість значно економити час;

- автентифікація користувачів. Функціонал, що надає можливість для створення профілю, та аутентифікації користувачів за допомогою різних методів, у тому числі за допомогою соціальних мереж;
- рейтингові дошки, та ранг. Цей функціонал дає можливість будувати рейтингові дошки, разом з системою рангу. Це мотивує гравців продовжувати грати;
- сховище для даних користувачів. GameSparks надає певне сховище для зберігання даних користувачів гри;
- інтеграція з AWS Lambda. Функціонал, що дає можливість викликати AWS Lambda з гри, інтегруючи хмарні обчислення;
- інтеграція з DynamoDB. Функціонал, що дозволяє інтегрувати DynamoDB в ігровий додаток, для зберігання певних ігрових даних.

Також варто розглянути цінову політику [29]. Використання GameSparks вимірюється у кількості запитів на виконання коду, викликів API GameSparks, використання простору для зберігання даних. У рамках безкоштовного пакету, який триває 12 місяців після створення нового облікового запису AWS, наявно 1 мільйон викликів API GameSparks, що є комбінованим числом для викликів коду та викликів самого API, та 0.5 гб сховища на місяць. Після закінчення 12 місячного періоду, або після перевищення безкоштовного об'єму викликів API, або сховища даних, потрібно сплачувати за послуги з наступним тарифом (таблиця 2.3).

Таблиця 2.3 – Тарифи GameSparks

Розмірність	Регіон US East (N. Virginia)	Регіон Asia Pacific (Tokyo)
Виклики коду (за мільйон запитів)	3\$	4.83\$
Виклики API GameSpark (за мільйон запитів)	3\$	4.83\$
Сховище (за гб-місяць)	0.48\$	0.59\$

Завдяки широкому спектру функціоналу, гнучкості у налаштуванні, можливості інтеграції з соціальними мережами, та можливості інтеграції стороннього коду в бекенд-сервісне рішення, GameSparks є досить потужним, та цікавим рішенням. Але

потрібно не забувати про недоліки, серед яких те, що ця бекенд-сервісна платформа не має такого широкого використання, тобто аудиторії, як інші рішення, що обмежує можливості для отримання допомоги. Також з обмежень можна назвати те, що через динамічну цінову політику, використання GameSparks може бути високовартністим для великих проектів.

2.2.2 Photon Engine

Photon Engine - це платформа, спеціалізована на наданні послуг для побудови багатокористувацьких ігор. Photon Engine створено компанією Exit Games з головним офісом у місті Гамбург, Німеччина. Photon Engine виступає у ролі Backend-as-a-Service (BaaS), або бекенд-сервіс. Цей бекенд-сервіс є лідером на ринку мультиплеєрних ігрових сервісів, підтримуючи сотні ігор, та маючи більше ніж 1.4 мільярди активних користувачів щомісяця. Прикладами відомих ігор, що використовують цей сервіс є такі ігри як: Among US, Stumble Guys. Photon Engine надає стабільну, швидку, та легкомасштабуєму інфраструктуру для ігрових додатків.

Однією зі складових Photon Engine є Photon Realtime, що надає можливості для створення ігрових додатків з малою затримкою. Photon Realtime підтримує широкий ряд ігрових рушіїв, та платформ, включаючи такі рушії, як Unity, Unreal Engine, Cocos2D, Construct2, Corona, та роботу на таких платформах персональних комп'ютерів як Windows, Linux, MacOS, мобільних платформах, як iOS, Android, закритих ігрових консолях, як PlayStation 4, PlayStation 5, Xbox One, Nintendo Switch.

Ця платформа пропонує наступні функції [30]:

- хмара реального часу. Хмара, що глобально розподілена по світу забезпечує мінімальні затримки прийому-передачі даних гравців;
- мультиплеєр. Ця платформа бере на себе усі задачі пов'язані з організацією мультиплеєра будь-якого типу, спрощуючи розробку ігор;
- матчмейкінг. Гнучке API підбору ігор дає можливість випадкового або параметризованого підбору гравців в сесію;
- синхронізовані та асинхронні ігри. Можливість створення різного типу ігор, включно з іграми, які дозволяють підключатись-відключатись від ігрової сесії;

- хмарне сховище. Сервіс надає можливість зберігати дані гравців, кімнат, подій на хмарному сховищі;
- крос-платформенна робота. Широкий спектр платформ, та ігрових рушіїв, що підтримуються платформою Photon Realtime, та можливість крос-платформеної гри робить цей інструмент ідеальним для цієї задачі;
- висока масштабуємість. Ігри, що використовують Photon Realtime масштабуються автоматично від декількох гравців, до десятків тисяч одночасно;
- можливість налаштування. Можливість створення різних типів кімнат, що дає можливість підтримувати різні режими гри;
- прозора цінова політика. Сервіс працює за схемою, що базується на кількості одночасно підключених користувачів. Це робить політику ціноутворення максимально прозорою.

Слід також розглянути тарифи на використання Photon Engine. Є 3 можливі варіанти тарифів, публічна хмара, преміум хмара, та ентерпрайз хмара. Публічна хмара надає 20 одночасних користувачів безкоштовно. Якщо потрібна більша кількість гравців, наявні наступні тарифи (таблиця 2.4).

Таблиця 2.4 – Тарифи на публічну хмару Photon Engine

Кількість одночасних користувачів	Ціна
100	95\$ за рік
500	95\$ за місяць
1000	185\$ за місяць
2000	370\$ за місяць

Преміум хмара пропонує гнучкий тариф, 0.29\$ за одного одночасного користувача. Преміум хмара може бути масштабована до 50000 одночасних користувачів. Ентерпрайз хмара це тариф за домовленістю для великих корпорацій. Цей тариф не має ніяких обмежень на кількість ігор, надає пріоритет з обслуговування командою, що працює 24\7. Також, включає виділені сервери, та захист від DDoS атак.

2.3 Висновки за розділом 2

Беручи до уваги дані про функціонал ігрових рушіїв Unity, та Unreal Engine неможливо однозначно зробити висновок, що є кращим для використання з доповненою реальністю. Ці два ігрові рушії прекрасно працюють на мобільних платформах, що підтверджено величезною кількістю комерційних проектів з використанням обох ігрових рушіїв. Також, обидва рушії мають величезні спільноти, що значно спрощує навчання. Крім цього, документація як Unreal Engine, так і Unity є досить детальною та якісною. Вибір конкретного ігрового рушія залежить від багатьох факторів, основними з яких є наявні навички у розробників з мовою програмування C#, або C++. Також, потрібно взяти до уваги бажану якість картинки. Unreal Engine надає значно потужніші інструменти для створення фотореалістичної картини у віртуальному світі. Крім цього, важливим фактором є час, що необхідно витратити розробнику на розробку. Ігровий рушія Unity є значно простішим для використання як під час навчання, так і під час розробки. Також, перевагою Unreal Engine є можливий доступ, та модифікація сирцевого коду ігрового рушія, що дає можливість розширювати будь-який аспект ігрового рушія. Також, потрібно взяти до уваги умови ліцензування. Ліцензія Unreal Engine є кращою для проектів до 1 мільйона доларів, так як при повному доступі до всіх ресурсів, не потрібно сплачувати жодних коштів, у той же час ліцензія Unity є потенційно дешевшою для проектів, що мають великі обороти коштів, так як немає роялті, як у Unreal Engine.

Що стосується бекенд-сервісних рішень, дані про функціонал GameSparks та Photon Engine вказують на те, що Photon Engine є дещо кращим рішенням порівняно з GameSparks. Photon Engine має кращу, прозору цінову політику, значно більшу спільноту розробників, що робить цей сервіс кращим рішенням, попри дещо менший функціонал через відсутність хмарного виконання функцій.

РОЗДІЛ 3 Розробка прототипу багатокористувацького ігрового додатку з доповненою реальністю

3.1 Вимоги до прототипу проекту

Основною метою розробки прототипу є демонстрація можливостей інтеграції доповненої реальності (AR) у багатокористувацький розважальний додаток. Використання доповненої реальності в багатокористувацькому додатку має за мету надати гравцям новий досвід у взаємодії з додатком, та нові соціальні інструменти.

Темою прототипу ігрового додатку є шашки, що мають просту для імплементації ігрову механіку, яка може бути адаптованою до використання як з AR, так і з багатокористувацьким режимом. Настільні ігри є одним з очевидних застосувань для доповненої реальності. Крім цього, жанр настільних ігор є одним з найпопулярніших для мобільних пристроїв.

