

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

Рівень вищої освіти другий (магістерський)
(рівень вищої освіти)
Дослідження технології доповненої реальності.
Використання AR в освітньому процесі
(тема)

Виконав:
студент 2 курсу, групи СШМ-18-2
Берестовий О.О
(прізвище, ініціали)

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

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

Освітня програма Системи штучного інтелекту
(СШІ)
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ другий (магістерський) _____

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

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

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

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____» _____ 2020 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Берестовому Олександрю Олеговичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження технології доповненої реальності. Використання AR в освітньому процесі

затверджена наказом по університету від 03.03.2020 р. № 480 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19.05.2020 р.

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

4. Перелік питань, що потрібно опрацювати в роботі аналіз дидактичних матеріалів за доповненої реальності, аналіз ігрового ядра Unity та його компонентів, аналіз засобів розробки, розробка додатку

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Ідентифікація хімічного елементу Сl, операційна система Android, Рисунок А.2 – Ідентифікація двох елементів «Н» та одного елементу «О» і формування молекули «Н₂О» з реальним уявленням, Рисунок А.3- Ідентифікація двох хімічних елементів «Н»

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Аналіз предметної галузі	доц. Чала Л.Е.		31.03.2020
Формування вимог до додатку	доц. Чала Л.Е.		13.04.2020
Розробка додатку	доц. Чала Л.Е.		22.04.2020
Аналіз готового продукту	доц. Чала Л.Е.		23.04.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	31.03.20 – 15.04.20	виконано
2	Формування вимог до додатку	13.04.2020	виконано
3	Засоби розробки додатку	14.04.2020	виконано
4	Програмна реалізація додатку	15.04.20 – 22.04.20	виконано
5	Аналіз готового продукту	23.04.2020	виконано
6	Написання пояснювальної записки	24.04.20 – 08.05.20	виконано
7	Попередній захист	14.05.2020	виконано
8	Захист перед ЕК	19.05.2020	

Дата видачі завдання _____ 2020 р.

Студент _____
(підпис)

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

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить: 88 с., 38 рис., 2 дод., 23 джерела.

ДОДАТОК, ASSETS, AUGMENTED REALITY, UNITY, VUFORIA, EMGUCV, FAST АЛГОРИТМ

Об'єкт дослідження – методи розробки додатків з використанням технології доповненої реальності для платформ Android, IOS та Windows.

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

Області застосування – навчальний процес.

Методи дослідження – аналіз технічної літератури, методи розпізнавання образів, методи створення технології доповненої реальності.

Результат атестаційної роботи – додаток, що використовує технологію доповненої реальності, який пройшов апробацію.

ABSTRACT

The Evaluation Explanatory Note contains: 88 pages, 38 pictures, 2 applications, 23 sources.

APPLICATION, AUGMENTED REALITY, UNITY VUFORIA,
EMGUCV, FAST ALGORITM

Object of research – methods of application development using augmented reality technology for Android, IOS and Windows platforms.

Purpose of work – development and implementation of AR project for mobile devices, using the graphic and physical core of Unity.

Scopes of application – educational process.

Methods of research – analysis of technical literature, analysis of methods of augmented reality technology creation.

The result of certification work – the product of the work was the application, which has been tested.

РЕФЕРАТ

Пояснительная записка к аттестационной работы содержит: 88 с., 38 рис., 2 приложения, 23 источника.

ПРИЛОЖЕНИЕ, ASSETS, AUGMENTED REALITY, UNITY VUFORIA, EMGUCV, FAST АЛГОРИТМ

Объект исследования – методы разработки приложений с использованием технологии дополненной реальности для платформ Android, IOS и Windows.

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

Области применения – образовательный процесс.

Методы исследования – анализ технической литературы, методы распознавания изображений, методы создания технологии дополненной реальности.

