

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри,
канд. техн. наук, доцент
_____ Я. М. Крайник
« __ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Галузь знань: 12 Інформаційні технології
Спеціальність: 123 Комп'ютерна інженерія
Тема: **Програмно-апаратний комплекс для організації командних ігор з елементами доповненої реальності**
Шифр: 123 – КР.ПЗ.01 – 405.21810520

Виконав:
студент 4 курсу, групи 405,
Спеціальності
123 Комп'ютерна інженерія
О.С.Плохой

Керівник:
ст. викладач
І. С. Бурлаченко

Миколаїв 2022

ЗАВДАННЯ

на виконання бакалаврської роботи

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 1

ця сторінка після друку буде замінена

ЗАВДАННЯ

на виконання бакалаврської роботи

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 2

ця сторінка після друку буде замінена

АНОТАЦІЯ

1 сторінка !!!!

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 1

ця сторінка після друку буде замінена

ABSTRACT

1 сторінка !!!!

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 2

ця сторінка після друку буде замінена

ЗМІСТ

ЗМІСТ	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ПАТЕНТНОЇ ІНФОРМАЦІЇ	11
1.1 Аналітичний огляд фізичних процесів для доповненої реальності в іграх	11
1.2 Типи доповненої реальності	13
1.3 Застосування AR на основі маркерів	18
1.4 AR у іграх	24
Висновки до розділу 1	26
РОЗДІЛ 2. ОПИС ТЕХНІЧНИХ ХАРАКТЕРИСТИК	27
2.1 Конкурентність на ринку доповненої реальності в командних іграх	27
2.2 Конкурентні мікроконтролери для апаратної платформи	32
2.2.1 Мікроконтролер Orange Pi One	32
2.2.2 Міні-комп'ютер Raspberry Pi 1B+	35
2.2.3 Міні-комп'ютер ASUS Tinker Board	39
2.2.4 Міні-комп'ютер Khadas Vim1	43
2.2.5 Міні-комп'ютер PINE64 Rock64	45
2.2.6 Порівняльна характеристика розглянутих моделей міні-комп'ютерів	47
2.3 Камери для доповненої реальності у командних іграх	48
2.3.1 Веб-камера Promate ProCam-2 FullHD USB Black	51
2.3.2 Веб-камера VERTUX VERTUCAM-4K UHD USB BLACK	52
2.3.3 Веб-камера VERTUX ODIN-4K UHD з LED підсвіткою USB BLACK	54

2.3.4 Порівняльна характеристика розглянутих веб-камер для доповненої реальності	55
Висновки до розділу 2	56
РОЗДІЛ 3. РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ	57
3.1 Фреймворк ARToolKit	57
3.1.1 Процес калібрування маркерів за допомогою ARToolKit	61
3.2 Налаштування апаратної платформи	62
3.3 Розробка програмної частини	65
3.3.1 Модулі доповненої реальності	67
3.3.2 Використанні змінних та функції	68
3.4 Алгоритм роботи застосунку	70
3.5 Тестування програмної платформи	71
Висновки до розділу 3	73
ВИСНОВКИ	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	75
ДОДАТОК А ПРОГРАМНИЙ КОД ДЛЯ КЕРУВАННЯ	78
ДОДАТОК Б ПРОГРАМНИЙ КОД	85
ДОДАТОК В БЛОК-СХЕМА АЛГОРИТМУ	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

3D	–	3-dimensional
AR	–	Augmented reality
2D	–	2-dimensional
GPS	–	Global Positioning System
BIM	–	Building Information Model
GPU	–	Graphics Processing Unit
IoT	–	Internet of Things
OpenCV	–	Open Source Computer Vision Library
PTAM	–	Parallel Tracking And Mapping
UI	–	User Interface
URL	–	Universal Resource Locator

ВСТУП

Технологія доповненої реальності (Augmented reality) — це технологія, яка поєднує віртуальну інформацію з реальним світом. Технічні засоби, які вона використовує, включають мультимедіа, 3D-моделювання, відстеження в реальному часі, інтелектуальна взаємодія, розпізнавання тощо. Її принцип полягає в застосуванні комп'ютером віртуальної інформації, такої як текст, зображення, 3D-моделі, музика, відео тощо, у реальний світ після моделювання. Таким чином, два види інформації доповнюють один одного [1].

Останніми роками все більше міжнародно-відомих науково-дослідних установ, університетів і підприємства інвестували в дослідження AR, опублікували багато статей та результатів наукових досліджень. Ці результати демонструють доцільність та інноваційність AR як взаємодії людини та комп'ютерних технологій. З покращенням обчислювальної потужності комп'ютерного програмного та апаратного забезпечення AR поступово перейшов від теоретичної дослідницької стадії лабораторії до стадії масового та промислового виробництва додатків, і як міст між цифровим світом і реальним світом, він надає людям новий спосіб розпізнавати і переживати речі навколо. Крім того, він увійшов до першої десятки у списку найбільш перспективних технологій в майбутньому, авторитетних організацій, таких як American Times [2]. В останні роки хмарні обчислення стали центром дослідження в Інтернет-сфері. Інтеграція хмарних обчислень може перенести сховище та складні обчислення AR від клієнта до середовища обслуговування хмарних обчислень, надати можливий шаблону розподіленої архітектури для обчислення складних інтерактивних фізичних ефектів [3].

Мета: розробити програмне забезпечення з елементами доповненої реальності для командних ігор.

Об'єкт: веб-орієнтована система командних ігрових змагань у доповненій реальності.

Предмет: методи та засоби реалізації програмно-апаратного комплексу доповненої реальності для використання у командних іграх

Для досягнення поставленої мети необхідно вирішити такі **завдання:**

- проаналізувати існуючі технології та фреймворки для роботи з доповненою реальністю;
- дослідити використання маркерів доповненої реальності у AR фреймворках;
- розробити веб-інтерфейс веб-орієнтованої системи для командних ігор;
- обрати апаратну платформу для розгортання веб-серверів з іграми у доповненій реальності;
- визначити достатні характеристики веб-камери для ігор доповненої реальності;
- розробити питання з охорони праці та безпеки життєдіяльності

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ПАТЕНТНОЇ ІНФОРМАЦІЇ

1.1 Аналітичний огляд фізичних процесів для доповненої реальності в іграх

Ігри з доповненою реальністю (AR-ігри) — це інтеграція ігрового візуального та аудіоконтенту з середовищем користувача в режимі реального часу. На відміну від ігор у віртуальній реальності, які часто потребують окремої кімнати або обмеженої зони для створення занурювального середовища, ігри доповненої реальності використовують існуюче середовище та створюють у ньому ігрове поле. Хоча для ігор віртуальної реальності потрібні спеціальні гарнітури VR, лише деякі системи доповненої реальності використовують їх. Ігри AR зазвичай грають на таких пристроях, як смартфони, планшети та портативні ігрові системи.

Гра з доповненою реальністю часто накладає попередньо створене середовище поверх реального середовища користувача. Сама гра може бути такою ж простою, як гра у віртуальні шашки, в яку грають на поверхні столу. Більш просунуті ігри AR можуть насправді створювати середовище з оточення користувача. Така гра може включати, наприклад, ігрових персонажів, які перелазять з журнальних столиків на дивани на віртуальних мостах. Створення середовища є трудомістким завданням у створенні ігор, і існує постійний попит на нові декорації, тому що як тільки користувач повністю дослідив середовище, він хоче перейти до іншого. Ігри AR розширюють ігрове поле, використовуючи переваги різноманітності реального середовища, щоб ігри залишалися цікавими.

Ігрова індустрія швидко інтегрувала технологію AR і є одним із найактивніших секторів з точки зору розвитку AR. Вихід гри Pokemon GO у 2016 році став переломним моментом, який продемонстрував потенціал AR в ігровій індустрії. Гра швидко стала глобальним явищем, і мільйони людей

грали в неї на своїх смартфонах. Успіх перших мобільних ігор AR привів до відновлення інтересу до технології AR з боку ігрової індустрії. Кілька великих ігрових компаній, включаючи Nintendo, Microsoft і Sony, вкладають значні кошти в технології AR.

Інші ігри наслідували цей приклад, і зараз є багато популярних ігор AR, включаючи Jurassic World Alive, Ghostbusters World та The Walking Dead: Our World. Ці ігри використовують AR, щоб перенести віртуальних істот і персонажів у реальний світ, дозволяючи гравцям взаємодіяти з ними різними способами. AR також використовується для покращення традиційних відеоігор. Наприклад, консоль Xbox One має додаток AR під назвою Kinect Sports Rivals, який дозволяє гравцям змагатися в багатьох видах спорту. А на консолі PlayStation 4 є гра AR під назвою The Playroom, яка використовує камеру PlayStation Camera для створення інтерактивного віртуального середовища для вивчення гравців.

Кілька ігор AR у розробці розроблені для гарнітур AR, таких як Microsoft HoloLens і Magic Leap One. Ці ігри включають Project X-Ray, захоплюючий шутер від першої особи, розроблений Microsoft, і Scraper: First Strike, пригодницьку гру, розроблену Labrodex, Inc. Очевидно, що AR має значний вплив на ігрову індустрію і буде назавжди змінити те, як ми граємо у відеоігри.

Майбутнє AR в ігровій індустрії, безсумнівно, яскраве. Успіх перших мобільних ігор AR довів, який попит на технологію AR серед звичайних геймерів. Ігрова індустрія інвестує значні кошти в технології AR, і кілька великих ігрових компаній розробляють продукти AR. Розвиток AR в ігровій індустрії призведе до нових та інноваційних ігрових можливостей.

1.2 Типи доповненої реальності

Наразі AR можна генерувати за допомогою додатків на звичайних пристроях, таких як смартфони, планшети, HoloLens тощо. Потроху ця технологія шукає нові сектори додатків, щоб покращити свої робочі процеси, особливо з огляду на прихід 5G.

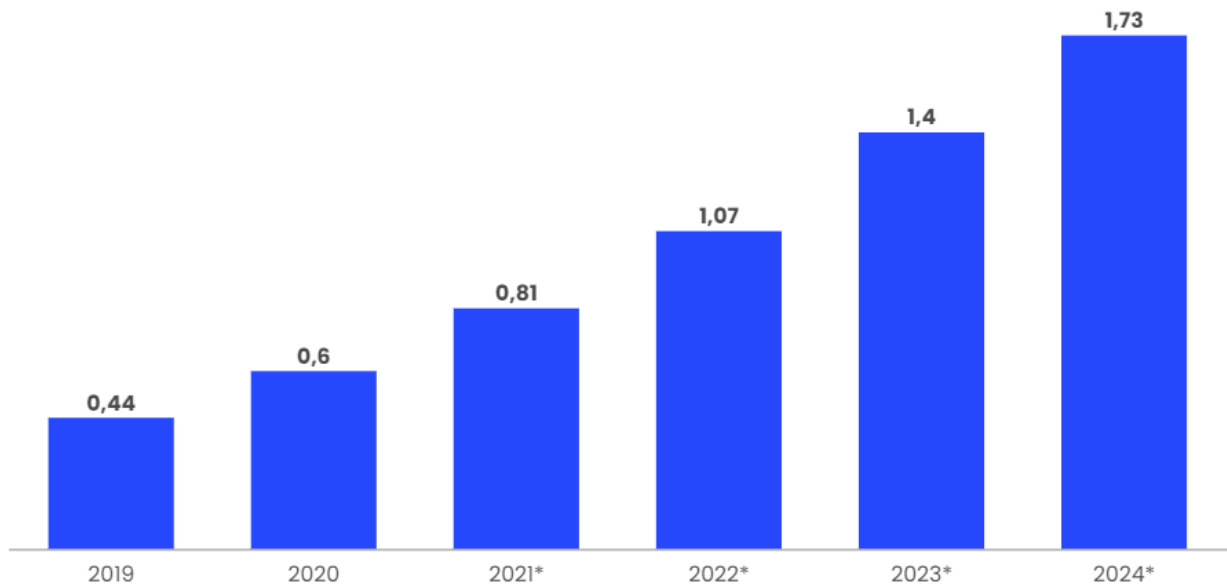


Рисунок 1.1 - Кількість активних мобільних доповненої реальності користувачів у всьому світі з 2019 по 2024

Google, Facebook і Amazon є одними з гігантів, які використовують програмне забезпечення AR для оптимізації своєї продуктивності. Наприклад, Instagram або Snapchat створюють веселі фільтри для своїх користувачів.

Таким чином, доповнена реальність визначається як змінена форма реальності, в якій комп'ютерний контент накладається на реальний світ користувача, що дозволяє додавати цифрові активи до їхнього фізичного середовища. AR також має відповідати трьом основним характеристикам:

- Поєднання реального та віртуального світу.
- Взаємодія в режимі реального часу.
- Точна 3D реєстрація реальних і віртуальних об'єктів.

Однак існують різні типи AR, і їх відмінності слід знати, оскільки кожен з них буде більш підходящим для певного використання, хоча всі вони мають спільні риси. Таким чином, основна диференціація буде між:

- AR на основі маркерів.
- AR без маркерів:
- AR на основі розташування.
- AR на основі проекції.
- Накладання AR.
- AR на основі контуру.

1.2.1 AR на основі маркерів

Додатки AR на основі маркерів використовують цільові зображення (маркери) для розташування об'єктів у заданому просторі. Ці маркери визначають, де програма розміщуватиме 3D цифровий вміст у полі зору користувача. Технології AR на ранній стадії були засновані на маркерах.

Іншими словами, ці програми пов'язані з певним маркером візерунка фізичного зображення в реальному середовищі, щоб накласти на нього 3D віртуальний об'єкт. Таким чином, камери повинні безперервно сканувати вхід і розміщувати маркер для розпізнавання малюнка, щоб створити його геометрію. Якщо камера не сфокусована належним чином, віртуальний об'єкт не відобразиться.

Отже, система розпізнавання зображень на основі маркерів вимагає, серед іншого, кількох модулів, таких як камера, захоплення зображення, обробка зображень та відстеження маркерів. Як правило, це проста і недорога система, яку можна впровадити у фільтри через спеціальну програму для розпізнавання конкретних шаблонів за допомогою камери.

Прикладом такого типу доповненої реальності є Instagram та Snapchat через фільтри та ігри. Таким чином, цей тип AR вже введений у повсякденне життя людей, оскільки це рутинна соціальна діяльність.

На відміну від цього, AR без маркерів дозволяє розташовувати віртуальні 3D-об'єкти в середовищі реальних зображень шляхом вивчення функцій, наявних у даних, у реальному часі. Цей тип вказівок покладається на апаратне забезпечення будь-якого смартфона, будь то камера, GPS або акселерометр, серед іншого, а програмне забезпечення доповненої реальності завершує роботу.

З цією моделлю немає потреби в системі відстеження об'єктів через останні технологічні досягнення в камерах, датчиках і алгоритмах штучного інтелекту. Таким чином, він працює з цифровими даними, отриманими цими датчиками, здатними записувати фізичний простір в режимі реального часу.

Насамперед, аналіз без маркерів використовує одночасну локалізацію та відображення (SLAM) для сканування середовища та створення відповідних карт, на яких розміщувати віртуальні об'єкти. Відстеження зображень SLAM без маркерів сканує середовище та створює карти місця розміщення віртуальних об'єктів у 3D, навіть якщо об'єкти не знаходяться в полі зору користувача, не рухаються, коли користувач рухається, і користувачеві не потрібно сканувати нові зображення .

Таким чином, ця технологія здатна виявляти об'єкти або характерні точки в сцені без попереднього знання навколишнього середовища, наприклад, вона може ідентифікувати стіни або точки перетину. Це технологія, яка характеризується асоціацією з візуальним ефектом поєднання комп'ютерної графіки з реальними зображеннями.

Перші системи, що використовували цей тип AR, використовували місцезнаходження та апаратні служби пристрою для взаємодії з ресурсами, наданими програмним забезпеченням AR, таким чином, щоб було визначено місцезнаходження та орієнтація користувача в просторі, де він перебував.

Ще одна особливість цього типу доповненої реальності полягає в тому, що користувачі можуть збільшити середній діапазон руху, відчуваючи досвід. Apple ARKit і Google ARCore SDK зробили без маркерну AR

доступною на розумних пристроях. Наразі безмаркерний AR є кращим методом розпізнавання зображень для додатків, що використовують цю технологію.

1.2.2 AR на основі розташування

Доповнена реальність без маркерів на основі розташування спрямована на злиття 3D віртуальних об'єктів у фізичному просторі, де перебуває користувач. Очевидно, що ця технологія використовує місцезнаходження та датчики смарт-пристрою, щоб розташувати віртуальний об'єкт у бажаному місці чи точці інтересу.

Найбільш репрезентативним прикладом цього типу доповненої реальності є гра для смартфонів Pokémon GO, яка використовує AR без маркерів, що базується на місцезнаходженнях, миттєво оживляючи середовище користувача залежно від того, куди він дивиться.

Ця AR пов'язує віртуальне зображення з певним місцем, зчитуючи дані в режимі реального часу за допомогою камери, GPS, компаса та акселерометра. Крім того, оскільки він заснований на безмаркерній доповненій реальності, для його роботи не потрібна доріжка зображення, оскільки вона здатна передбачити підхід користувача, щоб узгодити дані в реальному часі з місцем розташування користувача.

Крім того, ця типологія дає можливість додавати інтерактивний і корисний цифровий контент до регіонів, що цікавлять вас, що дуже корисно для мандрівників у певній місцевості, допомагаючи зрозуміти навколишнє середовище за допомогою 3D віртуальних об'єктів або відео.