Для розробки прототипу додатку було обрано ігровий рушій Unreal Engine 5.1 у поєднанні з бекенд-сервісом Photon Engine (Photon Realtime). Вибір ігрового рушія зумовлений наявністю візуальної системи програмування, що дає можливість швидкого прототипування Blueprint, та відсутністю явних недоліків, або переваг Unreal Engine, або Unity в контексті підтримки розробки AR додатків. Обидва ігрові рушії є надзвичайно потужними для проектів різного рівня, та мають величезну підтримку зі сторони спільноти. Unreal Engine забезпечує бездоганну підтримку безлічі платформ, даючи можливість прототипування мобільних додатків майже без необхідності запуску на реальних мобільних платформах, значно економлячи час витрачений на розробку додатку. Що стосується вибору бекенд-сервісного рішення для даної роботи, Photon Engine має переваги у порівнянні з GameSparks у прозорій ціновій політиці, та більш широкій підтримці зі сторони спільноти. Крім цього Photon Realtime надає зручний програмний інтерфейс для розробки.

Даний прототип має наступний варіант використання:

Передумови: наявність стабільного з'єднання з мережею інтернет. Також додаток потребує надання дозволу до файлової системи, та дозволу на використання камери девайсу.

Учасники: гравець 1, гравець 2.

Основний сценарій:

1. Гра стартує, запитує користувача про надання дозволів на використання внутрішнього сховища девайсу, та використання камери девайсу
2. Після надання дозволів, відкривається стартовий екран гри, запрошуючи користувача розпочати процес сканування навколишнього середовища за допомогою камери.
3. Після переходу до сканування, за допомогою натискання відповідної кнопки, змінюється екран інтерфейсу користувача, інформуючи про процес сканування заради пошуку горизонтальної площини.
4. Після знаходження горизонтальної площини, вона малюється за допомогою віртуального об'єкта. Гра очікує на підтвердження цієї площини.
5. Після вибору площини, користувачу пропонується вибрати позицію для ігрової дошки на даній площині, інформуючи про це на інтерфейсі користувача.
6. Після вибору положення ігрової дошки, користувачу пропонується підтвердити положення, нажавши відповідну кнопку, або повернутись до розміщення ігрової дошки.
7. Після підтвердження відбувається пошук, або створення онлайн ігрової сесії, та зміна інтерфейсу користувача, для інформування про це.
8. Після знаходження ігрової сесії, відображається аватар суперника, та фішки, комірки на дошці. Гравці можуть грати.
9. Після того, як один з гравців виграє, відображається екран, що інформує про виграш або програш. Також, пропонується почати нову онлайн сесію гри.

Прототип додатку передбачає 3 основних функціональних учасника взаємодії, це саме гравець, гра, та бекенд (Photon Engine). Логіку роботи з прототипом додатку продемонстровано на діаграмі послідовності (рисунок 3.1)

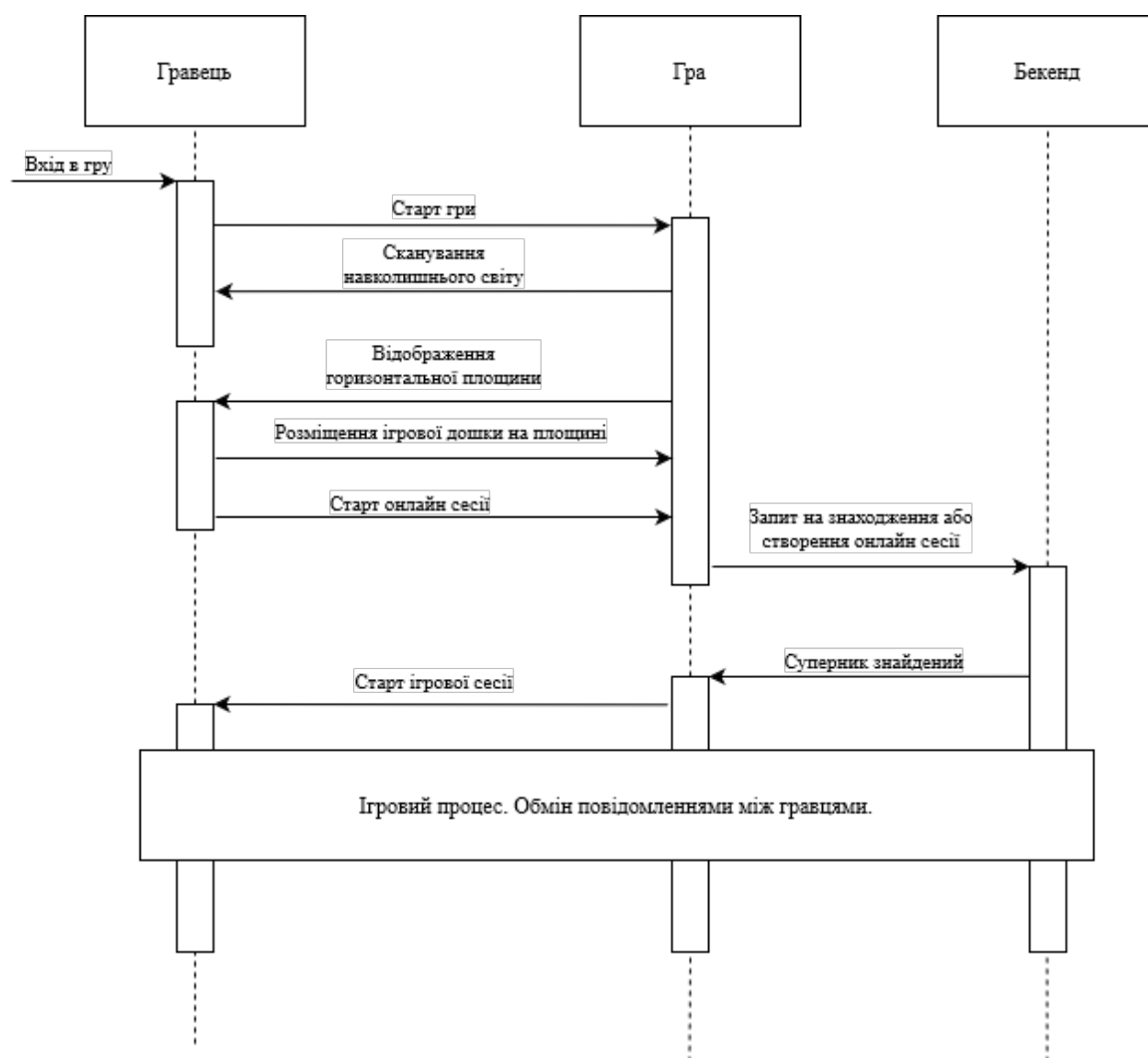


Рисунок 3.1 – Діаграма послідовності взаємодії з прототипом ігрового додатку

Прототип ігрового додатку повинен мати такі екрани, що повинні відповідати процесу наведеному на діаграмі послідовності (рисунок 3.1):

1. Стартовий екран з запрошенням до початку гри. Цей екран повинен містити логотип гри, та кнопку старту гри.
2. Екран сканування навколишнього простору. Цей екран повинен показувати зображення з камери телефону та візуальну інформацію про сканування зовнішнього простору.
3. Екран вибору площини. Цей екран повинен показувати знайдену площину, та інформацію про необхідність вибору площини. У разі якщо користувач не вибере знайдену площину, додаток повинен запропонувати нову через деякий час.
4. Екран розміщення ігрової дошки. Цей екран повинен запропонувати користувачу розмістити ігрову дошку на площині.

5. Екран з запрошенням до старту онлайн сесії. Цей екран повинен пропонувати гравцю розпочати онлайн сесію, або змінити розміщення ігрової дошки.

6. Екран очікування на суперника. Цей екран повинен мати візуальну інформацію, що відбувається пошук гри.

7. Екран з ігровим процесом. Екран повинен надавати можливості ігрового процесу, та суперника, надавати можливість демонстрації емоції.

8. Екран кінця гри. Цей екран повинен показувати повідомлення виграшу або програшу. Також, екран повинен пропонувати гравцю розпочати нову ігрову сесію.

Під час саме ігрового процесу гравець повинен бачити інтерактивну ігрову дошку, що розміщена на площині реального світу, та аватар суперника, що також є розміщеним у реальному світі, навпроти гравця. Прототип додатку повинен мати можливість взаємодіяти з ігровою дошкою завдяки сенсорному екрану, завдяки дотику до фішок, та комірок гральної дошки, роблячи ходи на ігровій дошці, створюючи враження гри на реальній дошці. Дошка повинна візуально відображати підказки відносно того, які наступні кроки може зробити гравець. Також, важливим є можливість відображення рухів аватару суперника, а саме переміщення, та обертання у просторі реального світу, відносно ігрової дошки, створюючи ефект присутності. Також, прототип ігрового додатку повинен надавати можливості для комунікації між гравцями. Зважаючи на простоту гри, можливість надсилати емоції є достатньою.