Результат аттестационной работы – приложение, использующее технологию дополненной реальности, прошедшее апробацию.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	7
Вступ.....	8
1 Аналіз предметної галузі та постановка задачі.....	10
1.1 Аналіз предметної галузі.....	10
1.2 Постановка задачі.....	20
2 Теоретичні дослідження.....	22
2.1 Принципи побудови доповненої реальності.....	22
2.2 Типи систем доповненої реальності.....	26
2.3 QSAR.....	29
2.4 FAST алгоритм для знаходження кутів.....	31
3 Експериментальні дослідження.....	35
4 Розробка додатку.....	41
4.1 Засоби розробки.....	41
4.1.1 Ігровий рушій.....	41
4.1.2 Бібліотека доповненої реальності.....	48
4.2 Розробка додатку.....	51
4.2.1 Інсталяція Unity.....	52
4.2.2 Інсталяція .Net Chemistry Development Kit, імплементація.....	53
4.2.3 Інсталяція EmguCV, імплементація.....	56
4.2.4 Підготовка зображень та завершення додатка.....	57
5 Аналіз реалізованого продукту.....	61
Висновки.....	66
Перелік посилань.....	68
Додаток А.....	71
Додаток Б.....	75
Додаток В.....	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AR – це технологія інтерактивної візуалізації, яка доповнює зображення реального світу віртуальними елементами.

QSAR – пошук кількісних співвідношень «структура-властивість» в хімічній галузі – це процедура побудови моделей, що дозволяють по структурам хімічних сполучень передбачати їх різноманітні властивості.

FAST – особливості прискорених випробовувань сегмента.

NCDK – це набір бібліотек для платформи .Net з відкритим кодом, що постачає функціонал для роботи з хімією.

Ізотопи – різновиди атомів (і ядер) будь-якого хімічного елемента, які мають однаковий атомний (порядковий) номер, але при цьому різні масові числа.

ВСТУП

На сьогоднішній день є безперечним стрімкий розвиток інформаційних технологій. Зокрема, на численних заходах (конференціях, семінарах, виставках тощо), присвячених інформаційним технологіям, презентується величезна кількість нових ідей та винаходів. Масштаб розвитку окремих наукових досліджень та практичних здобутків є таким, що потенціальні користувачі не завжди готові сприймати їх у повній мірі, а чимало сміливих рішень так і залишаються недостатньо затребуваними. Через це навчальну систему необхідно постійно вдосконалювати, використовуючи нові підходи та технології до підготовки спеціалістів. Однією з таких технологій є так звана доповнена реальність (Augmented reality – AR) [1]. Ця технологія може бути застосована у багатьох практичних сферах. Наприклад, використовуючи смартфон, обладнаний камерою, можна без проблем знаходити необхідний маршрут руху, навіть якщо у користувача є проблеми з орієнтацією на карті. Такий результат може бути досягнутий завдяки синхронній роботі систем глобального позиціонування, завантаженим на смартфон картам, і алгоритмам, що зможуть прив'язати дані з карти на зображення, яке отримується з камери смартфона. Як результат ми зможемо побачити виділений шлях (наприклад, кольоровою лінією), що буде проектуватися у реальність і вказуватиме користувачеві необхідний напрям руху.

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

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

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

В даній атестаційній роботі досліджуються актуальні завдання розвитку технологій доповненої реальності та їх впровадженню в освітній процес.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

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

Доповнена реальність є революційною технологією в області взаємодії з цільовою аудиторією за допомогою доповнення реального оточення образами та інформацією з віртуальної реальності. Суть технології полягає в тому, що при наведенні камери пристроїв на тригер або маркери доповненої реальності користувачеві відкривається інтерактивний контент, тобто користувач бачить віртуальний об'єкт або інформаційний блок (в залежності від конкретного випадку), яким можна керувати в реальному просторі. За допомогою доповненої реальності виробники зможуть вигідно підкреслити гідності своєї продукції, або продукції замовника, надати користувачеві інформацію про продукцію безпосередньо в момент вибору і в момент прийняття рішення про покупку. Шари доповненої реальності дають можливість користувачам досліджувати повнофункціональну 3D-модель будь-якої продукції або наочно демонструвати весь цикл надання тієї чи іншої послуги. Наприклад, користувач може побачити, як буде виглядати майбутній ремонт у щойно купленій квартирі, при цьому він отримує максимально реалістичні відчуття від продукту або послуги, а також отримує інформацію, яка підтверджує якість товару або послуги. Ця технологія

відкриває необмежену кількість нових можливостей у просуванні бренду, в маркетингових і PR-комунікаціях.

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

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

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

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

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

AR-технологія істотно розширює область даних, які будуть сприйматися користувачем. Актуальність цього питання пов'язана з

- доступністю інформації в реальному часі;

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

Кількість компаній, що використовують AR, зростає з кожним днем. За оцінками аналітиків, сукупний обсяг ринку доповненої реальності вже перевищив 5 млрд. доларів. Markets and Markets оцінює середньорічне зростання глобального ринку з 2013 по 2020 роки в 132% і більше [1].