1.2.3 AR на основі проєкції

Ця методологія використовується для доставки цифрових даних у стаціонарному контексті, тобто проектна AR зосереджена на відтворенні віртуальних 3D-об'єктів у фізичному просторі користувача.

Таким чином, AR дозволяє користувачеві вільно переміщатися по оточенню певної області, де розміщені стаціонарний проектор і камера стеження. Основне використання цієї технології — створення ілюзій щодо глибини, положення та орієнтації об'єкта шляхом проектування штучного світла на реальні плоскі поверхні.

Наприклад, AR на основі проекції підходить для спрощення складних завдань у бізнесі чи промисловості, виключаючи комп'ютери, оскільки інструкції можуть бути розміщені в заданому просторі. Крім того, ця технологія здатна забезпечити зворотний зв'язок для оптимізації процесів цифрової ідентифікації для виробничих циклів.

Накладання AR, як правило, ця AR використовується для заміни оригінального вигляду об'єкта оновленим віртуальним зображенням цього об'єкта для людського ока. Overlay AR забезпечує кілька переглядів цільового об'єкта з можливістю відображення додаткової відповідної інформації про цей об'єкт.

AR на основі контуру, по суті, ця технологія, завдяки використанню спеціальних камер, використовується для людських очей для окреслення конкретних об'єктів лініями для полегшення певних ситуацій. Наприклад, його можна використовувати для автомобільних навігаційних систем, щоб забезпечити безпечне водіння в умовах поганої видимості.

Наразі кількість додатків доповненої реальності досить велика і продовжує зростати, серед цих додатків є роздрібна хірургія для охорони здоров'я, налаштовані автомобілі для просування автономного водіння, це можна побачити в школах для навчання, можна робити негайні переклади, розумні сканери обличчя, військові додатки тощо.

Однак один і той самий тип доповненої реальності не може бути застосовано для кожного із згаданих додатків, оскільки потреби в кожному секторі різні. В основному, існує класифікація на дві основні групи: AR на основі маркерів і AR без маркерів, причому перша є найбільш примітивною версією цієї технології. У свою чергу, в межах безмаркерної AR є 4 типи: AR на основі розташування, AR на основі проекції, AR накладання та контурна AR. Як ми бачили, кожна типологія має застосування та заснована на іншому технологічному принципі.

1.3 Застосування AR на основі маркерів

Існує багато додатків щодо використання AR для підтримки виробництва та суміжних областей, починаючи від проектування, моделювання, тестування, ремонту, та обслуговування. Управління за допомогою технології AR на основі маркерів було розроблено та розроблено з використанням ігрового движка Unity 3D та бібліотеки OpenCV. Він був розроблений для використання багатьма пристроями, що працюють на різних операційних системах, таких як Windows, OSX, iOS та Android.

За останні кілька років з'явилися деякі значущі ознаки прогресу в галузі інформаційних технологій, особливо у віртуальній реальності (VR) і доповненій реальності (AR). VR є технологія, яка дозволяє користувачеві повністю зануритися в віртуальний світ, тоді як AR інтегрує віртуальний світ, створений цифровий комп'ютер з реальним середовищем. Незважаючи на те, що AR розглядається як підмножина VR, але з точки зору додатків, вони істотно відрізняються. Можуть бути як AR, так і VR застосовується в багатьох галузях, але у виробництві та пов'язаних промисловість серйозно сприйняла цю технологію. Оскільки AR угоди з віртуальним і реальним словом, він підходить для космосу управління, що вимагало моделювання

розрахунку форми і розмір віртуального простору, спроектованого поверх реального світ.

Існує дві категорії AR, тобто AR на основі маркерів і безмаркерний AR. Відмінності між цими двома залежать від джерело інформації про позицію цифрового контенту в поле зору користувача. У випадку AR на основі маркерів, система повинна знати, де знаходиться користувач і яке положення користувач хоче зосередитися. Одним із рішень є додавання виявленого заздалегідь визначений знак або «маркер» у середовищі та використанні методики комп'ютерного зору для його виявлення, що називається «відстеження». Маркер може бути у вигляді 2D картинки, QR код або тривимірний об'єкт. Функція маркера полягає в ідентифікації положення віртуальних об'єктів, що відображаються. В випадку безмаркерної AR користувач може розмістити віртуальний об'єкт будь-яке положення в межах огляду камери за допомогою а поєднання GPS і датчика компаса на смартфон, як ми бачили у випадку з Pokémon Go [4].

Сьогодні з'явилася нова платформа для додатків AR без маркерів наприклад ARKit для пристроїв iOS і ARCore для Android пристрої, які спрощують розробку додатків AR. Ці двоє платформи використовують дані з датчика руху та камери пристрою визначити просторовий рух по шести осях. Це дозволяє віртуальні об'єкти залишаються точно розташованими в реальному світ. За допомогою функції виявлення площини можна використовувати віртуальні предмети розміщені на поверхнях реалістично [5]. Ці двоє платформи здавалися успішними з точки зору розваг перспективи, але для вимірювання простору до них ще далеко досконалий [6]. Проблеми полягають у точності визначення де підлога. Технічно система може виявити нерівномірність кольорова підлога, але не однорідна біла підлога, тому не можна розмістити віртуальну модель на призначеному місці. Оскільки керує простір потребує відстані від стіни до стіни. Закріплення положення на кожному кінці необхідно з урахуванням позначок AR є

відповідним. Вимірювальні системи AR можуть не тільки вимірювати відстань але також створюють форму повного розміру кімнати або будь-який близький простір. Система також може вимірювати і створювати об'єкт окремої форми, наприклад посилку або гранулу, швидко та точно, що робить його ідеальним для складу управління також. У цьому дослідженні технологія вимірювання AR, який є тривимірним, безконтактним вимірюванням були використовується для підтримки управління простором у виробництві і логістична платформа.

Різні напрямки досліджень використання AR на основі маркерів застосування у виробництві було здійснено в останні кілька років [7]. Michalos G та ін. розроблено на основі AR додаток для підтримки інтерактивної співпраці людина-робот-автомобільне середовище. Їхні цілі – покращити безпеку людей та підвищення їх продуктивності в промисловому середовищі, де роботи співіснують з людьми. Результати вказують на те, що цей підхід може значно покращити умови роботи оператора та їх інтеграцію в процес складання [8]. Азіз Ф.А. та ін. [9] розробив AR платформу для допомоги технікам автомобільної промисловості в процесі перевірки та технічного обслуговування.

Дослідження щодо використання AR для вимірювання простору починають зростати завдяки розвитку мобільного обладнання та додатків програмного забезпечення. Дапonte та ін. [11] представлено опитування з програми, що використовують AR як інтерфейс користувача система вимірювання. Тенденції майбутніх досліджень орієнтовані на розробку процедур калібрування для систем відстеження. Bergquist R. et al. [12] застосував пристрій ARKit для вимірювання площ поверхонь шляхом розміщення якоря для позначення площі, яку потрібно виміряти. Незважаючи на те, що AR інструмент не забезпечує такої ж точності, як ручні вимірювання, але все ще достатньо в межах, якщо оцінка прийнятна. Whitlock M. та ін. [13] досліджує AR взаємодії на відстані, вимірюючи, як

програми можуть підтримувати плавний, ефективний та інтуїтивно зрозумілий інтерактивний досвід у доповненої реальності у масштабі кімнати.

Багато компаній, що займаються програмним забезпеченням, також починають розробляти прикладні програми для перетворення мобільних пристроїв у вимірювальній системі [14, а].

Зазвичай увага приділяється дослідженню застосування AR відстеження положення маркера, а потім відображення віртуальної інформації у визначеному місці. Відстеження – це техніка яка допомагає камері знайти орієнтацію та положення маркера, що стосується реального світу. У цьому дослідженні замість використання інформації для відображення віртуальних об'єктів, інформацію про маркери, тобто їх положення (x, y, z) та їх вісь обертання, була використана для розрахунку простору і форма кімнати або контейнера. З цією концепцією об'єкт управління ефективніше. Доступно багато типів маркерів, але потрібно вибирати підходящий тип маркерів залежно від системних та користувацьких запитів [16].

У цьому випадку маркер ArUco був обраний, оскільки вони є відповідні нашим вимогам з точки зору надійності, міцності і універсальності. Крім того, їх визнання швидкий, з низьким рівнем неоднозначності в приміщеннях.

Крім того, маркер було включено в останню версію OpenCV [17]. OpenCV (комп'ютер з відкритим кодом) — це бібліотека функцій програмування, переважно спрямованих на комп'ютерний зір в режимі реального часу. Зазвичай це маркер ArUco представлений чорно-білим квадратним візерунком 7×7 блокувати. Чорна облямівка має товщину 1 блок, тому внутрішній шаблон являє собою підматрицю 6×6 блоків. Під час відстеження, цей шаблон буде представляти тип маркера. І розташування

внутрішнього блоку матриці буде представляти розташування та орієнтація камери в режимі реального часу.

Етап відстеження маркера можна розділити на три кроки:

1. Виявлення кандидатів на маркери: на цьому етапі інформація з камери буде проаналізована, щоб перевірити, чи є квадратний маркер чи ні, за допомогою технології адаптивного порогу, а потім відфільтровано іншу інформацію. окрім маркера із зображення за допомогою контурної фільтрації.

2. Аналіз маркера. На цьому кроці для поділу чорного і білого біта використовувалася порогова техніка Оцу. Після розрахунку маркер може визначити, чи є його ідентифікатор у базі даних чи ні.

3. Техніка оцінки положення. Методика, що використовується для оцінки положення маркера шляхом поєднання координат творця в реальному та віртуальному світі. Процес дасть положення та поворот маркера.

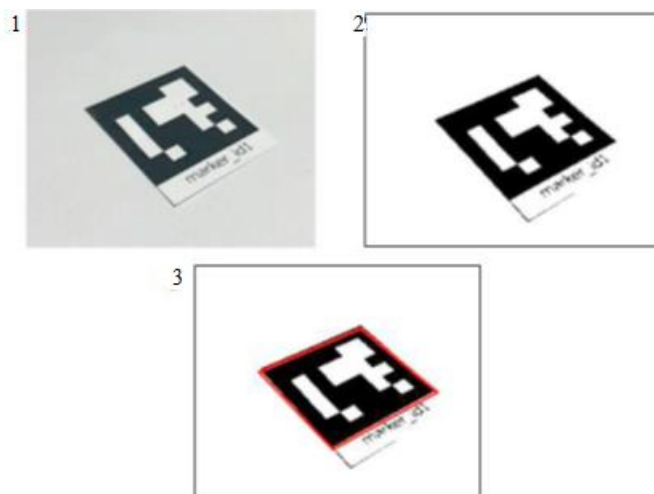


Рисунок 1.2. (1) Зображення з камери; (2) після адаптивної порогової та контурної фільтрації; (3) оцінка пози.

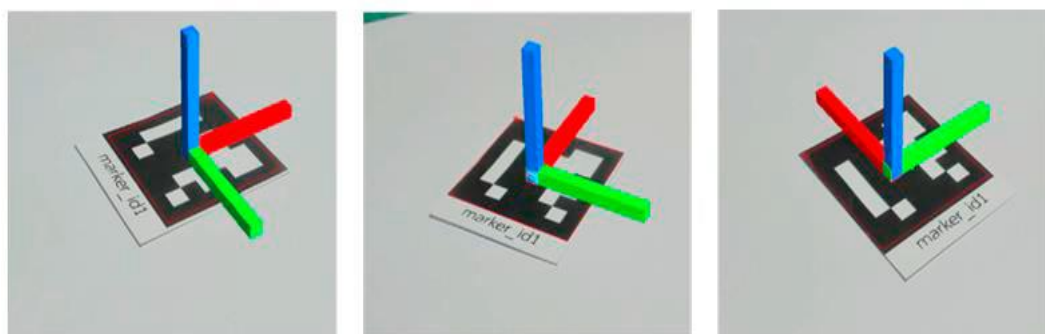


Рисунок. 1.3. Моделювання центру повороту на поворотному маркері.

Оскільки положення (x, y, z) маркера знаходиться у віртуальному просторі, тож опорної точки немає, то відстань не може бути перетворена в реальний простір. Щоб вирішити цю проблему, вирішено використовувати маркер розміром 10×10 см., який буде системою відліку.

Оскільки потрібно знайти відстань між двома позиціями, тобто двома гравцями або ракетками, то маркери двох позицій потрібно зібрати або зафіксувати на одному знімку. Це підтвердить, що обидві позиції знаходяться в одній системі відліку. Під час кожного захоплення дані, отримані від маркера, такі як ідентифікатор, положення та кут повороту, зберігатимуться в базі даних для подальшого аналізу. У випадку, де два маркери з протилежних сторін можуть бути захоплені одночасно, дані з обох маркерів можна зберігати в базі даних безпосередньо без будь-яких подальших змін. Якщо дистанція між гравцями занадто велика, то між ними потрібно розмістити ще один маркер. У цьому випадку нам доведеться захопити два або більше маркерів. Спочатку знімаються перші два маркери, а потім захоплюються ще два, при цьому один маркер з першого зйомки повинен бути в кадрі. Загальний маркер буде використовуватися як еталон або з'єднувач між двома записами.

Положення маркера у віртуальному просторі (x, y, z) можна представити у вигляді матриці так:

$$V_{(p)} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \quad (1)$$

Можливість використання технології AR на основі маркерів полягає в тому, щоб використовувати цю технологію для управління використання приміщень на фабриках або виробництвах. Алгоритм вимірювання відстані за допомогою маркера AR може дати високу точність вимірювань. Ця техніка може застосовуватися в інших цілях, наприклад для обчислення обсягів, який займає продукція або товар на складі.

1.4 AR у іграх

Ігри – це форма розваги, яка приносить задоволення та задоволення». Із розвитком комп'ютерів від першого покоління до теперішнього часу комп'ютерні ігри розвивалися поряд із введенням команди як за допомогою клавіатури (як засобу взаємодії), так і маханням рукою за допомогою детектора руху.

Ігрова індустрія охопила як технології доповненої реальності, так і технології віртуальної реальності, а такі назви, як Pokémon Go від Niantic, допомогли розкрити поняття ігор з доповненою реальністю для широкої публіки.

Гравці в Pokémon Go повинні подорожувати по реальному світу, щоб зібрати різноманітність покемонів. Гравець може відвідувати спортивні зали Pokéstop і Pokémon, щоб отримати речі, а також змагатися з цими гравцями. Коли покемон виявляється, гравець має можливість захопити його. Під час взаємодії користувач має можливість активації доповненої реальності (AR), яка перетворює середовище на зображення, зроблене камерою.

Ігри доповненої реальності можна розглядати як міст між комп'ютерними іграми та іграми реального світу. Це новий жанр ігор, у якому генеровані комп'ютером сенсорні дані використовуються для створення гри, створеної у фізичному світі. Нову можливість, яку надала AR для створення ігор, які використовують переваги як комп'ютерних, так і закадрових класичних ігор.

Ігри з доповненою реальністю також називають доповненими іграми. Доповнені ігри дотримуються тих же принципів дизайну, що й класичні комп'ютерні ігри, але через додатковий вимір змішування реальності з віртуальністю є нові аспекти, які необхідно враховувати. Виникають питання, як поєднати елементи класичних комп'ютерних ігор і фізичний світ, щоб створити задовільний і розважальний досвід ігор AR.

Одним із ключових елементів розширеної гри є ігровий простір. Тому важливо врахувати необхідний простір для гри. Ігрове поле може бути обмежене на основі використовуваної технології AR (наприклад, якщо для виявлення руху використовуються стаціонарні камери, гравець повинен знаходитися в полі видимості налаштувань системи). Або ігровий простір може бути обмежено через характер гри, як у випадку з додатками на основі GPS, наприклад, ігровий простір має бути на відкритому повітрі, де можна легко отримати сигнал GPS.

Введення та зворотній зв'язок через інтерфейс є ще одним важливим аспектом ігрового дизайну в розширених іграх. Інтерфейс, через який користувач взаємодіє, може допомогти йому краще ставитися до світу гри. Наприклад, у військовій грі гравець може мати кращий досвід, якщо інтерфейс, який використовується для бою, представляє собою консоль у формі гармати. Робимо висновок, що інтерфейс має здатність впливати на емоційні переживання гравця.

Основними вхідними сигналами через інтерфейси є рух, звук і тактильні елементи (чутливі до дотику елементи інтерфейсу, такі як кнопка на сенсорному екрані). У відповідь система дає зворотний зв'язок у вигляді візуальних дисплеїв, звуку та вібрації. Ігри AR на смартфонах використовують усі апаратні функції телефону. У них не використовуються дисплеї або інші типи автономних консолей. Іншими словами, вони використовують лише вбудований GPS, магнітометр, акселерометр, камеру та сенсорний екран для входу та зворотний зв'язок. GPS, магнітометр і акселерометр використовуються для розрахунку положення, орієнтації та курсу. Камера надає відеовходи для розпізнавання об'єктів у випадку AR на основі маркерів і використовує фізичний світ як фон для накладання зображень. Сенсорний екран діє як монітор і інтерфейс для тактильних входів.

До дизайну ігор у додатках для смартфонів можна також підійти з точки зору орієнтованих на користувача міркувань. Ця точка зору враховує проблеми, з якими стикаються обмеження пристрою та технології. Деякі з проблем проектування полягають у затримці між відстеженням і відтворенням зображень, низькій роздільній здатності дисплея, що спричиняє труднощі з розпізнаванням об'єктів, поле зору, обмежене екраном інтерфейсу телефону, і шум відстеження від датчиків, таких як магнітометр.

Висновки до розділу 1

У розділі увагу було зосереджено на відстеженні об'єктів на основі маркерів. Також оглянуто альтернативні методи відстеження, котрі використовуються в різних типах додатків AR. Описана взаємодія поєднання реального та віртуального світу, а також способи взаємодії між реальними та віртуальними об'єктами.