Діаграма класів (рисунок 3.2) зображує взаємодію основних класів ігрового процесу.

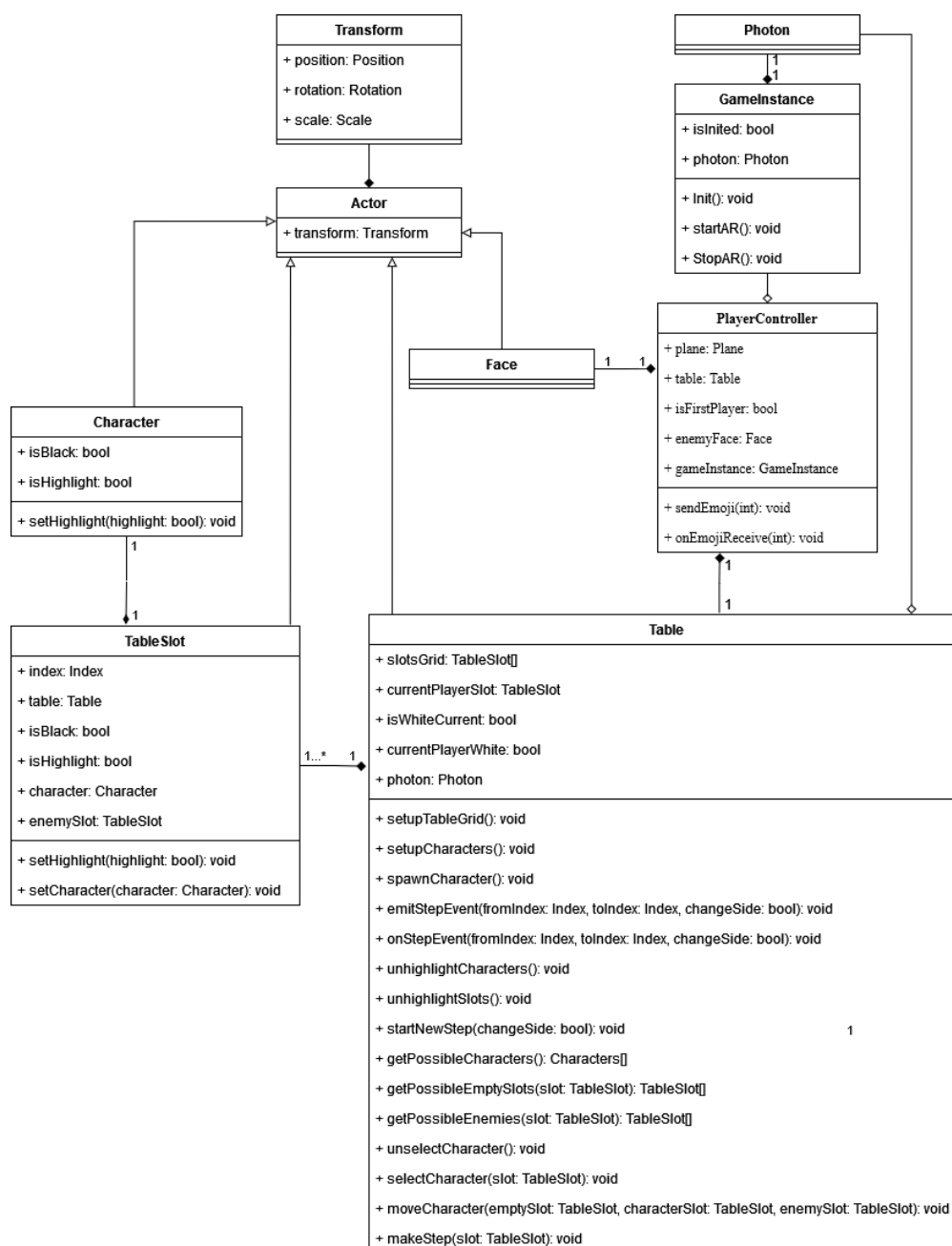


Рисунок 3.2 – Діаграма класів ігрового процесу

Клас Table є основним класом, що відповідає за ігрову логіку шашок, відповідаючи за створення дошки, створення фішок, розміщення фішок у просторі. Об'єкт цього класу є актором, та розміщується у 3D просторі, враховуючи положення площини доповненої реальності. Також, цей клас відповідає за обрахунок наступних можливих кроків, можливих фішок для наступного кроку, виділяючи їх візуально. Також цей клас відповідає за створення, та збереження фішок, та слотів ігрового поля. Окрім цього цей клас повинен передавати та приймати дані про ходи на ігровій дошці, забезпечуючи багатокористувацьку гру.

Клас `PlayerController` відповідає за керування об'єктом класу `Table`, та розміщенням аватара суперника в просторі, тобто є керуючим ігровим процесом.

Клас `Character` є класом, що описує ігрові фішки. Об'єкти цього класу також розміщуються у 3D просторі. Об'єкти цього класу можуть бути візуально виділеними, та можуть обробляти події дотику до них.

Клас `TableSlot` описує слот ігрової дошки. Об'єкти цього класу розміщуються у 3D просторі, на позиції, що задається ігровою дошкою. Об'єкти цього класу відповідають за розміщення об'єктів класу `Character` у 3D просторі. Об'єкти цього класу можуть бути візуально виділеними, та можуть обробляти події дотику до них.

Клас `Face` описує аватар суперника. Об'єкт цього класу може бути розміщеним у 3D просторі. Керуванням позицією та орієнтацією об'єкту цього класу займається `PlayerController`, передаючи та приймаючи дані про положення та обертання за допомогою мережевих можливостей. Позиція і орієнтація повинна бути відносно ігрової дошки.

Клас `Photon` відповідає за мережеві події. Об'єкт цього класу реалізує інтерфейс фреймворку бекенд-сервісу `Photon Engine`, а саме `Photon Realtime`. Цей об'єкт відповідає за створення, знаходження, та підключення до мережевої сесії. Завдяки цьому об'єкту передаються та приймаються дані через мережу інтернет.

3.2 Технічна реалізація

Прототип багатокористувацького ігрового додатку з використанням доповненої реальності, що був створений у ході виконання даної роботи, побудований по більшій частині з використанням системи візуального програмування `Blueprint`, що є частиною стандартного пакету `Unreal Engine`. `Blueprint` є повноцінною системою написання ігрових скриптів. Ця система візуального програмування використовує об'єктно-орієнтований підхід, тобто концепцію класів, об'єктів, наслідування, інкапсуляцію, та поліморфізм. Система `Blueprint` надає зручний інструмент, доступний для дизайнерів. Крім цього, використання візуального програмування є зручним для прототипування проектів, так як не потребує перекомпіляції сирцевого

коду проекту. Blueprint Система Blueprint може бути легко інтегрована з сирцевим кодом, наслідуючи класи та інтерфейси написані на C++.

Робота з мережею інтернет, для забезпечення можливостей багатокористувацького додатку реалізована за допомогою бекенд-сервісу Photon Realtime. Цей сервіс надає можливості мережевої роботи з досить низькими затримками, при цьому забезпечуючи стабільну роботу, що доказано його використанням в сотнях ігрових та не тільки проектах. Робота з цим сервісом є досить простою для його використання в прототипі ігрового додатку. Програмний інтерфейс API цього сервісу представлений у вигляді невеликого набору досить простих класів C++, забезпечуючи просту розробку.

За візуальне представлення 3D-об'єктів відповідають моделі, та матеріали. Матеріали представляють собою абстракцію над шейдерами, прощуючи роботу з ними, та надаючи інструмент, що не залежить від конкретного графічного API, будь то DirectX, OpenGL, Vulkan, або Metal. Робота з матеріалами в Unreal Engine також використовує систему візуального програмування, подібну до Blueprint. Візуальний редактор матеріалів є зручним і доступним для використанням дизайнерами.

Інтерфейс користувача представлений системою Slate Widgets. Ця система є досить гнучкою, для розробки ігрових інтерфейсів, представляючи такі елементи, як різноманітні кнопки, зображення, робота з текстом, та різноманітні контейнери, що можуть керувати розміщенням під елементів.