Основні характеристики, властиві середовищу доповненої реальності:

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

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

На даний момент можливо виділити декілька перспективних ринкових ніш, де використання доповненої реальності найбільш добре проглядається. Це, перш за все, маркетингові комунікації, комерція, навчання і обслуговування, військова сфера, сфера медицини, сфера розваг і дозвілля тощо.

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

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

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

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

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

У сучасній військовій техніці використовуються індикатори, які надають важливу інформацію для швидкого аналізу ситуації, розрахунку траєкторії пострілу, розрахунку траєкторії приземлення у пілотів.

Перспективи використання технологій доповненої реальності в сфері дозвілля і розваг безмежні, адже програми, що використовують AR, можуть бути дуже видовищними, цікавими і незвичайними.

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

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

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

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

Крім названих плюсів і ефекту від використання доповненої реальності, варто згадати про матеріальні переваги, адже відпадає необхідність у виробництві і використанні будь-якого виду поліграфії, друкованої продукції. Камера, двовимірний маркер, який розміщується перед камерою для зчитування та аналізу з нього інформації, – це все, що необхідно для отримання ефекту доповненої реальності.

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

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

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

Величезна кількість способів, методів, технологій продажів, видів реклами, способів залучення уваги до продукту і стимулювання попиту

привела до того, що в комерційній сфері все складніше донести інформаційний посил до цільової аудиторії, все складніше акцентувати увагу саме на своєму товарі або послугі. Саме жорстка ринкова конкуренція, боротьба за клієнта, необхідність виділитися серед величезного асортименту товарів і послуг вимагає нових високотехнічних мультимедійних інструментів для посилення ефекту інформаційних суспільств. Дана ситуація в комерційній сфері вимагає інноваційного підходу, а, значить, ефективних засобів інноваційних комунікацій.

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

Існує багато способів створювати доповнену реальність, які можуть суттєво відрізнятись за своїм функціоналом. Найбільш поширеним з них є QR-код, за допомогою якого здійснюється швидкий перехід до інформації або джерел інформації. Наступним способом є спосіб шарів – спосіб, коли до об'єктів місцевості прив'язуються інформаційні шари і за допомогою них відбувається перехід до джерел інформації. Третім способом є аура: при наведенні смартфона на будь-який носій інформації, розпізнається картинка і поверх картини з'являється аура – це може бути відео,

інформаційне повідомлення або фотографія, а також будь-який тривимірний об'єкт.

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

Окуляри доповненої реальності «GOOGLE Glass» розробляються компанією Google і є автономним пристроєм, керувати яким можливо за допомогою голосу і жестів. Принцип роботи окулярів доповненої реальності набагато відрізняється від пристроїв віртуальної реальності. Якщо окуляри віртуальної реальності лише умовно носять свою назву «окуляри», то даний пристрій дійсно є окулярами.

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

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

Дана розробка пов'язана з використанням голограм, які інтегруються з об'єктами навколишнього середовища. Користувач може не

тільки спостерігати за тим, що відбувається, а й взаємодіяти з голограмами інтерфейсу за допомогою рук.

Також за допомогою даного пристрою доповненої реальності можливо проектувати відео або будь-який інший презентаційний матеріал на будь-яку поверхню. Пристрій доповненої реальності є повністю бездротовим, легким і зручним в експлуатації. Також окуляри відтворюють об'ємний звук і мають вбудовані камери. Найбільш перспективною сферою їх застосування є архітектура і будівництво. Для презентацій масштабних будівельних об'єктів даний шолом представляє особливу цінність, так як дозволяє розглянути всі особливості архітектурних рішень, найменші об'єкти споруд будь-якої складності, донести інформацію про користь і необхідність тих чи інших рішень в різних будівельних конструкціях.

Окуляри доповненої реальності SpaceGlasses META.01 комплектуються інфрачервоними камерами для розпізнавання рухів, а також чохлам для смартфона Google Cardbox. Визначення положення і формування доповненої реальності здійснюються програмно-апаратними засобами смартфона.

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

Таким чином, сучасний світ вимагає інтерактивності в комунікації, динаміки і, як наслідок, особливого просторового бачення. Саме ці вимоги дозволяє реалізувати технологія доповненої реальності як інструмент інноваційних комунікацій.