Виявлено, що з існуючих технологій, найбільш ефективною для створення командної гри є використання маркерів у доповненій реальності.

На основі аналізу ігрового простору зроблено висновок, що інтерфейс має здатність впливати на емоційні переживання гравця

РОЗДІЛ 2. ОПИС ТЕХНІЧНИХ ХАРАКТЕРИСТИК

В якості апаратного комплексу було розглянуто багато мікроконтролерів та камер для реалізації доповненої реальності.

2.1 Конкурентність на ринку доповненої реальності в командних іграх

Зараз ринок керується технологічним прогресом у поєднанні зі зростанням кількості мобільних геймерів. Доповнена реальність, також відома як AR, — це інтеграція цифрової інформації з середовищем реального часу користувача. В іграх, на відміну від віртуальної реальності, доповнена реальність використовує існуюче середовище та покращує його за допомогою накладання окремих функцій. На додаток до цього, доповнена реальність допомагає створити для гравців графіку та звук за допомогою камери пристрою. Для ігор на смартфонах доповнена реальність стала важливим інструментом, оскільки дозволяє геймерам створювати власних персонажів, цілі, ландшафти.

Технологія доповненої реальності нова і має величезний потенціал. Протягом багатьох років різні великі виробники представили численні програми доповненої реальності, які представили геймерам нові способи взаємодії з реальним світом.

Постійне зростання кількості мобільних геймерів позитивно вплинуло на попит ігор доповненої реальності. Очікується, що ця тенденція збережеться протягом наступних п'яти років.

Китай, США та Японія наразі є основними споживачами на цьому ринку.

Початкова вартість отримання якісного ігрового обладнання доповненої реальності висока, що ускладнює звичайним споживачам можливість дозволити собі консоль.

Компанії використовують додатки на основі AR для функцій складання, виробничих функцій, функцій виробничої лінії та аналізу завдань та пошуку рішень. Таким чином, зростаючий попит на дистанційну допомогу та взаємодію з боку галузей, які працюють в області оптимізації та управління робочими процесами, також є хорошим пророком для зростання ринку.

Впровадження технології AR у маркетингу та рекламі для проведення віртуальних подій, запуску продуктів, віртуальних виставок та онлайн-реклами набирає обертів після спалаху пандемії COVID-19. Наприклад, смартфон OnePlus Nord був запущений в липні 2020 року на платформі AR Vիրրar. Очікується, що дедалі більше впровадження технології AR серед представників галузі охорони здоров'я також відіграє важливу роль у зростанні ринку. Кілька постачальників рішень AR співпрацюють із закладами охорони здоров'я та пропонують свої інструменти доповненої реальності, щоб полегшити медичне обслуговування на основі AR. Наприклад, у лютому 2022 року компанія Altoida Inc. з доповненої реальності оголосила про партнерство з Click Therapeutics, щоб використовувати технологію AR, щоб краще зрозуміти базові показники пізнання та те, як це може потенційно вплинути на результати пацієнтів.

Очікується, що зростання використання технології AR у будівництві та архітектурних додатках відіграє вирішальну роль у зростанні ринку. Технологія AR потенційно може допомогти у створенні 3D-моделі креслення та допомогти з'єднати уяву з реальністю. Технологія AR також може використовуватися в освітній галузі, щоб покращити досвід навчання студентів. Технологія AR може допомогти забезпечити захоплюючий досвід використання 2D/3D-моделей, книг AR та ігор AR, а згодом допомогти у розвитку різних наборів навичок у студентів, тим самим стимулюючи зростання ринку доповненої реальності.

Технологія AR має потенціал для революції в навігації всередині приміщень. Наприклад, MobiDev, компанія з розробки програмного забезпечення, використовує технологію AR для трансформації внутрішньої навігації. Платформа ARcore компанії може допомогти рекомендувати найкращий можливий маршрут до потрібного пункту призначення та відобразити його на мобільному пристрої. Коли справа доходить до навігації на вулиці, технологія AR може допомогти мандрівникам у пошуку туристичних місць і відповідного житла за допомогою віртуальних турів на основі AR. Наприклад, Hub Hotels by Premier Inn використав технологію AR і перетворив свої номери на карту міста, щоб зорієнтувати туристів про туристичні пам'ятки поблизу. Таким чином, зростає використання технології AR в навігації, а згодом і зростаючий попит на рішення на основі AR з боку представників індустрії подорожей та туризму, як очікується, призведе до зростання ринку протягом прогнозованого періоду.

Технологія AR безперервно розвивається відповідно до все більшого поширення цієї технології в споживчих додатках. Як наслідок, попит на чіпсети AR також зростає. Зростаючий попит на чіпсети AR спонукає провідні фірми в галузі, такі як Qualcomm і NXP, впроваджувати інноваційні чіпсети з AR для задоволення мінливих потреб кінцевих користувачів. Наприклад, у січні 2022 року Qualcomm оголосила про партнерство з Microsoft для виробництва унікальних чіпів для окулярів доповненої реальності, які будуть сумісні для використання з додатками Metaverse.

У 2021 році сегмент апаратного забезпечення становив найбільшу частку доходу – 59,8% і, як очікується, продовжить домінувати на ринку в прогнозований період. Зростаюче поширення пристроїв AR у різних галузях і галузях промисловості, таких як освіта, охорона здоров'я, будівництво та роздрібна торгівля, дозволяє сегменту апаратного забезпечення домінувати на ринку. Наприклад, AccuVein Incorporated, компанія, що надає медичні рішення, використовує рішення на основі AR, щоб допомогти медичним

працівникам знайти вену пацієнта для внутрішньовенного лікування або взяти кров для аналізів крові, серед інших цілей. Зростаючий попит на обладнання, сумісне з доповненою реальністю, не показує ознак зменшення, оскільки все більше компаній активно інвестують у сучасні продукти та рішення AR. Наприклад, у січні 2020 року Hangzhou Tairuo Technology Company Limited оголосила про вдосконалений продукт під назвою Nreal Light AR окуляри, який використовує технологію AR для створення значно покращеного та захоплюючого досвіду, що поєднує віртуальну реальність та доповнену реальність.

Таблиця 2.1 - Обсяг звіту про ринок доповненої реальності

Вартість ринку в 2022 році	38,56 млрд дол
Прогноз доходів у 2030 році	597,54 млрд доларів США

Темп зростання	CAGR 40,9% з 2022 по 2030 рік
Базовий рік для оцінки	2021
Історичні дані	2018-2020
Прогнозний період	2022-2030
Кількісні одиниці	Дохід у мільярдах доларів США та CAGR з 2022 по 2030 рік
Висвітлення звіту	Прогноз доходу, рейтинг компанії, конкурентний ландшафт, фактори зростання та тенденції
Охоплені сегменти	Компонент, дисплей, додаток, регіон
Регіональний масштаб	Північна Америка; Європа; Азіатсько-Тихоокеанський регіон; Латинська Америка; Близький Схід і Африка
Розмах країн	Америка; Канада; Великобританія; Німеччина; Китай; Японія; Індія; Південна Корея; Бразилія
Профільовані ключові компанії	Microsoft; Google LLC; корпорація Sony; Blippar Limited; Infinity Augmented Reality Limited; Niantic, Inc.; Zappar Limited; Magic Leap, Inc.; Wikitude GmbH
Обсяг налаштування	Безкоштовне налаштування звіту (еквівалентно до 8 робочих днів аналітика) з покупкою. Додавання або зміна країни, регіону та сегмента.

Очікується, що сегмент програмного забезпечення досягне найвищого CAGR 41,8% за прогнозований період. Технологія AR проникає в повсякденне життя в результаті спалаху пандемії. Наприклад, безконтактний і занурювальний досвід, який забезпечує технологія AR, є хорошим пророком як для студентів, так і для співробітників, які працюють вдома. Оскільки технологія AR продовжує розвиватися, її легко і зручно впроваджувати в портативних пристроях. Розробники використовують більш високу обчислювальну потужність, яку забезпечують новітнє обладнання та програмне забезпечення, для розробки простих, але масштабних рішень AR.

2.2 Конкурентні мікроконтролери для апаратної платформи

Для вибору апаратної платформи було розглянуто багато мікроконтролерів

2.2.1 Мікроконтролер Orange Pi One

Orange Pi One, випущений на початку 2016 року, все ще залишається популярним вибором завдяки своїм надійним характеристикам і низькою вартістю. Це може бути повільним через свій вік, але його низька ціна виправдовує компроміс. Найбільш схожий на Raspberry Pi Model A+, не дивно, що на ньому може працювати OctoPrint, подібний до Raspberry, що дозволить вам розширити гру для друку за набагато нижчою ціною.

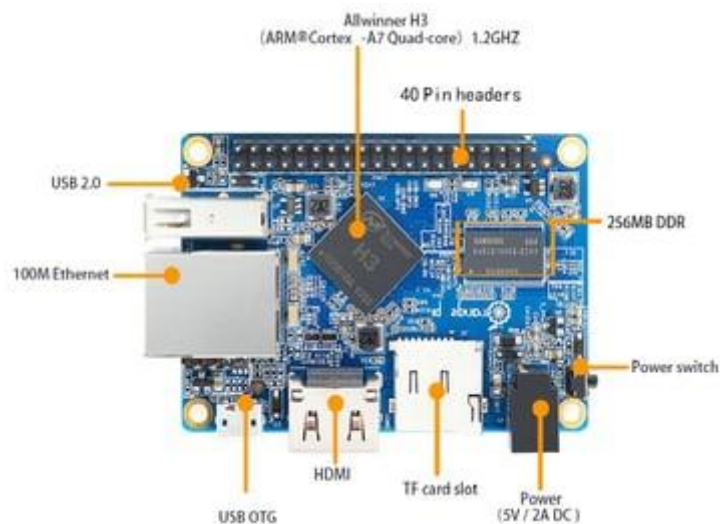


Рисунок 2.1 – Плата Orange Pi One

Процесори

З ARM Cortex-A7, таким як Raspberry Pi 2, і багатьма іншими сучасними SBC, цей 32-розрядний мікропроцесор буде комфортно працювати з відносно вимогливими програмами з невеликим споживанням енергії. Чотириядерний чіпсет H3 з тактовою частотою 1,2 ГГц, виготовлений компанією Allwinner, є відкритим вихідним кодом, і його таблиця даних легко доступна. Порівняно, пропозиція Raspberry Broadcom A7 не є відкритим кодом, що дає Orange перевагу перед Raspberry.

Інтегрований графічний процесор Mali-400 MP2 (2-ядерний) підтримує OpenGL 2.0 і, з HEVC, здатний відтворювати відео 4K зі швидкістю 60 кадрів в секунду. З іншого боку, чіп добре працює вже більше десяти років, тому не розраховуйте на нього для трасування променів у реальному часі.

512 МБ пам'яті DDR3 SDRAM розподіляються між центральним і графічним процесором, розміщеними у двох мікросхемах по 256 МБ (один у верхній частині плати, а другий знизу).

Варто зазначити, що, на думку деяких, процесори в широкому діапазоні плат Orange Pi накопичують значне нагрівання під час великого навантаження. Саме з цієї причини, наприклад, Orange Pi Lite продається з ребрами охолодження в стандартній комплектації. У більшості роздрібних продавців One пропонується з додатковим доповненням невеликого радіатора за кілька додаткових копійок.

Входів і виходів Orange Pi One, хоча і не надихає, достатньо для більшості повсякденних додатків.

Вихід HDMI означає, що, незважаючи на те, що на платі відсутній 3,5-мм I/O, аудіо вихід все ще доступний.

CSI (послідовний інтерфейс камери) присутній на нижній частині плати, тому можливий запис 1080p зі швидкістю 30 кадрів в секунду. Гарною відправною точкою для можливості відеозйомки стане 2-мегапіксельна ширококутна камера, яка продається разом із платою в упаковці за додаткові 5 доларів – корисно для зйомки уповільненої зйомки ваших улюблених відбитків, які оживають!

Є два USB-роз'єми, одне джерело 2.0 і одне OTG. Зауважте, що OTG, на жаль, не налаштований для живлення – це буде через міні-роз'єм 5-V постійного струму.

Підключення до мережі здійснюється через звичайний роз'єм Ethernet RJ45, який забезпечує 100-мегабітну доставку. Незважаючи на те, що він не найбільший, він все ще цілком придатний для потокового відео,

завантаження мікропрограми та підключення до налаштувань домашнього сервера.

Один слот для карти micro SD дозволяє користувачеві розширити ПЗУ до 64 ГБ, але потрібен знімний накопичувач, оскільки плата не має вбудованої пам'яті.

40-контактний роз'єм має живлення GPIO і повністю сумісний з такими, як Raspberry (B+), але повернутий на платі на 180 градусів порівняно з більш популярними SBC.

За допомогою цього зручного для гаманця SBC ви можете налаштувати домашній сервер, грати в ретро-ігри, автоматизувати свій 3D-друк, створювати музику, транслювати HD-відео, керувати налаштуванням динаміків домашнього кінотеатру та багато іншого.

Подібно до Raspberry Pi 1, Orange Pi One залишається універсальним і, можливо, трохи застарілим. Запуск Android 4.4 є поширеним варіантом, але PiHole і RetroPi також є на столі для тих конкретних завдань, які ви можете мати на увазі.

Однак, на відміну від сімейства Raspberry, як і решта лінійки Orange Pi, Orange Pi One дещо бракує підтримки на офіційному веб-сайті. Спільнота, яка читає Orange Pi, зовсім не схожа на Raspberry, і тому цей маленький комп'ютер, можливо, не є правильним вибором для абсолютних новачків. Однак існують невеликі довідкові теми.

Не забуваймо, що Orange Pi на 30% дорожче Raspberry Pi 2 Model B, який, за загальним визнанням, має трохи більший діапазон функцій і кращий процесор. Незважаючи на це, це все ще вдвічі дешевше Raspberry Pi 1 Model B+ (що ближче SBC за характеристиками).

Таблиця 2.2 – Специфікація мікроконтролеру Orange Pi One

ЦП	ARM Cortex-A7 (1,2 ГГц)
Графічний процесор	Графічний процесор Mali-400 MP2 (600 МГц і підтримує OpenGL ES 2.0)
Синхронна динамічна пам'ять	512 МБ DDR3 (спільний з графічним процесором)
Варіанти зберігання	TF-карта (максимум 32 ГБ)
Мережа	Порт Ethernet 10/100 Мб
Відео вхід	CSI (підтримка відеозйомки до 1080р при 30 кадрах в секунду)
Відео вихід	HDMI
Джерело живлення	5 В / 2 А постійного струму
Порти USB 2.0	Хост USB 2.0 (1), USB 2.0 OTG (1)
Підтримувана ОС	Android 4.4, Ubuntu, Debian
Розміри і вага	69 x 48 мм; 36 г

Зрештою, середня кількість функцій і конкурентоспроможна ціна роблять Orange Pi One хорошою альтернативою SBC, що розглядаються так само. Ще кращим є Orange Pi Lite, випущений у травні 2016 року, який ідентичний One, але має можливості WiFi замість Ethernet.

2.2.2 Міні-комп'ютер Raspberry Pi 1B+

Raspberry Pi — це комп'ютер розміром із кредитну картку з процесором ARM, який може працювати з Linux. Цей елемент — Raspberry Pi 1 Model B+ 512 МБ, який має 512 МБ оперативної пам'яті, вихід HDMI, аудіовихід, композитний відеовихід RCA (через 3,5-мм роз'єм), один USB-порт і контакти на відстані 0,1 дюйма, які забезпечують доступ до входів і виходів загального призначення (GPIO). Модель A Raspberry Pis менша і дешевша, ніж модель B Raspberry Pis. Для Raspberry Pi потрібна карта microSD з операційною системою (не входить в комплект). Raspberry Pi дуже популярний, з великою кількістю прикладів проектів та інформації, доступної в Інтернеті.

Raspberry Pi — це комп'ютер розміром з кредитну картку. Raspberry Pi Model A+ — це менша, недорога версія Raspberry Pi. Він заснований на системі на чіпі (SoC) BCM2835, яка включає процесор ARM11 і потужний графічний процесор. Raspberry Pi Model A+ рекомендується більше для вбудованих та малопотужних проектів, які не потребують Ethernet або кількох портів USB.

Raspberry Pi був розроблений Raspberry Pi Foundation, щоб забезпечити доступну платформу для експериментів та навчання комп'ютерного програмування. Raspberry Pi можна використовувати для багатьох речей, які виконує звичайний ПК, включаючи використання комп'ютерних мов високого рівня (наприклад, Python), виконання ролі хоста USB та багатозадачність операційної системи. Понад дванадцять мільйонів Raspberry Pi було продано, і багато ресурсів для Raspberry Pi доступні в Інтернеті.

Завдяки своєму заголовку GPIO з відстанню 0,1 дюйма та невеликому розміру Raspberry Pi також працює як програмований контролер у широкому спектрі робототехнічних та електронних програм. Його також можна поєднати з нашим контролером робота A-Star 32U4 LV з Raspberry Pi Bridge, щоб зробити чудовий контролер для маленького робота. Ми також пропонуємо широкий вибір плат розширення Raspberry Pi.