Структура проекту зображена (рисунок 3.3)

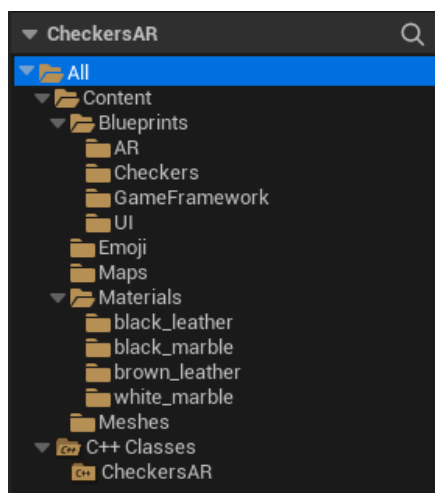


Рисунок 3.3 – Структура проекту

Як можна побачити (рисунок 3.3) проект має 2 структурні частини:

- Content. Включає об'єкти створені за допомогою візуальних редакторів, у тому числі і Blueprint, та ресурси гри.
- C++ Classes. Ця частина відповідає за C++ класи гри.

Частина Content містить такі директорії, як Blueprints, Emojis, Maps, Materials та Meshes. Директорія Blueprints зберігає класи, які реалізують ігрову логіку за допомогою системи візуального програмування Blueprint. Директорія Emojis містить ресурси зображень для емоцій, які можуть надсилати користувачі. Директорія Maps містить ігрову карту. Ця карта зберігає дані про розміщення світла у віртуальному середовищі. Директорія Material зберігає матеріали 3D об'єктів, та текстури деяких з них у піддиректоріях. Директорія Meshes відповідає за збереження 3D моделей фішки, та аватару суперника.

Ігрова логіка розділена на такі модулі, як:

- AR. Це модуль, що містить класи, які відповідають за роботу з реальним простором навколо гравця. У даному випадку, такими класами є:

CARPlane, який представляє візуальне відображення для відсканованої горизонтальної площини реального світу.

CARTablePlaceholder, що є об'єктом, який може бути розміщений на площині. За роботу з цими класами відповідає контроллер гравця (CARPlayerController), який є частиною модуля GameFramework.

- Checkers. Цей модуль відповідає саме за логіку шашок. Модуль представлений трьома класами:

CARCharacter, що відповідає за відображення фігури, та обробку подій дотику гравцем.

CARTableInteractSlot, цей клас відповідає за відображення комірки на ігровій дошці, та обробку подій дотику гравцем.

CARTable, що є основним класом цього модуля. Цей клас відповідає за створення ігрової дошки, що включає створення фішок, та слотів. Також цей клас відповідає за ігрову логіку дошки шашок, відповідає за обрахунок можливих кроків, виділяє можливі фішки, та слоти, що можуть бути використані на даному ігровому

кроці. Крім цього, цей клас відправляє та отримує мережеве повідомлення про зроблений ігровий хід.

- `GameFramework`. Це основний модуль прототипу додатку. Цей модуль включає наступні класи:

`CARGameInstance`, що відповідає за стан гри, у тому числі ініціалізує роботу з доповненою реальністю, та роботу з мережевою роботою за допомогою бекенд-сервісу `Photon Realtime`, передаючи дані про адресу сервера, ID додатку, та версію додатку на сервер.

`CARGameMode`, що відповідає за інформацію про основні ігрові класи проекту. Цей клас використовується для ініціалізації додатку при його старті.

`CARPawn`, клас який відповідає за ігровий об'єкт гравця. Цей клас також має в собі ігрову камеру, що відповідає за позицію з якої відбувається рендерінг зображення. Також, цей клас ініціалізує камеру для роботи додатку на десктоп платформах, що допомагає тестувати додаток без необхідності використання мобільного девайсу з камерою.

`CARPlayerController` цей клас є основним ігровим класом, так як керує ігровим станом. Стан гри є простою машиною станів. Об'єкт цього класу відповідає за запити на сканування навколишнього простору для пошуку горизонтальних площин що можуть бути використані для розміщення віртуальних об'єктів, а саме ігрової дошки у даному випадку. Після знаходження підходящої горизонтальної площини, цей клас оновлює стан та очікує на підтвердження цієї площини зі сторони користувача. Якщо користувач не підтвердить площину на протязі деякого часу, цей клас може замінити площину на іншу. Крім цього, цей клас відповідає за розміщення ігрової дошки на площину. Він очікує на дотик гравця до площини, щоб визначити положення дошки. Орієнтація дошки в просторі вираховується таким чином, щоб дошка була навпроти камери гравця. Також, цей клас відповідає за запит на старт онлайн ігрової сесії, та керуванням її повідомленнями, та створенням аватару суперника. Під час самої гри, цей клас оновлює дані про освітленість 3D-об'єктів, використовуючи дані, що отримуються з AR фреймворку, оновлює положення та орієнтацію аватару суперника, рахуючи позицію та орієнтацію відносно ігрової дошки, передаючи, та приймаючи ці

дані через Photon Realtime, та рахуючи положення, та орієнтацію використовуючи відносні дані. Крім цього, цей клас опрацьовує події помилок з'єднання з сервером.

— UI. Цей модуль відповідає за екрани інтерфейсу користувача. Кожен екран представлений окремим класом. У цьому модулі є наступні класи:

UI_GameMenu, що є основним екраном. Цей екран який підписується на події з оновлення стану гри з об'єкту класу CARPlayerController, змінюючи поточні віджети, тобто екрани, та поточний стан інтерфейсу користувача. Окрім цього, цей екран відповідає за вихід з гри.

UI_WelcomeWindow це екран, що зустрічає користувача після запуску прототипу додатку. На цьому екрані зображено логотип гри, та кнопку старту. Натиск на кнопку старту змінює стан гри на сканування.

UI_ScanEnvironment є наступним екраном, що повідомляє користувача про те, що відбувається сканування простору реального світу. Цей екран не має інтерактивних елементів інтерфейсу користувача.

UI_SelectPlane показується користувачу після того, як знайдено площину в реальному світі, повідомляючи користувача про необхідність вибору площини для розміщення ігрової дошки. Інтерфейс користувача немає ніяких інтерактивних елементів.

UI_PlacePlaceholder після вибору площини сповіщає користувача про необхідність розміщення ігрової дошки на площині. Цей інтерфейс є інформативним, без інтерактивних елементів.

UI_ApprovePlaceholder після вибору розміщення ігрової дошки, цей екран дає можливість зміни положення ігрової дошки, або розпочати онлайн сесію. Кнопка зміни положення ігрової дошки змінює стан гри на попередній. Кнопка початку онлайн сесії змінює стан гри на очікування суперника.

UI_WaitForPlayer цей екран сповіщає користувача про те, що відбувається пошук суперника для гри. Цей екран не має інтерактивних елементів.

UI_Game цей екран відображається після знаходження суперника та початку онлайн сесії. Крім цього, на екрані присутня можливість надсилання емоцій суперника, для певної комунікації.

UI_Finish цей екран відображається коли один з гравців виграв гру. На цьому екрані користувачу відображається виграв чи програв поточний гравець, та присутня кнопка що запрошує до нової ігрової сесії. Натяття на кнопку початку нової сесії змінює стан гри на очікування суперника.

UI_ErrorMessageDialog цей екран відображається у випадку помилки мережевого з'єднання, повідомляючи про помилку та даючи змогу вийти з гри.

Робота з мережевим з'єднанням за допомогою бекенд-сервісного рішення Photon Reatime реалізована з допомогою C++ класів, які розширюють систему візуального програмування Blueprint. Робота з Photon Reatime представлена за допомогою декількох класів C++:

PhotonListener це клас що реалізує інтерфейс прослуховування подій Photon Reatime, реалізуючи ExitGames::LoadBalancing::Listener. Цей клас працює з об'єктом що реалізує інтерфейс PhotonInterface, викликаючи методи об'єкту цього класу, надаючи можливість реагувати на мережеві події.

PhotonInterface це інтерфейс, що створений для взаємодії між Photon Reatime та системою класів Unreal Engine.

UPhoton є реалізацією інтерфейсу PhotonInterface, що окрім цього реалізує й інтерфейс FTickableGameObject та наслідується від базового класу UObject. Наслідування від UObject дає можливість інтеграції його класу в систему типів Unreal Engine, та розширювати систему візуального програмування Blueprint. Крім цього реалізація інтерфейсу FTickableGameObject розширює клас, даючи можливість реагувати на кожен такт ігрового додатку, за допомогою методу Tick(). Усі методи, що реалізують інтерфейс PhotonInterface викликають події, на які реагує CARPlayerController. Окрім цього клас UPhoton відповідає за ініціалізацію підсистеми Photon Reatime, та створює об'єкт класу PhotonListener для прослуховування подій.