Ще однією особливістю доповненої реальності є простота використання. Для того, щоб бачити віртуальні шари, що доповнюють

реальність, не обов'язково мати спеціальні окуляри або шолом, досить звичайного смартфона і підключення до мережі Інтернет.

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

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

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

Технології доповненої і віртуальної реальності презентують інтерактивні інноваційні технічні рішення, які відкривають світ нових можливостей для соціальних суб'єктів. Використання даних технологій як засобів інноваційних комунікацій безмежно. Ефективність використання даних технологій досягається за рахунок «ефекту занурення» і максимального залучення користувача в процес. Поняття «доповнена реальність» визначається як «технологія інтерактивної візуалізації, яка доповнює зображення реального світу віртуальними елементами».

1.2 Постановка задачі

Метою даної атестаційної роботи є дослідження технології доповненої реальності та розробка AR-додатку для допомоги в освітньому процесі у галузі хімії. Згідно з метою у роботі мають бути вирішені такі задачі:

- загальний аналіз сучасних досліджень за темою роботи;
- дослідження методу Feature Detection [10];
- програмна реалізація алгоритму FAST;
- розробка додатку для автоматичної ідентифікації зображень в відео потоці з можливістю подальшої прив'язки до них інтерактивних об'єктів і маніпуляції над інтерактивними об'єктами.

Абстрактна візуалізація роботи програми «Augmented Reality» представлена на рис. 1.1.

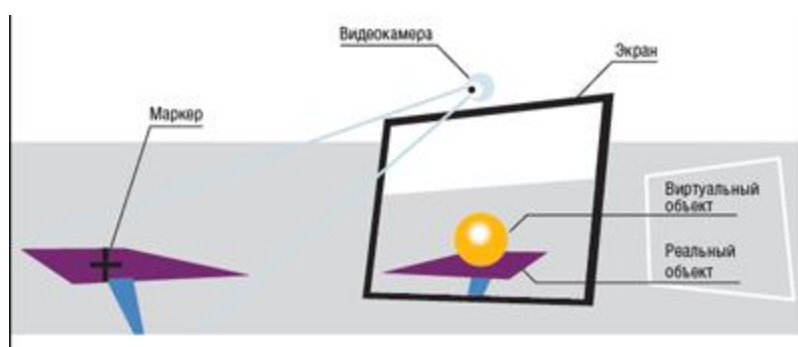


Рисунок 1.1 – Візуалізація роботи «Augmented Reality»

У даному випадку проект розділений на три частини AR-модуль системи, де мають здійснюватися розрахунки, пов'язані з використанням доповненої реальності; Core – графічний рушій для реалізації програми; App – власне сам додаток, що об'єднує систему доповненої реальності і графічний рушій.

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

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

Тестування додатку має здійснюватися на прикладі освітнього процесу в галузі вивчення хімічних дисциплін. Зокрема, додаток реалізовуватиме ідентифікацію серії карток з зображеннями хімічних елементів. При цьому необхідно підготувати базу даних зображень, які згодом будуть служити маркерами. У разі ідентифікації в реальному часі на екрані смартфона будуть з'являтися віртуальні об'єкти у вигляді хімічних елементів які матимуть можливість взаємодіяти між собою за законами хімії.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Принципи побудови доповненої реальності

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

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

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

Найчастіше в ролі маркера виступає аркуш паперу з деяким спеціальним зображенням, також маркером можуть бути і геометричні фігури, тривимірні об'єкти, і навіть очі та обличчя людей.

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

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

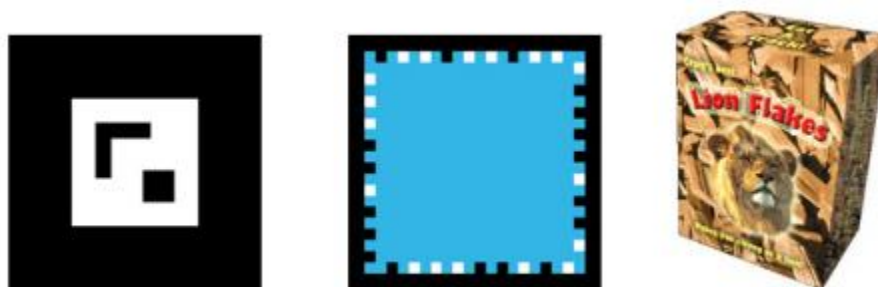


Рисунок 2.1 – Приклади маркерів

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

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

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

ньому ключових особливостей. Дані особливості можуть бути як у вигляді ізольованих точок, так і кривих або пов'язаних областей. Не існує строгого визначення того, що таке ключова особливість зображення. Кожен алгоритм може використовувати свої специфічні ключові особливості (кути, межі, області тощо).