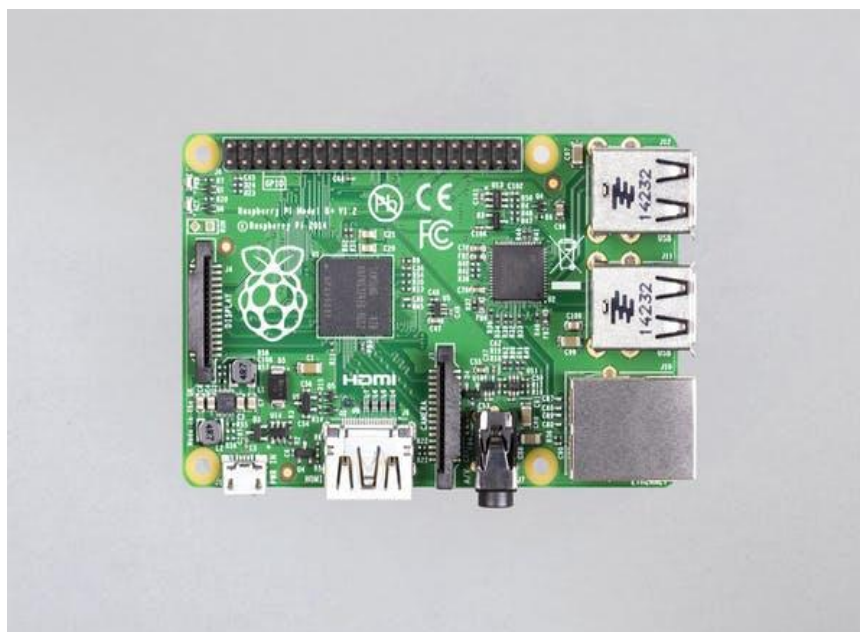


Рисунок 2.2 – Raspberry Pi 1 B+

Якщо вам потрібен повноцінний, мініатюрний комп'ютер, який може до того ж взаємодіяти із зовнішнім світом, платформа Raspberry Pi підійде на цю роль якнайкраще.

З Raspberry Pi ви зможете реалізувати безліч креативних проєктів. Просто для прикладу:

- Виділений веб-сервер.
- Сервер управління розумним будинком.
- Охоронну систему з розпізнаванням облич.
- Робота з комп'ютерним зором.

На платі розміром із кредитну картку ви знайдете все те, що можете знайти у звичайному персональному комп'ютері: процесор, оперативну пам'ять, роз'єм HDMI, USB, Ethernet, аналогові аудіо- та відеовиходи. Крім того, на платі розташовано 40 контактів введення/виведення загального призначення. До них ви зможете підключати периферію для взаємодії із зовнішнім світом: виконавчі пристрої на кшталт реле і сервомоторів або будь-які сенсори; загалом все, що працює від електрики.

Штатною операційною системою для Raspberry Pi є Linux. Вона встановлюється на micro-SD карту, а та у свою чергу – у спеціальному слоті на платі. Якщо ви не знаєте Linux, не варто лякатися. Навпаки: цей комп'ютер - чудова можливість у всьому розібратися. Втратити дані або сильно помилитися з налаштуваннями не так страшно, адже образ на карті SD можна відновити за лічені хвилини. Після цього можна продовжити експерименти з чистого аркуша або певної контрольної точки.

Raspberry Pi Model B Plus - це друга ревізія надпопулярної Raspberry Pi Model B. У ній збільшено кількість USB-портів, знижено енергоспоживання, збільшено кількість входів/виходів загального призначення, поєднано композитний відеовихід та аудіовиход, покращено форм-фактор.

Комп'ютер виконаний з урахуванням SoC (System on Chip) Broadcom BCM2835. Процесором є ARM1176JZ-F із тактовою частотою 700 МГц. Вона може бути збільшена до 1 ГГц програмними налаштуваннями.

Raspberry Pi Model B+ має 512 МБ оперативної пам'яті, що працює на частоті 400 МГц. Пам'ять ділиться із графічною підсистемою.

GPU комп'ютера апаратно підтримує OpenGL ES 2.0, OpenVG, MPEG-2, VC-1, кодування та відтворення 1080p30 H.264/MPEG-4.

Роздільна здатність можна варіювати від 640×350 (EGA) до 1920×1200 (WUXGA). На композитному виході можна генерувати сигнали 576i чи 480i у форматі PAL чи NTSC.

Таблиця 2.3 – Специфікація міні-комп'ютера Raspberry Pi 1 B+

Портів введення-виведення загального призначення	40
Шини	UART, I2C, SPI
Живлення	3.3, 5 В / 2 А постійного струму
Відео вихід	HDMI
Порти USB 2.0	Хост USB 2.0 (1), USB 2.0 OTG (1)
OS	Rasbian, Debian
Розміри і вага	69 x 48 мм; 36 г

Замість традиційного для звичайних комп'ютерів жорсткого диска, Raspberry Pi використовує microSD флеш-карту. Вона має бути попередньо підготовлена, на неї слід встановити операційну систему на вибір.

Маючи кілька флеш-карток, ви можете по черзі використовувати їх і таким чином отримати, по суті, кілька ізольованих образів комп'ютерів

Флеш-карта не входить.

Підтримуються картки розміром від 2 ГБ. Об'єм, що рекомендується, — не менше 4 ГБ.

2.2.3 Міні-комп'ютер ASUS Tinker Board

Tinker Board це мініатюрний комп'ютер на одній платі (SBC), що має кращу продуктивність у своєму класі, а також безліч можливостей щодо створення систем на його основі. Tinker Board призначений для розробників, ентузіастів та інших користувачів, котрі хочуть втілити свої ідеї в реальність.

Сам пристрій являє собою друковану плату розмірами 85x55 мм (вага - 55 г) з розпаяними на ній контролерами та портами, а також з маркуванням, що за що відповідає. На лицьовій стороні плати розпаяно більшість найважливіших компонентів.

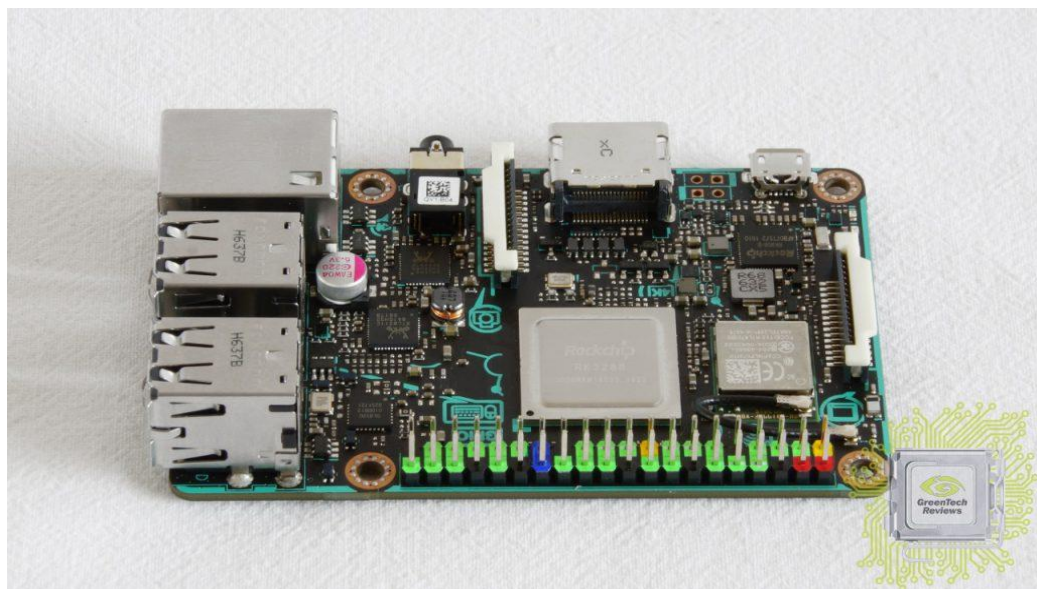


Рисунок 2.3 – Плата мікрокомп'ютера ASUS Tinker Board

На одній із сторін розміщено microUSB порт (для живлення рекомендується адаптер з вихідними характеристиками 5В при 2.5А), HDMI (1.4) вихід для підключення екрану, 3.5 мм аудіо роз'єм для підключення навушників або мікрофона.

Біля USB порту для живлення пристрою реалізовано спеціальний конденсатор, який мінімізує перепади напруги для стабільної роботи пристрою. На одній з бічних сторін виробник розмістив чотири повнорозмірних (Type-A) USB порту і гігабітний мережевий порт RJ-45.

Постійна пам'ять не реалізована - операційна система "крутиться" з microSD карти. Слот підтримує стандарт SDIO 3.0. Відмінною особливістю порівняно з конкурентами є використання 2-канальної оперативної пам'яті LPDDR3 об'ємом 2 ГБ. Мікросхеми вироблені Samsung і мають маркування K4E8E32 4EBEGCF. Серцем пристрою є 4-ядерний SoC RK3288, що функціонує на частоті 1.8 ГГц (з можливістю розгону до 2.2 ГГц) і вміщує графічне ядро Mali-T764 з чотирма ядрами на частоті 600 МГц, що робить його самим . За звук відповідає аудіо кодек Realtek ALC4040, за мережу контролер RTL8211E цього ж виробника. USB порти реалізовані хабом GL852G.

За бездротові підключення мережі відповідає AW-NB177NF, а за підсистему живлення RK808-B.

GPIO.Setmode (GPIO.ASUS)	GPIO.Setmode (GPIO.BOARD)	Pinout	Physical Pin Number	Pinout	GPIO.Setmode (GPIO.BOARD)	GPIO.Setmode (GPIO.ASUS)
	1	VCC3.3V_IO	1 2	VCC5V_SYS	2	
252	3	GP8A4_I2C1_SDA	3 4	VCC5V_SYS	4	
253	5	GP8A5_I2C1_SCL	5 6	GND	6	
17	7	GP0C1_CLKOUT	7 8	GP5B1_UART1TX	8	161
	9	GND	9 10	GP5B0_UART1RX	10	160
164	11	GP5B4_SPIOCLK_UART4CTSN	11 12	GP6A0_PCM/I2S_CLK	12	184
166	13	GP5B6_SPIO_TXD_UART4TX	13 14	GND	14	
167	15	GP5B7_SPIO_RXD_UART4RX	15 16	GP5B2_UART1CTSN	16	162
	17	VCC3.3V_IO	17 18	GP5B3_UART1RTSN	18	163
257	19	GP8B1_SPI2TXD	19 20	GND	20	
256	21	GP8B0_SPI2RXD	21 22	GP5C3	22	171
254	23	GP8A6_SPI2CLK	23 24	GP8A7_SPI2CSN0	24	255
	25	GND	25 26	GP8A3_SPI2CSN1	26	251
233	27	GP7C1_I2C4_SDA	27 28	GP7C2_I2C4_SCL	28	234
165	29	GP5B5_SPIOCSN0_UART4RTSN	29 30	GND	30	
168	31	GP5C0_SPIOCSN1	31 32	GP7C7_UART2TX_PWM3	32	239
238	33	GP7C6_UART2RX_PWM2	33 34	GND	34	
185	35	GP6A1_PCM/I2S_FS	35 36	GP7A7_UART3RX	36	223
224	37	GP7B0_UART3TX	37 38	GP6A3_PCM/I2S_SDI	38	187
	39	GND	39 40	GP6A4_PCM/I2S_SDO	40	188

Рисунок 2.4 – Підключення інтерфейсів до плати

Tinker OS – це операційна система, заснована на Debian і модифікована для роботи з Tinker Board.

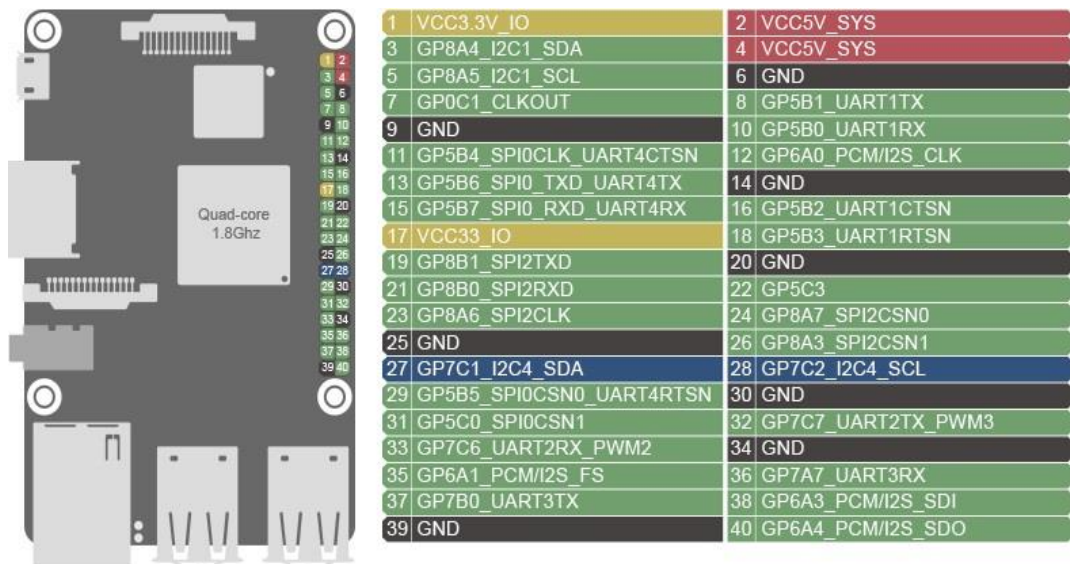


Рисунок 2.5 – Розміщення інтерфейсів на платі

Однією з основних функцій Tinker Board є наявність GPIO (general purpose input/output). Ці конектори є зв'язком Tinker Board з навколишнім світом. З 40 контактів 28 є GPIO (суміщені з SPI/UART/I2C). Плата Tinker Board обладнана однією шиною SPI.

H264, H265

H.264/AVC	
Video Codec	H.264
Supported Image Size	48x48pixels to 4096x2304pixels
Maximum Frame Rate	4K@25fps,1080P@60fps
Maximum Bit Rate	30Mbps(4K CAVLC) 60Mbps(4K CABAC) 60Mbps(1080p CABAC+CAVLC)
File Format	AVI, MKV, MP4, MOV, 3GP, TS, FLV
Remarks	<ul style="list-style-type: none"> Support Field Coding; Support MBAFF

H.265/HEVC	
Video Codec	H.265/HEVC
Supported Image Size	64x64pixels to 4096x2304pixels
Maximum Frame Rate	60fps
Maximum Bit Rate	80Mbps(8bit) 50Mbps(10bit)
File Format	MKV, MP4, MOV, TS
Remarks	<ul style="list-style-type: none"> Support Main Profile & Main10 Profile; Support 10-bit Color decoding; Support Tile & Slice;

Рисунок 2.6 - Підтримувані формати та обмеження ASUS Tinker Board
Відтворення відео у форматах HD та Ultra-HD зі швидкістю 30 кадрів в секунду зараз доступне лише за допомогою відеоплеєра Rockchip, який

працює в операційній системі TinkerOS. Сторонні відеоплеєри та програми можуть не підтримувати апаратне прискорення або мати якісь інші обмеження щодо продуктивності чи стабільності при відтворенні відео.

Таблиця 2.3 – Специфікація мікрокомп'ютера ASUS Tinker Board

SoC:	RK3288
ЦП	4-ядерний 1.8 ГГц ARM Cortex-A17
Графічний процесор	4-ядерний 600 МГц Mali-T764
ОЗП	2 ГБ 2-канальна LPDDR3
Постійна пам'ять	Слот для картки MicroSD (SDIO 3.0)
Провідна мережа	1 Гбіт/с (не суміщена з шиною USB)
Бездротова мережа	Bluetooth 4.0 + EDR, Wi-Fi 802.11 b/g/n (з конектором для антени IPEX-MHF4)
HDMI	HDMI 1.4 4K/30 кадрів в секунду
USB	4 x USB2.0
Звук	3.5 мм аудіо роз'єм (лінійний вихід та вхід для мікрофона)

На відміну від багатьох інших продуктів цього класу Tinker Board має аудіокодек формату HD Audio з підтримкою форматів аж до 192 кГц / 24 біти. Крім того, Tinker Board має стандартні для свого класу інтерфейси, включаючи 40-контактний GPIO і два інтерфейси MIPI для підключення дисплеїв та камер формату HD. Ті, хто хотіли б використовувати Tinker Board у складі комп'ютерної мережі, неодмінно оцінять наявність високошвидкісного порту Gigabit Ethernet, який пропонує високу пропускну здатність за рахунок шини, що не розділяється з іншими пристроями. Також є бездротові інтерфейси Wi-Fi і Bluetooth, причому їхній модуль екранований для захисту від електромагнітних перешкод, а роз'єм i-PEX дозволяє підключити додаткову антену замість вбудованої. Комп'ютер Tinker Board створювався як платформа, дружня як до досвідчених користувачів, так і до новачків. Це проявляється в таких деталях як колірне кодування контактів GPIO-колодки, що полегшує підключення кабелів, а завдяки стандартному розміру друкованої плати та топології компонентів Tinker Board сумісний з великою кількістю доступних у продажу корпусів та аксесуарів. Для

зручності використання компоненти пристрою мають підписи прямо на друкованій платі.

2.2.4 Міні-комп'ютер Khadas Vim1

VIM1 — це оригінальна плата розробника Amlogic S905X ARM, розроблена Khadas для любителів, виробників та любителів електроніки; багато хто також ласкаво називають його «телевізійною коробкою з відкритим кодом». Він крихітний, трохи менший за Raspberry Pi з розмірами, подібними до розмірів кредитної картки. VIM1 доступний лише в двох версіях: VIM1 Basic з 8 ГБ пам'яті EMMC і AP6212 (b/g/n) Wi-Fi і VIM1 Pro з 16 ГБ пам'яті EMMC і AP6255 (b/g/n/ac) Wi-Fi. Обидві версії мають 2 ГБ оперативної пам'яті DDR3 і Bluetooth 4.2.