3.3 Демонстрація роботи додатку

Демонстрація проекту була запущена на двох мобільних девайсах під управлінням операційної системи Android 11. Попри це, додаток може працювати і на

телефонах під управлінням операційної системи iOS. Теоретично, зважаючи на кращу реалізацію функціоналу доповненої реальності на телефонах під управлінням операційної системи iOS, додаток буде працювати навіть дещо краще через вищу потужність девайсів від компанії Apple [41].

Після старту прототипу додатку відбувається запит на дозволи для використання камери та внутрішнього сховища девайсу (рисунок 3.4).

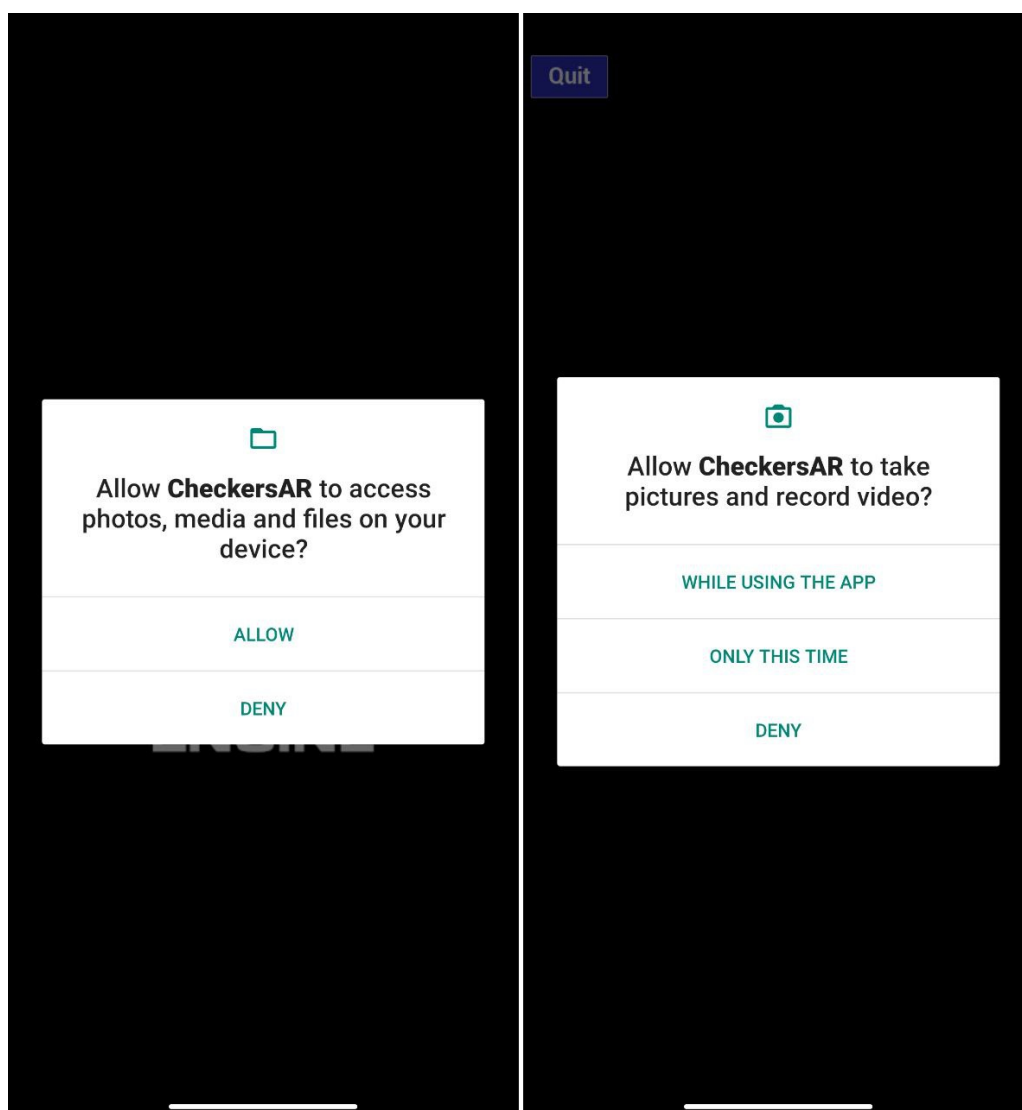


Рисунок 3.4 – Екрани запиту дозволів на використання внутрішнього сховища, та камери девайсу

Після того, як користувач дозволяє використання необхідних ресурсів, відбувається ініціалізація можливостей доповненої реальності, та ініціалізація Photon Realtime, для роботи з мережевими можливостями, відображається стартовий екран (рисунок 3.5)

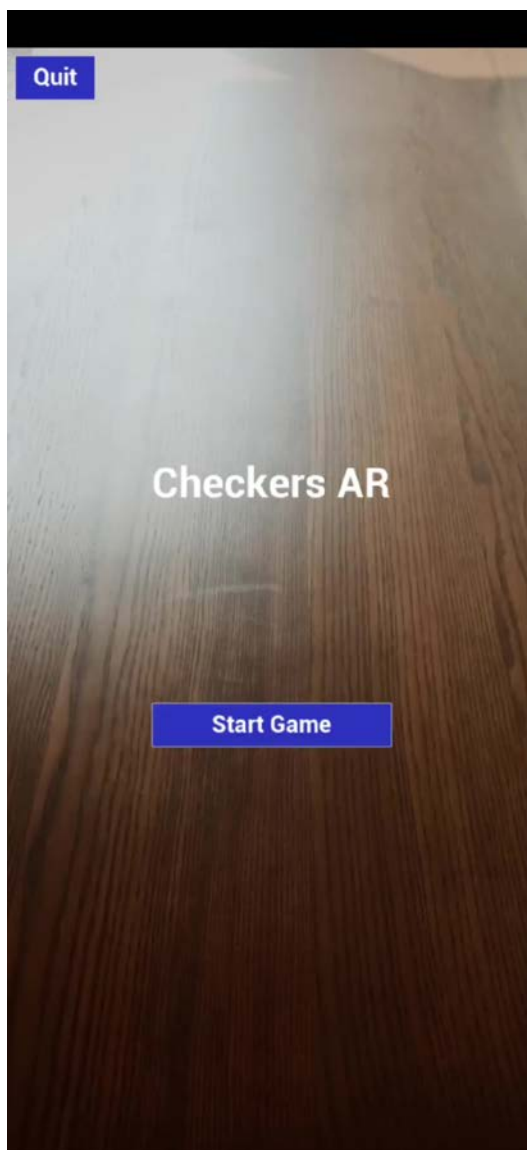


Рисунок 3.5 – Стартовий екран додатку

На стартовому екрані користувач бачить зображення з камери телефону, а такі елементи інтерфейсу користувача, як логотип гри, та кнопка початку гри.

Після натискання на кнопку початку гри, розпочинається пошук горизонтальної площини реального світу за допомогою можливостей доповненої реальності, що може бути використана для розміщення ігрової дошки, та відображається екран, що повідомляє користувача про це (рисунок 3.6)