Так само, як і ваш знайомий настільний комп'ютер, одноплатний комп'ютер VIM1 підтримує завантаження кількох операційних систем. Ви можете вибрати один із різноманітних образів ОС (ПЗУ) з наших документів Khadas або на форумі спільноти. Популярні ОС: Android 7.1 / 9.0, Ubuntu 18.04, LibreELEC, Armbian, Buildroot тощо. Є навіть образи подвійного завантаження, щоб розробники могли завантажувати Android і Ubuntu з одного EMMC.

Розробники виявлять, що вбудований 40-контактний GPIO, годинник реального часу (RTC) і порт VIN для додаткового джерела живлення будуть дуже зручними. Крім того, три користувацькі кнопки збоку забезпечують легкий доступ до кнопок живлення, скидання та функціональних клавіш. Повний відкритий вихідний код доступний на Github, а апаратні схеми можна знайти в нашій документації. На довершення до всього, є навіть вбудований блок керування живленням для підтримки включення живлення за допомогою пульта дистанційного керування.

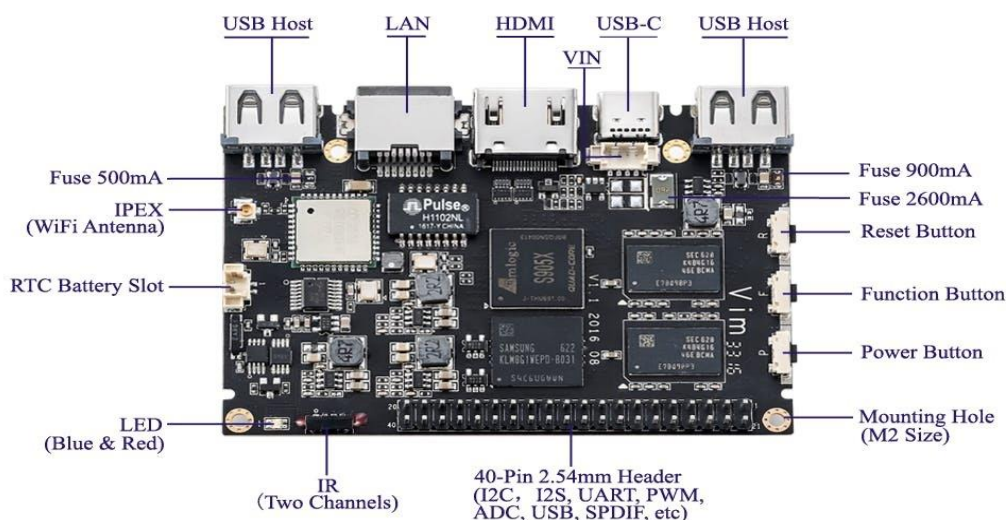


Рисунок 2.7 – Міні-комп'ютер Khadas Vim1

У порівнянні з восьмиядерним SoC Amlogic S912 у VIM1, чотириядерний SoC S905X у VIM1 генерує значно менше тепла. Це робить VIM1 особливо корисним у середовищах, які дозволяють лише пасивну конвекцію повітря для розсіювання тепла, а не примусовий потік повітря. У багатьох випадках достатньо одного радіатора, і деякі користувачі можуть навіть вважати радіатор непотрібним для їх застосування.

Мінікомп'ютер Khadas VIM1 працює з системами Android, AndroidTV, LibreELEC, Ubuntu, DualOSandU-Boot. Систему можна завантажити на карту microSDor безпосередньо у внутрішню пам'ять міні-комп'ютера.

В якості джерела живлення в системі використовується роз'єм USB типу C. Виробник рекомендує, щоб напруга, зазначена на вилці, мала значення 5,0, а мінімальна ефективність по струму використовуваного адаптера не менше 2,0 А. У нашому асортименті ви знайдете блоки живлення, які відповідають цим параметрам, наприклад: блок живлення Extreme 3,1 А.

Таблиця 2.4 – Специфікація мінікомп'ютера Khadas Vim1

Процесор	Amlogic S905X
Ядро	Чотириядерний ARM Cortex A53
Графічний процесор	ARM Mali-450

OS	Android, AndroidTV, LibreELEC, Ubuntu, DualOS, U-Boot
Тактова частота	1,5 ГГц
ОЗП	2 ГБ DDR3 64-розрядна
Пам'ять	16 ГБ eMMC, картка microSD
Слот GPIO	40-контактний GPIO
Живлення	5V / 2A - USB type C
Розміри плитки	82 x 57,5 x 11,5 мм

У надзвичайних випадках відновлення після аварії вимагатиме використання MRegister для скидання SBC VIM1. USB Serial Debug Tool також корисний для розробників, які налагоджують складні проблеми програмного забезпечення.

Провідний металевий пінцет (для скидання мертвого SBC через MRegister) USB Serial Debug Tool (Для діагностики проблем програмного та апаратного забезпечення)

2.2.5 Міні-комп'ютер PINE64 Rock64

ROCK64 — це платний медіа-комп'ютер розміру 4K60P HDR, що працює на основі чотирьохядерного 64-розрядного процесора Rockchip RK3328 ARM Cortex A53 і підтримує до 4 ГБ пам'яті LPDDR3 1600 МГц. Він забезпечує роз'єм для модуля eMMC, слот для карти MicroSD, шину Pi-2, шину Pi-P5+, USB 3.0 та багато інших периферійних пристроїв інтерфейс для виробників для інтеграції з датчиками та пристроями. Різні операційні системи (ОС) доступні спільноту з відкритим кодом, наприклад Android 7.1, Debian, Yocto та багатьма іншими.

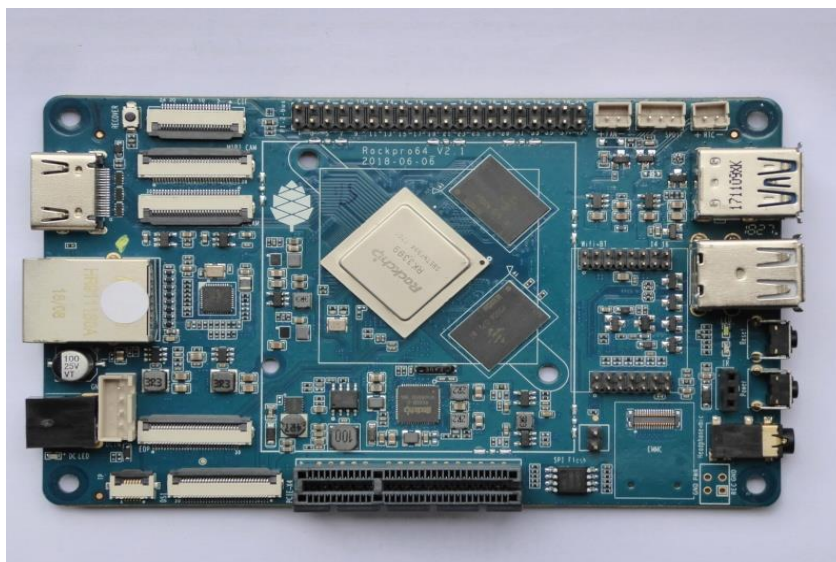


Рисунок 2.8 – Міні-комп'ютер PINE64 Rock64

- Зелений світлодіод біля вхідного роз'єму 12 В горітиме, доки до роз'єму подається 12 В. (Навіть якщо Rock64 вимкнено.)
- Білий світлодіод за кнопкою скидання буде світитися, поки RockPro64 працює (він загоряється через кілька секунд після ввімкнення, коли управління передається операційній системі).
- Червоний світлодіод за кнопкою скидання є DIY - він горить, наприклад, якщо плата перебуває в режимі OTG із зображенням Auufan, або якщо зображення Android знаходиться в режимі очікування.
- Жовтий і зелений світлодіоди на розетці LAN поведуться стандартним чином.

Між роз'ємом eMMC (14) і мікросхемою SPI (17) є 2-контактна перемичка (16) без маркування (на шовкографії друкованої плати). На принциповій схемі він позначений як SW4. Умова за замовчуванням – ВІДКРИТО (без перемички). Це корисно для керування завантаженням .

Пристроєм завантаження за замовчуванням (без програмного забезпечення SPI) є eMMC, потім SDcard. Якщо і eMMC, і SD-карта містять завантажувальні образи, eMMC можна вимкнути, встановивши перемичку. Це повністю видаляє eMMC з отриманої ОС. Якщо ви хочете, щоб eMMC був

видимий у завантаженій ОС, перемичку слід видалити через 2 секунди після подачі живлення (і до того, як загориться білий світлодіод).

Випуску програмного забезпечення ROCK64 ви знайдете повний список підтримуваних на даний момент образів операційної системи, які працюють з ROCK64, а також іншим супутнім програмним забезпеченням. На сторінці випуску програмного забезпечення є посилання для завантаження зображень, а також інструкції високого рівня щодо завантаження кожного зображення.

Таблиця 2.5 – Специфікація міні-комп'ютеру ROCK64

Процесор	RK3328 64-біт 4 ядра CoreA53
Оперативна пам'ять	4 ГБ LPDDR3
Пам'ять програм	microSD карта пам'яті до 64Гб
Роз'єм живлення	3,55мм / 1,1мм силовий коаксіальний роз'єм
Відеовихід	HDMI 4K60P HDR цифровий інтерфейс
Мережевий інтерфейс	Gigabit Ethernet
USB інтерфейс	USB 2.0 - 2 шт, USB 3.0 - 1 шт
GPIO інтерфейси	Шина Pi-P5 +, Шина Pi-2
Напруга живлення	5В
Максимальний споживаний струм	3А

ROCK64 має сокет для підключення модуля пам'яті родного eMMC, що забезпечує ще більш швидку роботу системи. Також можна підключити обычну microSD. Плата має порти HDMI 2.0, USB 2.0 і USB 3.0, що дозволяє підключити з гарною швидкістю, наприклад, зовнішній диск.

2.2.6 Порівняльна характеристика розглянутих моделей міні-комп'ютерів

Тепер, розглянувши мікрокомп'ютери, можна зробити висновки, який мікрокомп'ютер більше підійде для конкретних завдань. Ну і в цілому, що хочу сказати: вибирати мікрокомп'ютери за заявленими характеристиками

від виробника абсолютно безглуздо, тому що є ще мільйон нюансів, які потрібно перевіряти практично або самостійно, або дивитися тести, причому не один, а кілька, порівнювати їх і дивитися, як та чи інша модель мікрокомп'ютера поводить у різних завданнях

Таблиця 2.6 Порівняльна характеристика розглянутих моделей міні-комп'ютерів

	Вага	Ціна	Живлення	GPIO	Оперативна пам'ять	Відеовихід
Raspberry Pi 1 B+	36 г	1650 грн	3,3 В, 2А	Так	512 МБ DDR3	HDMI
ASUS Tinker Board	55 г	5865 грн	3,3 В, 2А	Так	2 ГБ 2-канальна LPDDR3	HDMI
Khadas Viml	28.5 г	3805 грн	3,3 В, 2А	Так	2 ГБ DDR3 64-розрядна	HDMI
PINE64 Rock64	38 г	1893 грн	3,3 В, 2А	Так	4 ГБ LPDDR3	HDMI
Orange Pi One	36 г	999 грн	3,3 В, 2А	Так	512 МБ DDR3	HDMI

Для поставленої задачі було обрано Raspberry Pi 1B+, цей міні-комп'ютер буде використовуватись для реалізації апаратної частини.

2.3 Камери для доповненої реальності у командних іграх

Люди грають в ігри для кількох цілей, у тому числі освіти, охорони здоров'я, розваг тощо. Ігри можна грати різними способами, використовуючи папір, усно, вручну або з використанням електронного носія. Ігри, які використовують паперові, усні чи ручні засоби, цього не роблять обов'язково потрібні технологічні пристрої для їх виконання, що дозволяє використовувати природну взаємодію. До того ж електронні ігри переміщують користувача у віртуальний світ, переступаючи межі логіки та уява, що дозволяє використовувати руки безпосередньо або через спеціальні пристрої для підтримки взаємодії.

Основним недоліком звичайних електронних ігор є адаптація користувача, необхідна для маніпулювання неприродним пристрої інтерфейсу. Завдяки технічним досягненням, за допомогою техніки доповнена реальність, віртуальний світ може бути представлений користувачеві простір, забезпечуючи природний і приємний досвід нове середовище. Тут комп'ютер поєднує реальне середовище декорації з віртуальними об'єктами, що показують одиночний змішане середовище перед користувачем, який може візуалізувати це середовище безпосередньо через дисплей, встановлений на голові або опосередковано через монітор, наприклад. Використовуючи руки, у додатку доповненої реальності користувач може маніпулювати реальними та віртуальними об'єктами в змішаному довкіллі, без необхідності спеціального обладнання, наприклад джойстик, клавіатура та миша.

Доповнену реальність зазвичай визначають як перекриття тривимірних віртуальних об'єктів, створених комп'ютером, з реальним середовищем, через якийсь технологічний пристрій [19]. Це поняття можна краще зрозуміти через його вставка в ширший контекст, пов'язаний із змішаною реальністю.

Змішана реальність поєднує реальне з віртуальним середовищем і включає дві можливості: доповнену реальність, чия переважним середовищем є реальний світ, і доповнений віртуальність, переважним середовищем якої є віртуальне світ [19], [20]. На рисунку 2.9 представлена схема безперервна реальність/віртуальність, на основі [20], що показує поступові можливості перекриття реального зі своїм віртуальне середовище і навпаки.



Рисунок 2.9. - Схема безперервної реальності/віртуальності

Доповнена реальність забезпечує дружній комп'ютер взаємодія, оскільки вона приносить віртуальні об'єкти для простору користувача, що дозволяє йому/їй візуалізувати та маніпулювати як реальними, так і віртуальні об'єкти. Ця функціональність досягається завдяки методики комп'ютерного зору та віртуальної реальності, які дозволяють перекривати віртуальні об'єкти з реальним світом

Крім того, доповнена реальність дозволяє користувачеві працювати предмети своїми руками, що робить можливим привабливим взаємодії у змішаному середовищі для цього необхідно використовувати програмне забезпечення, здатне до відстежувати положення рук або щось еквівалентне цьому. Хоча деякі пристрої використовуються в середовищах віртуальної реальності можна використовувати в середовищах доповненої реальності, вони іноді потребують адаптації

Проста платформа для розробки доповненої реальності додатків базується на мікрокомп'ютері з встановленим веб-камера. У цьому середовищі є спосіб змішати віртуальне з реальне середовище полягає у використанні будь-якого програмного забезпечення, яке фіксує кадри зображень із реального середовища через Інтернет та камери і обробляє їх, розміщуючи віртуальні об'єкти в справжні пейзажі. Крім того, програмне забезпечення також потребує керувати взаємодіями з віртуальними об'єктами, змінюючи їх позиції та інші характеристики в реальному часі, надаючи враження єдиного середовища. На рисунку 2.10 це показано модель доповненої реальності, яка забезпечує бачення за допомогою відео, на основі монітора та веб-камери.

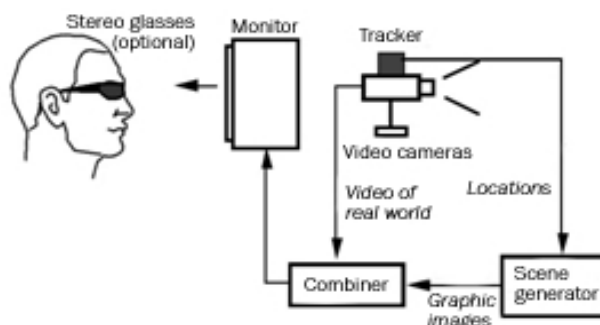


Рисунок 2.10 - Доповнена реальність на основі монітора та веб-камери

Доповнена реальність значною мірою сприяє цьому області ігор, що надає користувачеві привабливу візуалізація та природна взаємодія. Крім того, збільшений реальність середовища не вимагає спеціальних пристроїв і інтенсивне навчання користувачів.

2.3.1 Веб-камера Promate ProCam-2 FullHD USB Black

Природні кольори, плавний відеопотік, чисте передачі звуку та простота використання. Камера надає максимум від онлайн спілкування.

Висока якість: транслюється потокове відео з роздільною здатністю FullHD (1920x1080) за 30 кадрів в секунду.

Автофокусування: забезпечує чітке та контрастне зображення об'єкта перед камерою незалежно від фону.



Рисунок 2.11 – Камера Promate ProCam-2 FullHD

Ширококутна лінза: тримає все найважливіше у полі зору завдяки куту огляду 120°.

Автоматична корекція кольору завдяки чіпсету Hisilicon-3518EV300 вебкамера підбере оптимальну яскравість та встановить баланс білого. Якісна картинка навіть в умовах поганого освітлення.

Вбудований HD-мікрофон Завдяки системі шумозаглушення гарантує високу якість передачі голосу.

Швидка установка: за допомогою вбудованої прищипки гаджет можна закріпити на моніторі та підключити за допомогою USB-кабелю.

Антиковзний трипод можна надійно зафіксувати на поверхні і встановити комфортну висоту (17-23 см), а завдяки панорамній голівці камера обертається на 360°.

Plug & Play: Windows або Mac OS і жодного додаткового ПЗ, тобі потрібен лише вільний USB-порт.

Таблиця 2.7 Специфікація веб-камери Promate ProCam-2 FullHD

Чіпсет	Hisilicon-3518EV300
Сенсор	1/2,9 F23
Максимальна роздільна здатність	FullHD (1920x1080) 30 кадрів/с.
Фокусування	Авто
Кут огляду	120 °
Корекція кольору	Авто
Живлення	5В/160мА
Сумісність	Windows, Mac OS (10.7 та вище)
Висота триподу	17-23 см
Розміри камери ШхВхГ	100×50×50 мм
Довжина кабелю USB	140 см.
Вага	195 г

2.3.2 Веб-камера VERTUX VERTUCAM-4K UHD USB BLACK

Корпус камери зроблений із пластику, складання якісне. Передня панель легко збирає відбитки пальців. Є заслінка для закриття камери. Камера може обертатися та нахилитися в різні боки (подивитися це можна у відеоогляді, починаючи з 35 секунд).

Знизу знаходиться 1.4-дюймовий роз'єм, за допомогою якого можна встановлювати камеру на штатив. Камера зручно та надійно фіксується як на звичайному моніторі, так і на моніторі ноутбука. На сторінці опису лота вказано, що камера має два динаміки, проте мені не вдалося їх задіяти.

Максимальна роздільна здатність також виявилася нижчою, ніж зазначено на сторінці виробника: 2592x1944 замість 3840x2160.

Що стосується якості відео – вона чітка, навіть незважаючи на те, що роздільна здатність нижча від заявленої. Автофокус працює коректно та досить швидко навіть в умовах поганого освітлення



Рисунок 2.12 – Веб-камера VERTUX VERTUCAM-4K UHD USB
BLACK

Завдяки можливості підключення за технологією Plug-and-Play USB 3.0 вам не потрібно турбуватися про встановлення драйверів, просто підключіть веб-камеру до комп'ютера, і все готово. Камера також сумісна з Windows, Mac OS, основним програмним забезпеченням .

Таблиця 2.8 Специфікація веб-камери VERTUX VERTUCAM-4K UHD
USB BLACK

Роздільна здатність відео	Ultra HD 4K (3840x2160)
Сенсор	Full HD
Фокусування	Автофокус
Кріплення	Настільне
Сумісність ОС	Windows 10, macOS 10.10
Поле огляду	170
Живлення	USB
Особливості	Частота кадрів за секунду 30

Чіпсет	Sunplus SPCA2688
Потужність вбудованого динаміка	1.2 Вт
Розміри	138 x 53 x 63 мм
Вага	133 г

За рахунок чіпсету Sunplus SPCA2688 веб-камера підбере оптимальну яскравість і налаштує баланс білого. Якісна картинка навіть за умов поганого освітлення.

2.3.3 Веб-камера VERTUX ODIN-4K UHD з LED підсвіткою USB BLACK

Odin-4K здатний записувати, потокове передавання та дзвінки VOIP з повною чіткістю зображення 4K Ultra HD. Автоматична корекція зображення та освітлення забезпечує найкращу чіткість у будь-яких умовах. Кільцеве підсвічування з рівнів яскравості забезпечує різні вимоги до освітлення



Рисунок 2.13 - Веб-камера VERTUX ODIN-4K UHD

Об'єкти зйомки будуть виглядати професійно незалежно від рівня освітленості приміщення, у тому числі за нестачі світла або висококонтрастного світла, а також під прямими променями сонця, адже ця камера підтримує технологію RightLight є сертифікованою моделлю бізнес-класу та оснащена інтегрованими оптичним та інфрачервоним датчиками, а також підтримує функцію розпізнавання облич Windows Hello.

Таблиця 2.9 Специфікація веб-камера VERTUX ODIN-4K UHD

Чіпсет	Lingtong 5801
Сенсор	CMOS
Фактичні пікселі	3264x2448
Ефективні пікселі	3840x2160
Частота кадрів/с	30
Формат відеосигналу	MJPEG/YUV2
Фокусування	Авто
Кут огляду	70 °
LED підсвічування	3 режими
Корекція кольору	Авто
Живлення	5В/160мА
Сумісність	Windows, macOS
Висота триподу	17-23 см

За рахунок просунутого чіпсету веб-камера підбере оптимальну яскравість та налаштує баланс білого. Надійно фіксується на поверхні і дозволяє встановити комфортну висоту (17-23 см), а за рахунок панорамної головки камера обертається на 360 см.

2.3.4 Порівняльна характеристика розглянутих веб-камер для доповненої реальності

Основними критеріями для оцінки була компактність та доступність камери для зменшення вартості проекту, розглянуто декілька камер з високою якістю картинки та сумісністю з різними операційними системами.

Таблиця 2.10 Порівняльна характеристика веб-камер

	Вартість	Сумісність ОС	Якість зображення	Кут огляду
Promate ProCam-2	999 грн	Windows, macOS	1920x1080, 30 к/с	120

VERTUX VERTUCAM- 4K	3499 грн	Windows, macOS	3840x2160, 30 к/с	170
VERTUX ODIN-4K UHD	2999 грн	Windows, macOS	3840x2160, 30 к/с	70

Для поставленої задачі було обрано веб-камеру Promate ProCam-2 через її низьку вартість та велику ефективність виконання поставлених задач, та низьку вартість і легко доступність.

Висновки до розділу 2

Апаратна частина реалізована за допомогою Raspberry Pi 1B+ та з використанням веб-камери Promate Procam-2, проаналізовано сумісність вибраних компонентів з Raspberry Pi 1 B+. Веб-камера була підключена до RPi через USB порт .

Апаратна частина відповідає за роботу застосунку до гри, та розвантажує основну базу даних для точного обчислення алгоритму гри з математичної точки зору, так як дані повинні бути оброблені дуже швидко та одразу передаватися до гравців щоб визначити позиціонування у просторі маркерів та м'яча.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ

Для реалізації командної гри було обрано фреймворк ARToolKit так як він дуже ефективний у роботі з веб застосунками та легко реалізується на JavaScript.

3.1 Фреймворк ARToolKit

ARToolKit [21] — це програмне забезпечення, призначене для розробки додатків доповненої реальності на основі реального світу, знятого камерою, яка використовує комп'ютерний зір, обробку зображень та позиціонування простору.

Серед труднощів, пов'язаних з розвитком додатків доповненої реальності, існує багато видів діяльності, які виконуються в режимі реального часу, включаючи: відстеження реальних об'єктів, калібрування та позиціонування віртуальних об'єктів, поєднання реального та віртуального середовища та взаємодії з віртуальними об'єктами. У цьому контексті ARToolKit мінімізує проблеми, використовуючи веб-камеру та площини як довідники вставляти віртуальні об'єкти в реальні декорації. Це дозволяє відстеження положення опорних площин (маркерів) відносно до камери, що дозволяє розташовувати віртуальні об'єкти поблизу їх. Крім того, програмне забезпечення дозволяє маніпулювати площинами разом із пов'язаними віртуальними об'єктами за допомогою рук.

Під час виконання програми спочатку ARToolKit перетворює відеозображення, зроблене камерою, в а двійкове зображення з чорно-білими значеннями і досліджує зображення, намагаючись знайти квадратні ділянки. Потім ARToolKit аналізує зображення всередині квадратів (символи еквівалентні до штрих-кодів), що працюють як шаблон, порівнюючи їх з попередньо зареєстровані символи. Отже, якщо є подібність між захопленим символом і попередньо зареєстрованим символом, ARToolKit знайде маркер посилання. Після того, ARToolKit використовує відомі розміри квадрата та

шаблону орієнтація для обчислення положення камери відносно маркер, що визначає координати та орієнтацію маркер, який використовується для точного малювання віртуального об'єкта це [22].

ARToolKit можна використовувати в програмах доповненої реальності за допомогою табличок (еталонних маркерів), де розміщені віртуальні об'єкти розміщуються, коли вони з'являються в області огляду камери. Оскільки ARToolKit має відкритий вихідний код, з'явилися нові функції розроблений авторами даної роботи, щоб розширити її зручність використання та потенціал для розробки нових додатків.

У цьому контексті були розроблені деякі інтерактивні процеси, наприклад: зміна віртуального об'єкта, коли конкретний керуючий маркер входить у простір перегляду веб-камери; копіювання або захоплення та транспортування віртуальних об'єктів між маркерами в сцені, використовуючи транспортування маркери; відтворення звуку, пов'язаного з віртуальними об'єктами, починаючи з моменту розміщення віртуальних об'єктів на маркерах; зміна віртуальних об'єктів в режимі реального часу на основі отриманих зовнішні дані.

1. Контрольні маркери

Такий тип взаємодії дозволяє віртуальним об'єктам закінчитися маркери, які потрібно динамічно контролювати та змінювати, коли керуючий маркер входить у вид простору веб-камери. Кілька віртуальних об'єктів зі списку пов'язаний з маркером, щоб маркер керування міг початися обмін віртуальними об'єктами, видалення віртуального об'єкта і поміщення наступного зі списку над маркером.

Таким чином, кожен маркер може показувати кілька віртуальних об'єктів, залежно від введеної послідовності в просторі перегляду веб-камера. Ця процедура працює як повторне використання маркерів і він корисний для створення послідовності сцен і звуків контрольовані користувачем.

2. Транспортування маркерів

Ця взаємодія заснована на спеціальному маркері (транспорт маркер), що дозволяє копіювати або захоплювати віртуальний об'єкт з маркера та перенесення його на інший маркер, працюючи веслом. Коли транспортний маркер входить до сцени, ARToolKit обчислює відстань від неї до іншої маркери, поки він не стане достатньо близько до певного маркера. У цій ситуації він копіює або передає віртуальний об'єкт пов'язаний із цим маркером до транспортного маркера. Коли транспортний маркер наближається до іншого маркера, останнього отримує віртуальний об'єкт із транспортного маркера шляхом копіювання або перенесення. Для цього необхідно було змінити а частину вихідного коду ARToolKit, щоб зробити можливим визначення відстані між пластинами.

3. Зміна віртуальних об'єктів під час виконання

Можна змінювати віртуальні об'єкти під час виконання програми через зміни в ARToolKit код і доповнення конкретних коментарів до коду VRML що вказує дані, які потрібно змінити. В даному випадку це було необхідно виконати деякі зміни параметрів в код ARToolKit, такий як масштаб, поворот і переклад пов'язані з віртуальними об'єктами. Втім, також можна виконати деякі зміни в коді VRML віртуального об'єкт під час виконання, після чого слід перемалювати за участю цих об'єктів. Це означає, що віртуальні об'єкти слід знову заряджати під час роботи. Код VRML можна змінити під час роботи ARToolKit виконано, але коли змінний маркер входить на сцену, ARToolKit виконує процес перемалювання, показуючи нове віртуальний об'єкт, що відображає зміни.

ARToolKit забезпечує взаємодію не тільки віртуально, а й також людські, тому що учасники можуть грати зі своїми супротивників, без втручання комп'ютера. Можливо включити в процес комп'ютер, допомагаючи переможцю ідентифікація та встановлення того, хто гратиме. Дозволяє користувачам встановити можливості для пошуку рішень і

взаємодіяти з іншими користувачами через спільну діяльність використання колективних стратегій. Рисунок 3.1 ілюструє структуру і середовище гри в хрестики-нолики.

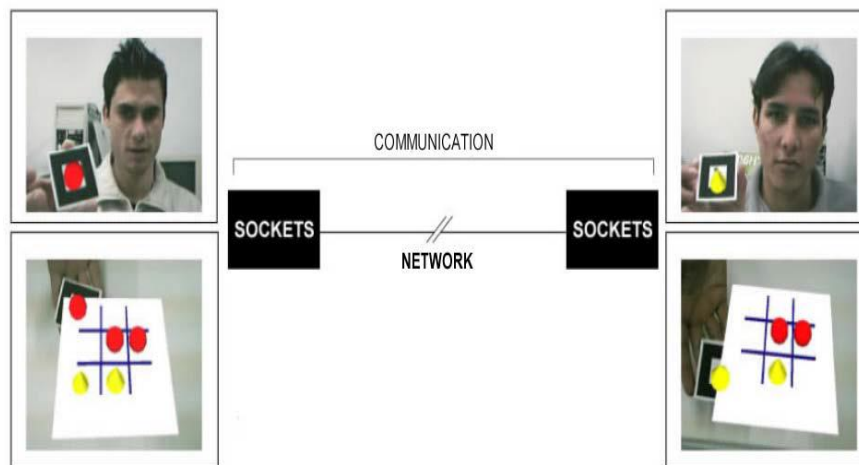


Рисунок. 3.1. - Середовище гри в хрестики-нолики.

Паралельно з діями користувачі можуть спілкуватися через текстові повідомлення у вікні чату, розмовляти голосом канал і переглядати у вікні відео. Залежно від якості мережі та транспортна ситуація, деякі з них комунікаційні елементи можуть бути деактивовані для системи працювати адекватно. Єдиний елемент, якого бути не може disabled — це програма доповненої реальності, яка є сама гра і потенційно вимагає менше трафіку в мережі. Коригування в системі проводилося для того, щоб мінімізувати трафік інформації, пов'язаної з грою. Спочатку інформація про дошку та фігури позиції постійно надсилали, щоб гарантувати послідовність позицій для віддалених користувачів, але це створювало інтенсивні і непотрібний трафік інформації в мережі. До зменшити цей трафік, було розміщено програму для аналізу позиції маркерів, які постійно збираються ARToolKit.

Якщо старі та нові позиції (X, Y або Z) дошки або шматки відрізняються менше, ніж попередньо визначений допуск (0,5 см, для наприклад), інформацію надсилати не потрібно. Інформація буде надіслана лише тоді, коли вона дійсно є значні зміни в розташуванні плати або штук. Таким чином, з невеликим трафіком короткої інформації мережа зазвичай

підтримує доповнену реальність застосування навіть в умовах низької швидкості.

3.1.1 Процес калібрування маркерів за допомогою ARToolKit

Потік шукає файл на диску та виконує його. Перевагою файлів є їх зменшений розмір, який підвищує їх портативність, а недолік пов'язаний до того, що веб-камера не може захватити маркер, у деяких випадках. Рисунок 3.2 демонструє калібрування маркера

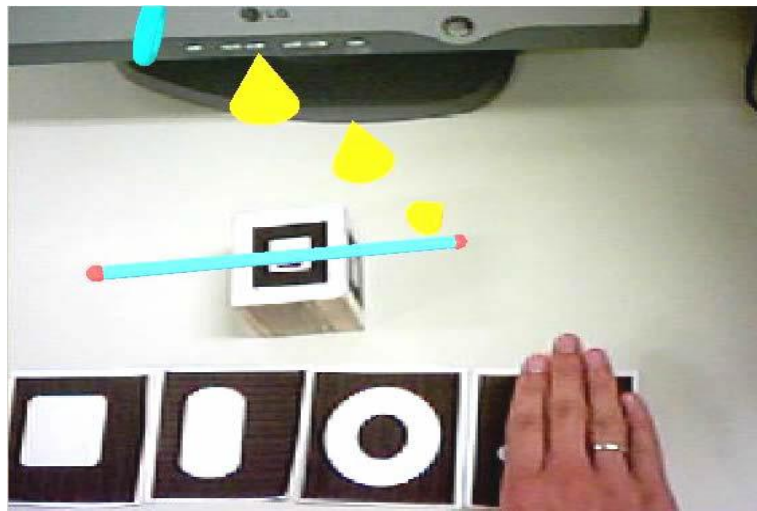


Рисунок. 3.2. Процес калібрування маркера

З процесом калібрування маркерів і деякі раніше розміщені символи, можна отримати великий різноманітність взаємодій користувачів, як створення персоналізованих маркерів .

Однак був розроблений з модулем живих віртуальних об'єктів, які працюють. Кожен віртуальний Об'єкт має формат, схожий на один із раніше розміщених символи. Віртуальна лінійка розміщена в декорації таким чином віртуальні об'єкти можуть жваво спускатися до нього. В даний момент Користувач повинен навести маркер на камеру, тоді об'єкт відбивається від маркера.

3.2 Налаштування апаратної платформи

Після обраної апаратної платформи потрібно провести налаштування Raspberry Pi 1B+ для керування програмним кодом проєкта.

До початку роботи треба завантажити Raspbian на microSDCard , для поставленої задачі було обрано Raspberry Pi Imager.

OS для Raspberry Pi можливо записати через Raspberry Pi Imager. На рисунку 3.3 відображено вигляд меню програми.

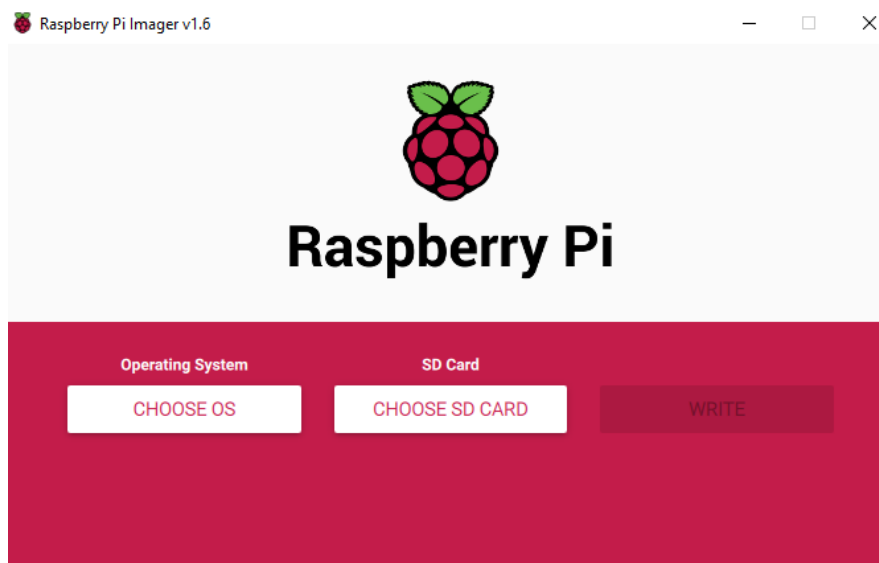


Рисунок 3.3 – Меню Raspberry Pi Imager

Після встановлення системи до міні-комп'ютера треба провести деякі налаштування.