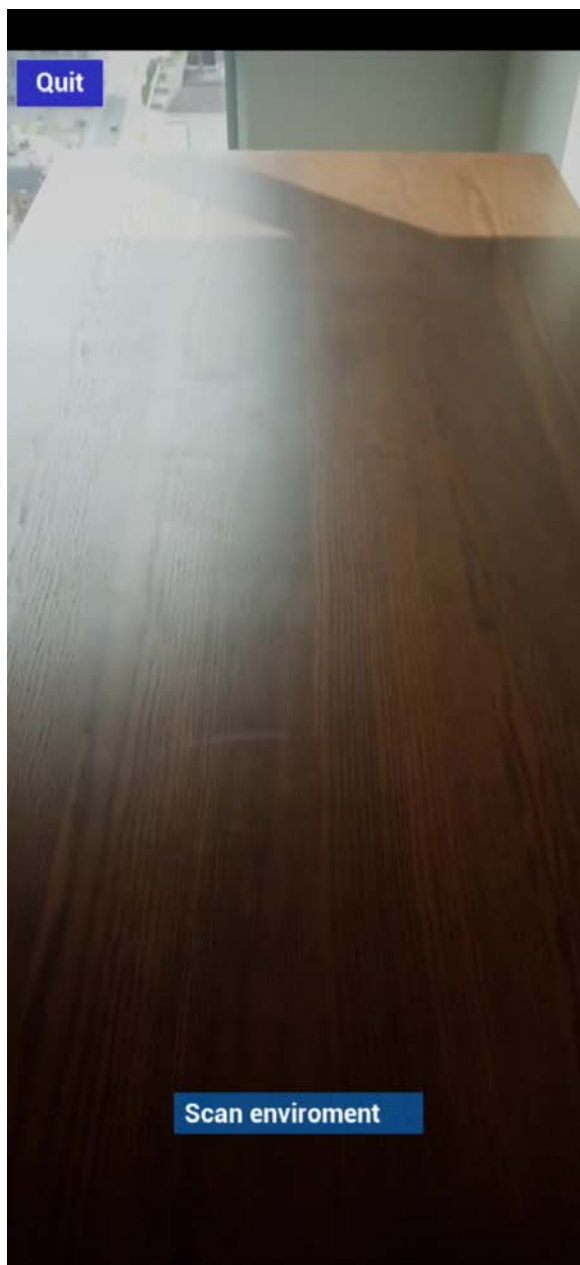


Рисунок 3.6 – Екран сканування навколишнього простору

Після знаходження першої горизонтальної площини, що може бути використана для розміщення ігрової дошки, відображається екран, що пропонує користувачу натиснути на площину, таким чином вибравши її (рисунок 3.7). Якщо гравець не вибере знайдену площину, вона може бути замінена на іншу, що буде автоматично знайдена ресурсами доповненої реальності.

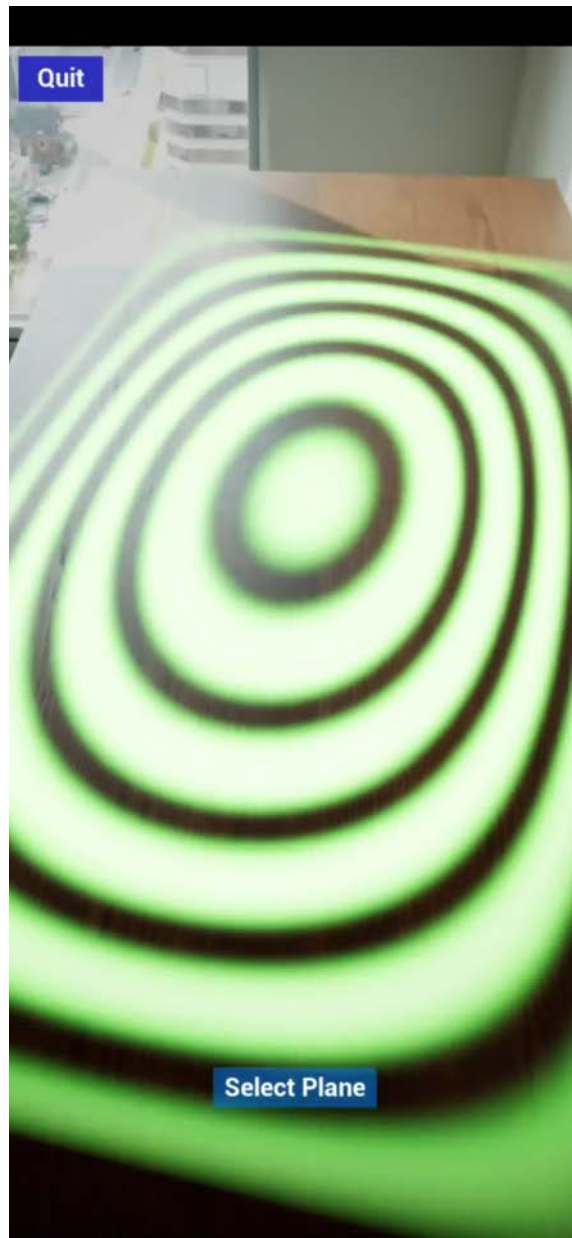


Рисунок 3.7 – Екран вибору площини

Після вибору площини, користувачу відображається екран, що пропонує розпочати онлайн сесію, або змінити розміщення ігрової дошки, натиснувши відповідну кнопку (рисунок 3.8).



Рисунок 3.8 – Екран підтвердження положення ігрової дошки

Після того, як гравець натискає на кнопку для старту онлайн сесії, додаток, за допомогою Photon Realtime, шукає ігрову кімнату, або створює нову, якщо немає вільного місця. Після чого гравець бачить екран, що повідомляє про пошук суперника.(рисунок 3.9)



Рисунок 3.9 – Екран пошуку суперника

Після того, як суперник знайдений, тобто знайдена ігрова кімната з вільним місцем, додаток відображає ігровий екран, що представляє саме ігровий процес. На екрані відображається ігрова дошка, аватар суперника, та можливість відправляти емоції. Положення та орієнтація аватару суперника по відношенню до ігрової дошки синхронізується за допомогою Photon Realtime з відповідним положенням та орієнтацією телефону суперника (рисунок 3.10)

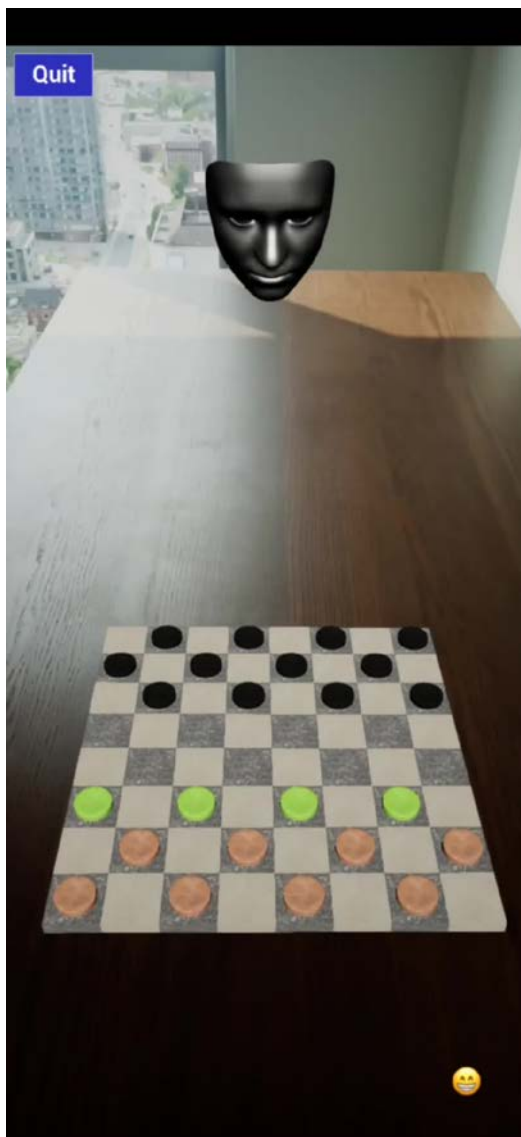


Рисунок 3.10 – Початок ігрового процесу

Окрім цього, гравці можуть надсилати емоції суперникам, що дає певні можливості для комунікації між собою (рисунок 3.11)



Рисунок 3.11 – Гравець 2 відправив гравцю 1 емоцію

Гравець, який повинен зробити свій хід, бачить виділені фішки, якими можливо зробити хід. Крім цього, після вибору фішки, гравець бачить перед собою комірki у які можна зробити хід (рисунок 3.12). Також, гравець може натиснути на фішку щоб відмінити вибір.