Якщо вам буде запропоновано ввести ім'я користувача та пароль, ім'я користувача за замовчуванням – «pi», а пароль – «raspberrypi». Якщо вас турбує безпека, ви захочете змінити їх. Під час першого завантаження вам, ймовірно, буде надано діалогове вікно «Ласкаво просимо до Raspberry Pi», яке проведе вас через процес вибору важливих налаштувань.

Ви можете керувати більшістю налаштувань Raspberry Pi, наприклад паролем, за допомогою програми «Конфігурація Raspberry Pi», яка знаходиться в меню «Параметри».

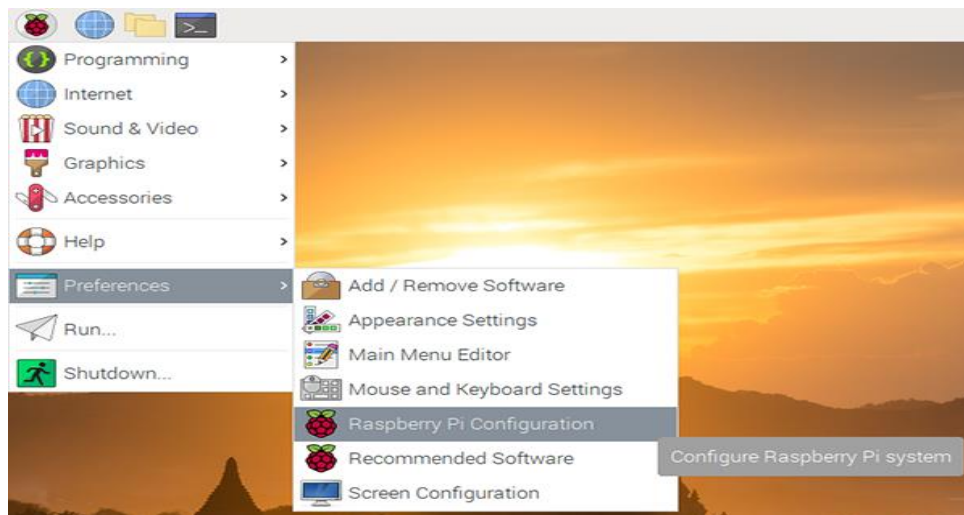


Рисунок 3.3 – Налаштування конфігурації Raspberry Pi

На цій вкладці ви можете змінити основні системні налаштування Raspberry Pi.

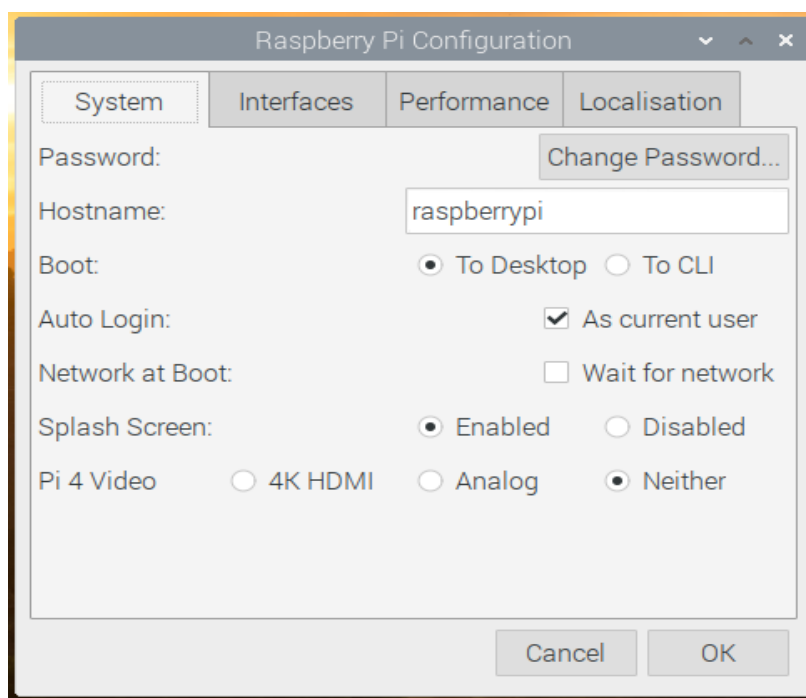


Рисунок 3.4 – Основні налаштування Raspberry Pi

- Пароль — встановіть пароль користувача pi (бажано змінити пароль із заводського за замовчуванням «raspberrypi»)
- Завантаження — виберіть, щоб відображати робочий стіл або CLI (інтерфейс командного рядка) під час запуску Raspberry Pi
- Автоматичний вхід — увімкнення цієї опції змусить Raspberry Pi автоматично входити в систему щоразу, коли він запускається

- Мережа під час завантаження — вибір цієї опції призведе до того, що ваш Raspberry Pi чекатиме, доки мережеве з'єднання стане доступним, перш ніж запускати
- Заставка — виберіть, чи показувати заставку (запуску) під час завантаження Raspberry Pi

Ви можете зв'язати пристрої та компоненти зі своїм Raspberry Pi, використовуючи безліч різних типів з'єднань. На вкладці «Інтерфейси» ви вмикаєте або вимикаєте ці різні з'єднання, щоб ваш Raspberry Pi розпізнавав, що ви щось пов'язали з ним через певний тип з'єднання.

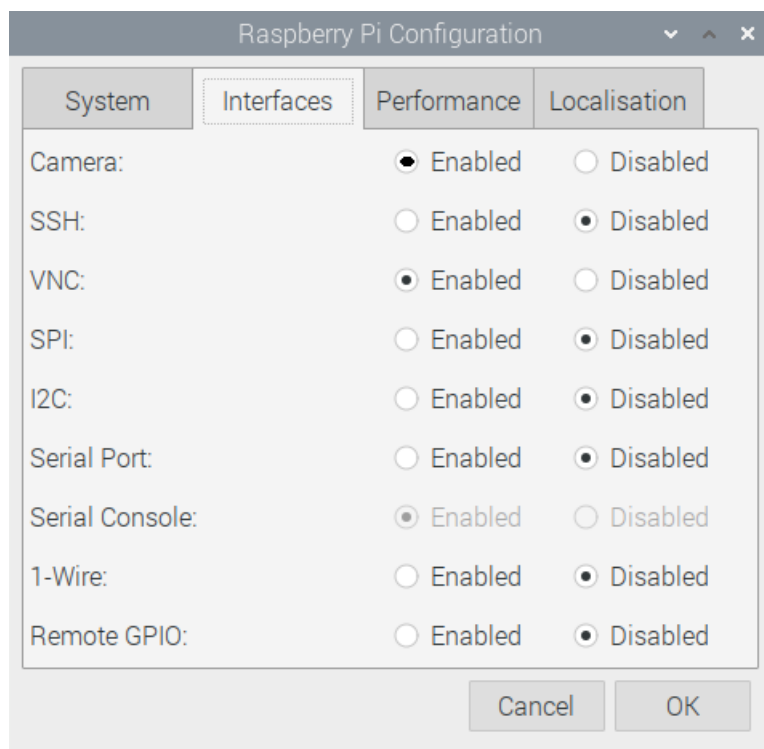


Рисунок 3.5 – Інтерфейси Raspberry Pi

- Камера — увімкніть модуль камери Raspberry Pi
- SSH — надайте віддалений доступ до вашого Raspberry Pi з іншого комп'ютера за допомогою SSH
- VNC — дозволить віддалений доступ до робочого столу Raspberry Pi з іншого комп'ютера за допомогою VNC
- SPI — увімкнути контакти SPI GPIO

- I2C — увімкнути контакти I2C GPIO
- Serial — увімкніть послідовні (Rx, Tx) контакти GPIO
- 1-Wire — увімкніть контакт 1-Wire GPIO
- Віддалений GPIO — надайте доступ до контактів GPIO Raspberry Pi з іншого комп'ютера

Ви можете розігнати свій Raspberry Pi, щоб отримати більше енергії. За замовчуванням параметри розгону Raspberry Pi вимкнено, а частота процесора становить 700 МГц. Однак за допомогою цього інструменту ви можете розігнати його до 1000 МГц. Ці значення будуть змінюватися залежно від версії Raspberry Pi у вас.

Слід також зазначити, що розгін може призвести до більшої нестабільності, а також скоротити термін служби вашого Pi.

3.3 Розробка програмної частини

Для розробки програмної частини було обрано мову Javascript та додаток доповненої реальності для неї ARToolKit.IDE для неї було обрано Atom.

Atom — це сучасний настільний текстовий редактор, на який ви звикли очікувати. Проте відкрийте капот, і ви виявите систему, яка просить зламати.

Що ще важливіше, браузер серйозно обмежує доступ до локальної системи з міркувань безпеки, і для нас текстовий редактор, який не міг записувати файли або запускати локальні підпроцеси, не був початком.

Натомість Atom — це спеціалізований варіант Chromium, розроблений як текстовий редактор, а не веб-браузер. Кожне вікно Atom, по суті, є локальною веб-сторінкою.

Усі API, доступні для типової програми, також доступні для коду, який виконується в контексті JavaScript кожного вікна. Цей гібрид забезпечує унікальний досвід розробки на стороні клієнта.

Оскільки все локально, вам не доведеться турбуватися про конвеєри активів, конкатенацію сценаріїв та визначення асинхронних модулів. Якщо ви хочете завантажити деякий код, просто вимагайте його у верхній частині вашого файлу.

Мережа не позбавлена недоліків, але два десятиліття розвитку перетворили її на неймовірно пластичну та потужну платформу. Тому, коли ми вирішили написати текстовий редактор, який ми б самі хотіли розширити, веб-технологія була очевидним вибором.

Взаємодія з рідним кодом також дуже проста. Наприклад, ми написали обгортку навколо механізму регулярних виразів Oniguruma для підтримки граматики TextMate. У браузері це вимагало б пригод з NaCl або Esprima. Інтеграція вузлів спростила це.

Ще одна велика перевага написання коду для Atom — це гарантія того, що він працює на новітній версії Chromium. Це означає, що ми можемо ігнорувати такі проблеми, як сумісність браузера та полізаповнення. Ми можемо використовувати всі блискучі функції Інтернету завтрашнього дня, сьогодні.

Наприклад, макет нашої робочої області та панелей базується на flexbox. Це новий стандарт, який зазнав багато змін з тих пір, як ми почали його використовувати, але все це не мало значення, поки він працював.

Оскільки вся індустрія просуває веб-технології вперед, ми впевнені, що будуємо Atom на благодатному ґрунті. Нативні технології інтерфейсу користувача з'являються і зникають, але Інтернет — це стандарт, який з кожним роком стає все більш ефективним і поширеним. Ми раді заглибитися в його набір інструментів.

Ми розглядаємо Atom як ідеальне доповнення до основної місії — створення кращого програмного забезпечення шляхом спільної роботи.

3.3.1 Модулі доповненої реальності

Для реалізації доповненої реальності був використаний фреймворк ARToolKit.

Все залежить від кількох факторів — від швидкості переміщення камер/об'єктів та від освітлення. В хорошо освіщенній кімнаті (не професійної студії). У мене вдалося відстежити Маркери з 10 метрів в напрямленні об'єктива в дозволі 720р. Чем вище дозвіл камери, тим вище якість відстеження маркерів. Одну річ, яку варто відзначити: ви можете знімати ваше відео у дозволі 1080, переглядати результати трекінгу та зменшувати число робіт на стороні клієнта. Можна покрутити коефіцієнт, якщо що-то погано відстежується або поріг, якщо кадр був погано освітлений.

Надалі під AR ми маємо на увазі здатність пристрою (мобільного пристрою або веб-браузера) до відстежувати зображення або представляти тривимірний об'єкт поверх зображення. Основна мета AR - показати а комп'ютерна модель онлайн в реальному режимі для встановлення взаємодії користувача в реальному режимі простір і 3D модель у віртуальній. AR може бути як маркерним, так і безмаркерним. У випадку маркера AR пристрій відстежує 2D маркер: коли він знайдений, 3D-об'єкт візуалізується. У безмаркерному варіанті пристрій шукає для плоскої поверхні (стіл, підлога тощо) і помістить туди 3D-об'єкт.

Використовуючи камеру, AR дозволяє візуалізувати створені комп'ютером об'єкти в іграх, маркетингу та інше програмне забезпечення, як у випадку розстановки меблів у вітальні чи примірки нового одягу перед покупкою. Це чудова можливість для бізнесу, оскільки вона демонструє продукт перед а клієнт дійсно купує його. Існують спеціальні пристрої, розроблені для AR, наприклад шоломи та гарнітури, які дозволяють користувачеві це робити зануритися в модельне середовище.

Доповнена реальність забезпечує реальний світ із 3D-моделями.

3.3.2 Використання змінних та функцій

1. Для початку запуску програми треба ініціалізувати певні модулі.

```
var ARController = function(width, height, camera) {  
    var id; #номер гравця  
    var w = width, h = height; #розміри  
    this.orientation = 'landscape'; # положення камери у  
просторі  
    this.listeners = {};
```

2. Ініціалізація модулів JS

```
var canvas; # створення ігрової площадки  
var canvasContext;  
var ballX = 50;  
var ballY = 50;  
var ballSpeedX = 8; #швидкість м'яча у осі X  
var ballSpeedY = 8; #швидкість м'яча у осі Y  
  
var player1Score = 0; #ініціалізація рахунку першого гравця  
var player2Score = 0; #ініціалізація рахунку другого гравця  
const winningScore = 3;  
var winScreen = false;  
  
var paddleRightY = 250;  
var paddleLeftY = 250;  
var paddleHeight = 100;  
var paddleWidth = 10;  
  
window.onload = function() {  
    canvas = document.getElementById('gameCanvas');  
    canvasContext = canvas.getContext('2d'); # ініціалізація маркерів  
  
    var framesPerSecond = 30;  
    setInterval(function() {  
        moveEverything();  
        drawEverything();  
    }, 1000/framesPerSecond);
```

3. Отримання позиціонування камери

```

var self = this;
this.cameraParam = new ARCameraParam(camera, function()#отримання
позиції камери у просторі {
self._initialize();
}, function(err) {
console.error("ARController: Failed to load ARCameraParam", err);
});

} else {

this.cameraParam = camera;#ініціалізація камери
this._initialize();

}
};

```

4.Отримання позицій маркерів.

```

ARController.prototype.process = function(image) {
this.detectMarker(image);

var markerNum = this.getMarkerNum();#отримання номерів маркерів
var k,o;
for (k in this.patternMarkers) {
o = this.patternMarkers[k]
o.inPrevious = o.inCurrent;
o.inCurrent = false;
}

for (k in this.barcodeMarkers) {
o = this.barcodeMarkers[k]
o.inPrevious = o.inCurrent;
o.inCurrent = false;
}

for (var i=0; i<markerNum; i++) {
var markerInfo = this.getMarker(i);

var markerType = artoolkit.UNKNOWN_MARKER;#Маркер не
розпізнається
var visible = this.trackPatternMarkerId(-1);

```

```
        if      (markerInfo.idPatt      >      -1      &&      (markerInfo.id      ===
markerInfo.idPatt || markerInfo.idMatrix === -1)) {
            visible = this.trackPatternMarkerId(markerInfo.idPatt);
            markerType = artoolkit.PATTERN_MARKER;

            if (markerInfo.dir !== markerInfo.dirPatt) {
                this.setMarkerInfoDir(i, markerInfo.dirPatt);
            }

        } else if (markerInfo.idMatrix > -1) {
            visible = this.trackBarcodeMarkerId(markerInfo.idMatrix);#
            markerType = artoolkit.BARCODE_MARKER;

            if (markerInfo.dir !== markerInfo.dirMatrix) {
                this.setMarkerInfoDir(i, markerInfo.dirMatrix);
            }
        }

        if      (markerType      !==      artoolkit.UNKNOWN_MARKER      &&
visible.inPrevious) {
            this.getTransMatSquareCont(i, visible.markerWidth,
visible.matrix, visible.matrix);
        } else {
            this.getTransMatSquare(i, visible.markerWidth, visible.matrix);
        }
    }
```

3.4 Алгоритм роботи застосунку

На рисунку 3.5 зображено розроблений алгоритм роботи застосунку тенісу. При ініціалізації перших модулів починається позиціонування та виявів маркерів у просторі. Ініціалізація юзерів та положення маркерів.

Алгоритм кожної секунди оновлює дані, положення у просторі маркера.



Рисунок 3.5 - Алгоритм роботи застосунку тенісу

3.5 Тестування програмної платформи

Для тестування програмної платформи було запущено Raspberry Pi та розгорнуто сервер для запуску застосунку, було розроблено маркер який служить ракеткою у командній грі.



Рисунок 3.6 – Маркер типу THREEx.

Спочатку ви завантажуєте внутрішнє зображення. Це зображення, яке ви бажаєте розмістити всередині чорного квадрата маркера. Потім ви завантажуєте навчений маркер, натиснувши кнопку завантаження. Вам потрібно буде додати його до свого коду, щоб він знав про цей новий маркер. Файл містить закодований маркер, який потрібно повторно використовувати

у своєму кодї. Ви вказуєте, що вам потрібен маркер шаблону і надаєте URL-адресу свого власного маркера. Тож просто замініть це у своєму кодї.

Для тестування гри було запущено сервер та необхідні модулі для початку гри.



Рисунок 3.7 – Перші запуски гри

AR.js потребує налаштування контексту, викликаючи для цього розширення Three.JS. Тут він приймає файл даних камери, який міститься в папці даних, і виявляє зі швидкістю 30 кадрів на секунду з налаштованими для нього шириною та висотою полотна.