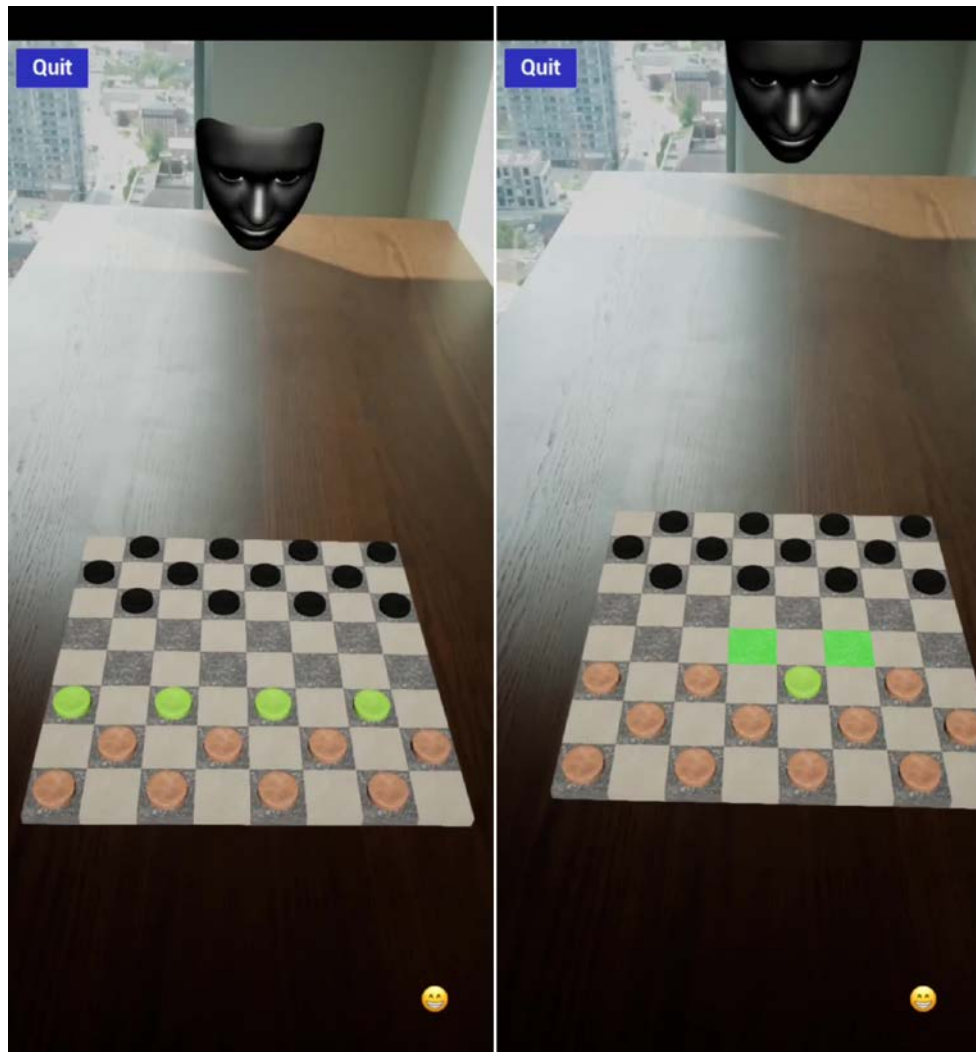


Рисунок 3.12 – Відображення можливих фішок, та можливих ходів після вибору фішки

Після завершення гри, а саме коли на дошці не залишається фішок одного з гравців, відображається екран, що повідомляє гравців про виграш, та пропонує почати нову онлайн сесію (рисунок 3.13) Крім цього, гравці автоматично відключаються від онлайн сесії, звільняючи ресурси Photon Realtime.

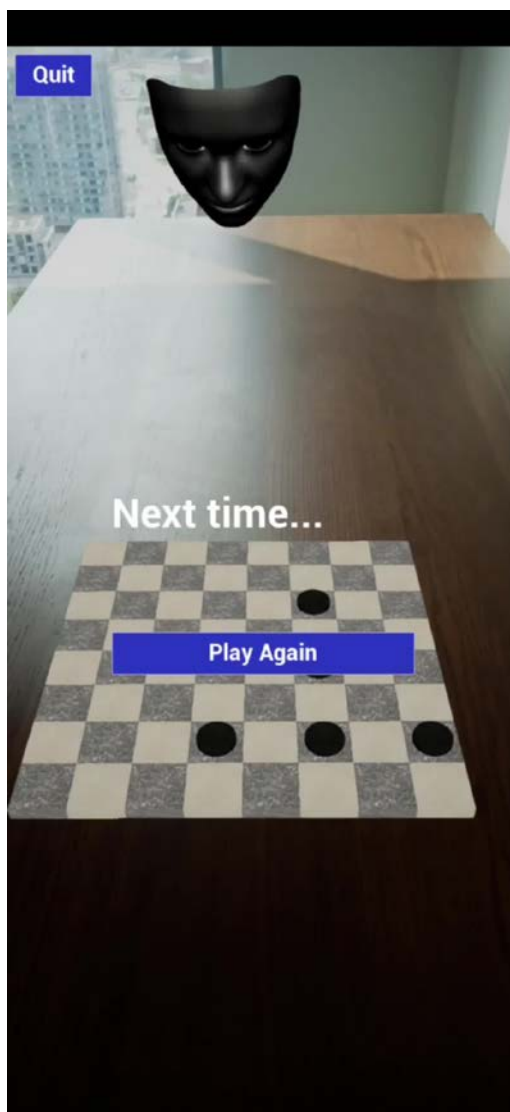


Рисунок 3.13 – Екран кінця гри

Крім цього, у разі проблем зі з'єднанням з серверами Photon, гра повідомляє про це гравця, відображаючи екран з помилкою, та пропонуючи завершити гру (рисунок 3.14)

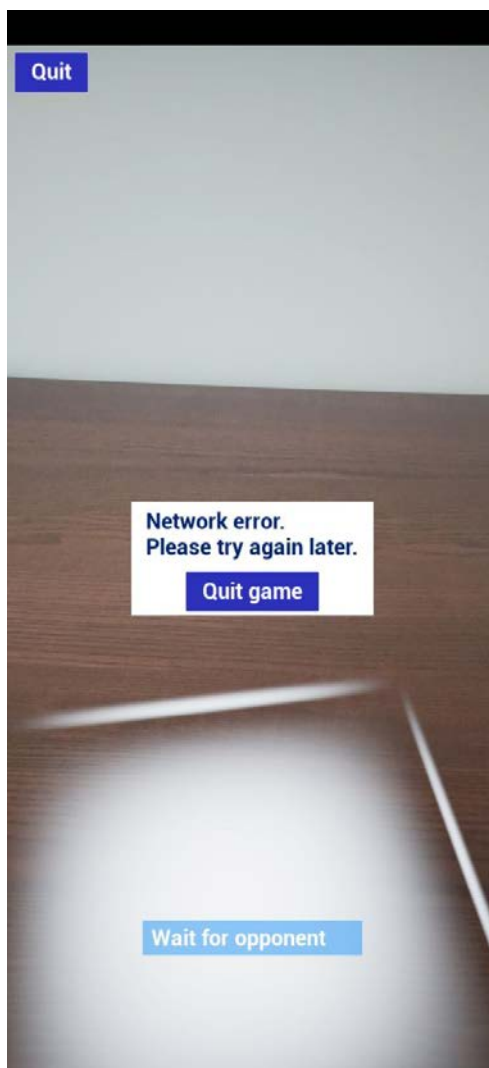


Рисунок 3.14 – Екран мережевої помилки

Також, якщо якийсь з гравців вийде з гри під час онлайн сесії, незалежно від того, чи це було через зв'язок з мережею, проблемою з девайсом, чи з додатком, або навмисно, інший гравець автоматично перейде до екрану пошуку нового суперника.

3.4 Висновки за розділом 3

Результатом роботи над даним розділом є прототип багатокористувацького ігрового додатку з використанням технологій доповненої реальності. Прототип було розроблено з використанням ігрового рушія Unreal Engine 5, та бекенд-сервісного рішення Photon Engine. Також було продемонстровано роботу цього додатку на мобільних девайсах під управлінням операційної системи Android 11.

ВИСНОВКИ

Дана робота складається з трьох розділів.

У першому розділі було зроблено огляд технології доповненої реальності, порівняння доповненої реальності з віртуальною реальністю. Також, було оглянуто технології доповненої реальності на мобільній платформі iOS, а саме ARKit, та на мобільній платформі Android, а саме ARCore. Крім цього, було розглянуто потенційний вплив доповненої реальності на багатокористувацькі додатки.

У другому розділі даної роботи було розглянуто ігрові рушії, що мають підтримку роботи з функціоналом доповненої реальності. Також, було зроблено огляд бекенд-сервіс платформ, що можуть бути використані для побудови багатокористувацького ігрового додатку.

У третьому розділі було описано розробку прототипу багатокористувацького ігрового додатку з використанням доповненої реальності. Було продемонстровано його роботу на смартфонах під управлінням операційної системи Android 11.

Використання доповненої реальності в багатокористувацьких ігрових додатках є досить цікавим як і для комерційних цілей, так і для надання користувачам нових соціальних можливостей під час гри, що сприяє популярності таких додатків.