Набір інструментів AR зараз ініціалізовано, і камера в сцені WebGL отримує ту саму матрицю проекції, що і вхідна камера з набору інструментів AR. Набір інструментів AR поміщається в чергу візуалізації, щоб його можна було відображати на екрані кожен кадр.



Рисунок 3.8 – Тестування гри з віддаленою камерою

Спочатку він додається до сцени, а потім використовується разом з набором інструментів AR для виявлення шаблону, який також знаходиться в папці даних. Ще на перших кроках модель завантажувалася та зберігалася у змінній моделі

Нарешті кожен з елементів у черзі візуалізації тепер відображається на екрані. Треба маркер тримати перед собою так, щоб камера могла побачити доповнений вміст.

Висновки до розділу 3

У розділі 3 описано етапи розробки програмної платформи та алгоритми виконання роботи програмної платформи. Було розглянуто та проведено налаштування апаратної платформи Raspberry Pi . Програмний додаток був реалізований на JavaScript з використанням модулю доповненої реальності ARToolKit.

Дані збираються таким чином що сервер отримує дані одного гравця та записує їх в базу даних в хмарному сховищі, посилає запит на сервер щоб отримати дані другого гравця та обробляє їх на сервері. Так визначається позиція маркера який служить для створення віртуальної ракетки та відбиває м'яч на полі ігри.

ВИСНОВКИ

Доповнена реальність значною мірою сприяє області командних ігор, що надає користувачеві привабливу візуалізація та природну взаємодія. Крім того, доповнена реальність не вимагає спеціальних пристроїв і інтенсивного навчання користувачів.

Ця класифікаційна робота була зосереджена на технології доповненої реальності, підкреслюючи ресурси, які пропонує програмне забезпечення ARToolKit. Це програмне забезпечення призначено для впровадження додатків на основі доповненої реальності.

Було досліджено використання доповненої реальності у командних іграх, розробленої за допомогою ARToolKit. Представлені тематичні дослідження, готові до використання, показали, що можливо розробляти ігри з використанням популярних ресурсів, у тому числі обладнанням і програмним забезпеченням, яке підходить для додатків доповненої реальності. це є важливо, щоб дати нові характеристики відомим іграм і розробка нових.

У дослідженні RPi показала себе добре, виконувала поставлені завдання. RP перегрівалась та їй не вистачало охолодження для підтримки стабільної роботи програмного додатку.

Недоліками проєкту є недостатня потужність обраної моделі RPi ,також виявлена нестача оперативної пам'яті для програмної частини. Із-за низького коефіцієнта роботи RPi не було досягнуто повного потенціалу роботи доповненої реальності

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hu Tianyu et al., Overview of augmented reality technology. Computer Knowledge and Technology, 2017(34): 194-196.
2. Wu Fan and Zhang Liang, An overview of the development and application of augmented reality technology. Computer Knowledge and Technology, 2012(34): 8319-8325.
3. Zhou Zhong, Zhou Yi and Xiao Jiangjian, Overview of VR technology. Science in China: Information Science, 2015(02): 157-180
4. Paladini M. 3 Different types of AR explained: Marker-based, Markerless & Location-based AR. Retrieved from <https://www.blippar.com/blog/2018/08/14/marker-based-markerless-or-location-based-ar-different-types-of-ar>. (дата звернення: 01.06.2022)
5. Tatyana Shavel. ARCore vs. ARKit: Which Is Better for Building Augmented Reality Apps. Retrieved from: <https://www.iflexion.com/blog/arcore-vs-arkit>, 2018 (дата звернення: 01.06.2022)
6. Yao A. A workaround for the limitations of ARKit 2. Retrieved from <https://tvt.studio/blog/a-workaround-for-the-limitations-of-arkit-2/>, 2019. (дата звернення: 01.06.2022)
7. Ong SK, Yuan ML, Nee AYC. Augmented reality applications in manufacturing: a survey, International Journal of Production Research 2008; 46: 2707-2742.
8. Michalosa G, Karagiannisa P, Makrisa S, Tokçalarb O, Chryssolourisa G. Augmented reality (AR) applications for supporting human-robot interactive cooperation, Procedia CIRP 41 (2016): 370 – 375.

9. Aziz FA, Alostad E, Sulaiman S, Ahmad KA. Augmented Reality Marker Based to Aid Inspection and Maintenance Process in Automotive Industry, *International Journal of Engineering and Advanced Technology (IJEAT)* 2019; 8, 3S:417-421.
10. Lee S, Akin O. Augmented reality-based computational fieldwork support for equipment operations and maintenance, *Automation in Construction* 2011, 20(4):338–352.
11. Daponte P, Devito L, Picariello F, Riccio M. 2014. State of the art and future developments of the Augmented Reality for measurement applications. *Measurement Journal*, Vol. 57, pp. 53–70.
12. Bergquist R, Stenbeck N. Using Augmented Reality to Measure Vertical Surfaces, Bachelor thesis, Linköping University, 2018.
13. Whitlock M, Hanner E, Brubaker JR, Kane S, Szafir DA. Interacting with Distant Objects in Augmented Reality. *Proceeding of IEEE Virtual Reality 2018*
14. Ro L. AR Measure turns your phone into a virtual measuring tape. Retrieved from: <https://www.curbed.com/2017/6/29/15894556/armeasure-app-augmented-reality-ruler-measuring-tape-ios>, 2018. (дата звернення 03.06.2022)
15. Marsden R. How measuring apps became a reality. Retrieved from: <https://www.thenational.ae/arts-culture/how-measuring-apps-became-a-reality-1.745876>, 2018. (дата звернення: 05.06.2022)
16. Siltanen S. Theory and applications of marker-based augmented reality. *Julkaisija-Utgivare*; 2012.
17. Avola D, Cinque L, Foresti GL, Mercuri C, Pannone D. A Practical Framework for the Development of Augmented Reality Applications by using ArUco Markers, *International Conference on Pattern Recognition Applications and Methods 2016*: 645-654.

18. Prensky M, From Digital Game-Based Learning, Chapter 5- Fun, Play and Games: What Makes Games Engaging, McGraw-Hill, 2001
19. P. Milgram, et al, "Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum," in Proc. Telemultiplicator and Telepresence Technologies - SPIE, 1994, pp. 282-292.
20. C. Kirner, and R. Tori, "Introduction to Virtual Reality, Mixed Reality and Hiper-Reality," in Virtual Reality: Concepts, Technology, and Tendencies, C. Kirner, and R. Tori, Ed. 1st ed. São Paulo, 2004, pp. 3-
21. ARTOOLKIT: <http://www.itl.washington.edu/artoolkit>. (дата звернення 20.06.2022)
22. Пат. на корисну модель CN. Pat. № H04N1/32101 * Display, printing, storage or transmission of additional information, e.g. ID code, date and time or title/ Ju Hee HWANG; заявл. 13.10.2010; опубл. 02.09.2014
23. Пат. на корисну модель CN. Pat. № G09B29/10 * Map spot or coordinate position indicators; Map reading aids/Chi Xiaoyu/Wang Lili/Zhao Nan; заявл. 09.05.2018; опубл. 19.10.2018
24. Пат. на корисну модель CN. Pat. № G06F8/20 * Software design/ Xie Xuehai/Shen Dongming/Zhang Guohui/Song Chen; заявл. 07.08.2020, опубл. 15.07.2021
25. Пат. на корисну модель US. Pat. № US20150286340A1 * Optical detector/ Basf Se; заявл. 19.08.2013, опубл. 08.10.2015

Додаток А

Програмний код для керування

```
var canvasContext;  
var ballX = 50;  
var ballY = 50;  
var ballSpeedX = 8;  
var ballSpeedY = 8;  
  
var player1Score = 0;  
var player2Score = 0;  
const winningScore = 3;  
var winScreen = false;  
  
var paddleRightY = 250;  
var paddleLeftY = 250;  
var paddleHeight = 100;  
var paddleWidth = 10;  
  
window.onload = function() {  
    canvas = document.getElementById('gameCanvas');  
    canvasContext = canvas.getContext('2d');  
  
    var framesPerSecond = 30;  
    setInterval(function() {  
        moveEverything();  
        drawEverything();  
    }, 1000/framesPerSecond);
```

```
        setInterval(function(){
            var mousePos = calculateMousePos();
            paddleLeftY = mousePos.y - (paddleHeight/2);
            console.log(paddleLeftY);
//    setInterval(function() {console.log(mousePos)},1000); -->
        },33);

canvas.addEventListener('mousedown',
    function(evt){
        if(winScreen){
            winScreen = false;
            player1Score = 0;
            player2Score = 0;
        }
    });
}

// left paddle mouse event
function calculateMousePos(){
    var rect = canvas.getBoundingClientRect();
    var root = document.documentElement;
    var mouse = createVector(markerRoot1.position.x,
markerRoot1.position.y, markerRoot1.position.z, camera, 800, 600);
//    console.log(mouse.y); -->
    var mouseY = mouse.y - rect.top - root.scrollTop;
    return {
        y:mouseY
    };
}
```

```
//right paddle (computer)
function computerMovement(){
    var centrePaddleRight = paddleRightY + (paddleHeight/2);
    if (centrePaddleRight < ballY - 35){
        paddleRightY += 7;
    } else if (centrePaddleRight > ballY + 35){
        paddleRightY -= 7;
    }
}

// Reset ball and change direction
function ballReset(){
    if (player1Score > winningScore || player2Score > winningScore){
        winScreen = true;
    }
    ballX = canvas.width / 2;
    ballY = canvas.height / 2;
    ballSpeedX =- ballSpeedX;
}

function moveEverything() {
    if(winScreen){
        return;
    }
    computerMovement();

    ballX += ballSpeedX;
    ballY += ballSpeedY;
}
```



```
if (ballX > canvas.width){
    if( ballY > paddleRightY &&
        ballY < paddleRightY + paddleHeight ){
        ballSpeedX =- ballSpeedX;
        // vary angle of ball when edge of paddle
        deltaY = ballY - (paddleRightY + paddleHeight/2);
        ballSpeedY = deltaY * 0.25;
    } else {
        player1Score ++;
        ballReset();
    }
}
if (ballX < 0){
    if( ballY > paddleLeftY &&
        ballY < paddleLeftY + paddleHeight ){
        ballSpeedX =- ballSpeedX;

        // vary angle of ball when edge of paddle
        deltaY = ballY - (paddleLeftY + paddleHeight/2);
        ballSpeedY = deltaY * 0.25;
    } else {
        player2Score ++;
        ballReset();
    }
}
if (ballY > canvas.height){
    ballSpeedY =- ballSpeedY;
}
```

```
    if (ballY < 0){
        ballSpeedY =- ballSpeedY;
    }
}

// Canvas rectangles
function colourRect(leftX, topY, width, height, bColour){
    canvasContext.fillStyle = bColour;
    canvasContext.fillRect(leftX, topY, width, height);
}

// canvas circle
function colourCircle(centerX, centerY, rad, bColour){
    canvasContext.fillStyle = bColour;
    canvasContext.beginPath();
    canvasContext.arc(centerX, centerY, rad, 0, 2*Math.PI, true);
    canvasContext.fill();
}

// canvas net
function gamenet(){
    for(var i=0; i<canvas.height; i+=20){
        colourRect(canvas.width/2-1, i, 2, 10, 'white');
    }
}

// Canvas parameters
function drawEverything() {
```

```
if(winScreen){
    colourRect(0, 0, canvas.width, canvas.height, 'black');
    canvasContext.font="30px Arial";
    canvasContext.fillStyle = 'white';
    canvasContext.fillText("Click to Start Game", 260, canvas.height/2);

    if (player1Score > winningScore){
        canvasContext.fillText("Left Player Won", 280, 100);
    } else if (player2Score > winningScore){
        canvasContext.fillText("Right Player Won", 280, 100);
    }

    return;
}

colourRect(0, 0, canvas.width, canvas.height, 'black');
// net
gamenet();
// left paddle
colourRect(0,paddleLeftY, paddleWidth, paddleHeight, 'white');
// right paddle
colourRect(canvas.width - paddleWidth, paddleRightY, paddleWidth,
paddleHeight, 'white');
// ball
colourCircle(ballX, ballY, 10, 'red');

//scores
canvasContext.font="30px Arial";
canvasContext.fillText(player1Score, 100, 100);
```

```
canvasContext.fillText(player2Score, canvas.width-120, 100)
}

function createVector(x, y, z, camera, width, height) {
    var p = new THREE.Vector3(x, y, z);
    var vector = p.project(camera);

    vector.y = -(vector.y - 1) / 2 * height;

    return vector;
}
```

Додаток Б

Програмний код

```
var ARController = function(width, height, camera) {  
    var id;  
    var w = width, h = height;  
  
    this.orientation = 'landscape';  
  
    this.listeners = {};  
  
    if (typeof width !== 'number') {  
        var image = width;  
        camera = height;  
        w = image.videoWidth image.width;  
        h = image.videoHeight image.height;  
        this.image = image;  
    }  
  
    this.defaultMarkerWidth = 1;  
    this.patternMarkers = {};  
    this.barcodeMarkers = {};  
    this.transform_mat = new Float32Array(16);  
  
    this.canvas = document.createElement('canvas');  
    this.canvas.width = w;  
    this.canvas.height = h;  
    this.ctx = this.canvas.getContext('2d');  
  
    this.videoWidth = w;
```

```
this.videoHeight = h;

if (typeof camera === 'string') {

    var self = this;
    this.cameraParam = new ARCameraParam(camera, function() {
        self._initialize();
    }, function(err) {
        console.error("ARController: Failed to load ARCameraParam", err);
    });

} else {

    this.cameraParam = camera;
    this._initialize();

}

};

ARController.prototype.process = function(image) {
    this.detectMarker(image);

    var markerNum = this.getMarkerNum();
    var k,o;
    for (k in this.patternMarkers) {
        o = this.patternMarkers[k]
        o.inPrevious = o.inCurrent;
        o.inCurrent = false;
    }
    for (k in this.barcodeMarkers) {
```

```
o = this.barcodeMarkers[k]
o.inPrevious = o.inCurrent;
o.inCurrent = false;
}

for (var i=0; i<markerNum; i++) {
    var markerInfo = this.getMarker(i);

    var markerType = artoolkit.UNKNOWN_MARKER;
    var visible = this.trackPatternMarkerId(-1);

    if (markerInfo.idPatt > -1 && (markerInfo.id === markerInfo.idPatt ||
markerInfo.idMatrix === -1)) {
        visible = this.trackPatternMarkerId(markerInfo.idPatt);
        markerType = artoolkit.PATTERN_MARKER;

        if (markerInfo.dir !== markerInfo.dirPatt) {
            this.setMarkerInfoDir(i, markerInfo.dirPatt);
        }

    } else if (markerInfo.idMatrix > -1) {
        visible = this.trackBarcodeMarkerId(markerInfo.idMatrix);
        markerType = artoolkit.BARCODE_MARKER;

        if (markerInfo.dir !== markerInfo.dirMatrix) {
            this.setMarkerInfoDir(i, markerInfo.dirMatrix);
        }
    }
}
```

```
        if (markerType !== artoolkit.UNKNOWN_MARKER &&
visible.inPrevious) {
            this.getTransMatSquareCont(i, visible.markerWidth, visible.matrix,
visible.matrix);
        } else {
            this.getTransMatSquare(i, visible.markerWidth, visible.matrix);
        }

visible.inCurrent = true;
this.transMatToGLMat(visible.matrix, this.transform_mat);
this.dispatchEvent({
    name: 'getMarker',
    target: this,
    data: {
        index: i,
        type: markerType,
        marker: markerInfo,
        matrix: this.transform_mat
    }
});
}

var multiMarkerCount = this.getMultiMarkerCount();
for (var i=0; i<multiMarkerCount; i++) {
    var subMarkerCount = this.getMultiMarkerPatternCount(i);
    var visible = false;

    artoolkit.getTransMatMultiSquareRobust(this.id, i);
    this.transMatToGLMat(this.marker_transform_mat, this.transform_mat);
```



```
for (var j=0; j<subMarkerCount; j++) {
  var multiEachMarkerInfo = this.getMultiEachMarker(i, j);
  if (multiEachMarkerInfo.visible >= 0) {
    visible = true;
    this.dispatchEvent({
      name: 'getMultiMarker',
      target: this,
      data: {
        multiMarkerId: i,
        matrix: this.transform_mat
      }
    });
    break;
  }
}
if (visible) {
  for (var j=0; j<subMarkerCount; j++) {
    var multiEachMarkerInfo = this.getMultiEachMarker(i, j);
    this.transMatToGLMat(this.marker_transform_mat,
this.transform_mat);
    this.dispatchEvent({
      name: 'getMultiMarkerSub',
      target: this,
      data: {
        multiMarkerId: i,
        markerIndex: j,
        marker: multiEachMarkerInfo,
        matrix: this.transform_mat
      }
    });
  }
}
```

```
    });  
  }  
}  
}  
if (this._bwpointer) {  
  this.debugDraw();  
}  
};
```

```
ARController.prototype.trackPatternMarkerId = function(id, markerWidth)  
{  
  var obj = this.patternMarkers[id];  
  if (!obj) {  
    this.patternMarkers[id] = obj = {  
      inPrevious: false,  
      inCurrent: false,  
      matrix: new Float32Array(12),  
      markerWidth: markerWidth || this.defaultMarkerWidth  
    };  
  }  
  if (markerWidth) {  
    obj.markerWidth = markerWidth;  
  }  
  return obj;  
};
```

Додаток В

Блок-схема алгоритму