У ході виконання даної роботи було продемонстровано можливість поєднання технологій доповненої реальності з мережевими можливостями для створення розважальних ігрових додатків. Продемонстровано роботу такого додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Digital around the world. Datareportal. URL: <https://datareportal.com/global-digital-overview> (дата звернення: 21.05.2023).
2. Mobile gaming statistics for 2023 that will blow your mind. udonis. URL: <https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics> (дата звернення: 21.05.2023).
3. Mobile gaming market report. P&S Intelligence. URL: <https://www.psmarketresearch.com/market-analysis/mobile-gaming-market> (дата звернення: 21.05.2023).
4. The past, present, and future of virtual and augmented reality research: a network and cluster analysis of the literature / P. Cipresso та ін. Front psychol. 2018. URL: <https://doi.org/10.3389/fpsyg.2018.02086> (дата звернення: 21.05.2023).
5. Luckin R. Limitless or pointless? An evaluation of augmented reality technology in the school and home. International journal of technology enhanced learning. № 2011. URL: <https://doi.org/10.1504/IJTEL.2011.042102> (дата звернення: 21.05.2023).
6. Augmented reality navigation system for endoscopic surgery based on three-dimensional ultrasound and computed tomography: application to 20 clinical cases / K. Konishi та ін. ScienceDirect. 2005. URL: <https://doi.org/10.1016/j.ics.2005.03.234> (дата звернення: 21.05.2023).
7. Kastrenakes J. Pepsi's bus stop ad in London might be the best use of augmented reality yet. The Verge. URL: <https://www.theverge.com/2014/3/25/5545842/pepsi-bus-stop-ad-augmented-reality> (дата звернення: 21.05.2023).
8. Unbelievable bus shelter. Grand Visual. URL: <https://grandvisual.com/work/pepsi-max-bus-shelter/> (дата звернення: 21.05.2023).
9. Ayoubi A. IKEA launches augmented reality application. Architect Magazine. URL: <https://www.architectmagazine.com/technology/ikea-launches-augmented-reality-application> (дата звернення: 21.05.2023).

10. Iqbal M. Pokémon go revenue and usage statistics (2023). Business of Apps. URL: <https://www.businessofapps.com/data/pokemon-go-statistics/> (дата звернення: 21.05.2023).
11. Augmented reality technologies, systems and applications / J. Carmigniani та ін. Multimedia tools and applications. 2010. URL: <https://link.springer.com/article/10.1007/s11042-010-0660-6> (дата звернення: 21.05.2023).
12. Colaner S. What's inside microsoft's hololens and how it works. Tom's Hardware. URL: <https://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm,32546.html> (дата звернення: 21.05.2023).
13. Stausbøl N. Another marker generator for ar.js. codepen. URL: <https://codepen.io/staus/full/oEOJpq> (дата звернення: 21.05.2023).
14. Image tracking in Web AR: what it is and how to use it. Onirix. URL: <https://www.onirix.com/learn-about-ar/image-tracking-web-ar/> (дата звернення: 21.05.2023).
15. Future of shoulder replacement surgery is here. UT Southwestern Medical Center. URL: <https://utswmed.org/physician-resources/physician-update/future-of-shoulder-replacement-surgery-is-here/> (дата звернення: 21.05.2023).
16. Rossi J. ARKit 2: the good, the bad and the ugly. OrangeLoops. URL: <https://orangeloops.com/2019/04/arkit-2-the-good-the-bad-and-the-ugly/> (дата звернення: 21.05.2023).
17. Rossi J. What's New in ARKit 3?. OrangeLoops. URL: <https://orangeloops.com/2019/09/whats-new-in-arkit-3/> (дата звернення: 21.05.2023).
18. Rossi J. What's New in ARKit 4?. OrangeLoops. URL: <https://orangeloops.com/2020/06/whats-new-in-arkit-4/> (дата звернення: 21.05.2023).
19. Rossi J. What's New in ARKit 5?. OrangeLoops. URL: <https://orangeloops.com/2021/06/whats-new-in-arkit-5/> (дата звернення: 21.05.2023).

20. Releases google-ar/arcore-android-sdk. GitHub. URL: <https://github.com/google-ar/arcore-android-sdk/releases> (дата звернення: 21.05.2023).
21. What platforms are supported by Unity?. Unity. URL: <https://support.unity.com/hc/en-us/articles/206336795-What-platforms-are-supported-by-Unity-> (дата звернення: 21.05.2023).
22. Government & aerospace. Unity. URL: <https://unity.com/solutions/government-aerospace> (дата звернення: 21.05.2023).
23. Compare plans. Unity. URL: <https://unity.com/compare-plans> (дата звернення: 21.05.2023).
24. AR foundation. Unity. URL: <https://docs.unity3d.com/Packages/com.unity.xr.foundation@5.0/manual/index.html> (дата звернення: 21.05.2023).
25. BBC News. Unreal games engine licensed to FBI and other US agencies. BBC News. URL: <https://www.bbc.com/news/technology-17535906> (дата звернення: 21.05.2023).
26. Unreal Engine (UE5) licensing options. Unreal Engine. URL: <https://www.unrealengine.com/en-US/license> (дата звернення: 21.05.2023).
27. Augmented reality overview. Unreal Engine. URL: [https://docs.unrealengine.com/4.26/en-US/SharingAndReleasing/XRDevelopment/AR/HandheldAR/AROverview/#:~:text=Augmented%20reality%20\(AR\)%20is%20a,iOS%20and%20Android%20handheld%20platforms](https://docs.unrealengine.com/4.26/en-US/SharingAndReleasing/XRDevelopment/AR/HandheldAR/AROverview/#:~:text=Augmented%20reality%20(AR)%20is%20a,iOS%20and%20Android%20handheld%20platforms) (дата звернення: 21.05.2023).
28. What is Amazon GameSparks? - GameSparks. Amazon. URL: https://docs.aws.amazon.com/gamesparks/latest/dg/welcome.html#_features_of_amazon_gamesparks (дата звернення: 21.05.2023).
29. Amazon GameSparks Pricing. Amazon. URL: <https://aws.amazon.com/gamesparks/pricing/> (дата звернення: 21.05.2023).
30. Global cross platform realtime multiplayer game framework. Photon Engine. URL: <https://www.photonengine.com/realtime> (дата звернення: 21.05.2023).

- 31.Susic P. 56+ mobile gaming statistics (2023). HeadphonesAddict. URL: <https://headphonesaddict.com/mobile-gaming-statistics/> (дата звернення: 21.05.2023).
- 32.HoloLens 2. Microsoft. URL: <https://www.microsoft.com/en-ca/hololens/buy> (дата звернення: 21.05.2023).
- 33.SkyView® lite. App Store. URL: <https://apps.apple.com/us/app/skyview-lite/id413936865> (дата звернення: 21.05.2023).
- 34.Dive into Epic's announcements from GDC 2023. Unreal Engine. URL: <https://www.unrealengine.com/en-US/blog/dive-into-epic-s-announcements-from-gdc-2023> (дата звернення: 21.05.2023).
- 35.Valentine R. Epic Games announces Unreal Engine 5 with first PS5 footage. GamesIndustry. URL: <https://www.gamesindustry.biz/epic-games-announces-unreal-engine-5-with-first-ps5-footage> (дата звернення: 21.05.2023).
- 36.Orland K. How Epic got such amazing Unreal Engine 5 results on next-gen consoles. Ars Technica. URL: <https://arstechnica.com/gaming/2020/05/behind-the-scenes-of-that-incredible-unreal-engine-5-tech-demo/> (дата звернення: 21.05.2023).
- 37.What are games built with and what technologies do they use?. SteamDB. URL: <https://steamdb.info/tech/> (дата звернення: 21.05.2023).
- 38.Smykil J. Apple Design Award winners announced. Ars Technica. URL: <https://arstechnica.com/gadgets/2006/08/4937/> (дата звернення: 21.05.2023).
- 39.2021 gaming report. Unity. URL: <https://create.unity.com/2021-game-report> (дата звернення: 21.05.2023).
- 40.Fanouraki E. SlashData. Game engines & their use in game development. URL: <https://www.slashdata.co/blog/did-you-know-that-60-of-game-developers-use-game-engines> (дата звернення: 21.05.2023).
- 41.Nowacki P., Woda M. Capabilities of ARCore and ARKit Platforms for AR/VR Applications. 12 трав. 2019 р. URL: https://link.springer.com/chapter/10.1007/978-3-030-19501-4_36 (дата звернення: 21.05.2023).

42. Wohlgenannt I., Simons A., Stieglitz S. Virtual reality. Business & information systems engineering. 2020. URL: <https://link.springer.com/article/10.1007/s12599-020-00658-9> (дата звернення: 21.05.2023).
43. Billinghurst M. Augmented reality in education. New horizons for learning. 2002.
44. An overview of augmented reality / F. Arena et al. Computers. 2022. URL: <https://doi.org/10.3390/computers11020028> (date of access: 22.05.2023).