Найчастіше для пошуку маркерів виконується пошук і порівняння зображень по ключових точках. Ключова точка – це певна ділянка, яка є характерною для заданого зображення. Приклади ключових точок на зображенні наведено на рисунку 2.2.



Рисунок 2.2 – Приклади ключових точок на зображенні

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

- детектор (здійснює пошук ключових точок на зображенні, тобто безпосередньо feature detector);
- дескриптор (здійснює опис знайдених ключових точок, оцінюючи їх позиції через опис навколишніх областей);

- матчер (здійснює побудову відповідності між двома наборами точок).

Спочатку за допомогою детектора проводиться пошук ключових точок шаблонного (шуканого) зображення. Отримані точки потім описуються за допомогою дескриптора. Дана інформація зберігається в окремий файл (або базу даних), щоб не виконувати цей процес повторно. При обробці відео потоку з метою пошуку заданого шаблону описаний процес виконується для кожного кадру (за винятком збереження даних). Для встановлення відповідності між ключовими точками і дескрипторами застосовується матчер (алгоритм порівняння). Приклад відповідності точок наведено на рисунку 2.3.

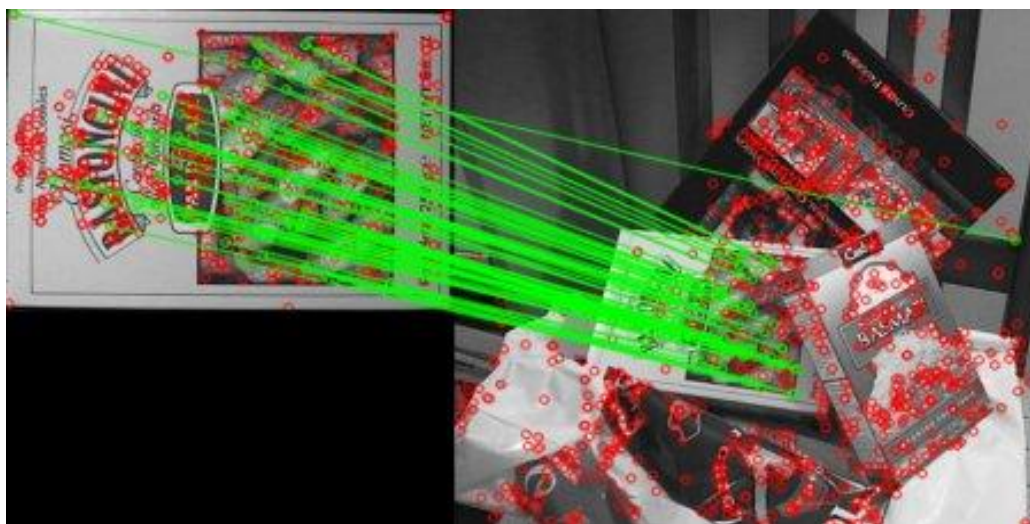


Рисунок 2.3 – Відповідності між точками шаблону та тестового зображення

Природно припустити, що різні алгоритми працюють з різною швидкістю та ефективністю. В умовах їх застосування для побудови доповненої реальності необхідно використовувати тільки ті, які показують високу швидкість роботи при досить хорошій якості відстеження позицій ключових точок. В іншому випадку ми можемо отримати помітні відставання у відеоданих, що знімаються. Для підвищення швидкості

роботи алгоритмів роботи feature points detection застосовуються різні способи фільтрації точок, щоб мінімізувати їх число і відсіяти зовсім невдалі поєднання. Таким чином можна домогтися не тільки підвищення швидкості роботи, але і якості трекінгу маркерів.

2.2 Типи систем доповненої реальності

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

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

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

Аудіовізуальні системи. Це комбінація двох попередніх типів, однак, аудіо інформація в них має лише допоміжний характер. Системи доповненої реальності завжди потребують інформації, одержуваної з навколишнього середовища. Саме на основі цих даних будуються віртуальні об'єкти. Кожна з таких систем має певний набір сенсорів – пристроїв, що дозволяють збирати інформацію з навколишнього середовища за допомогою звукових і електромагнітних коливань, прискорень тощо. Для класифікації має сенс розділяти сенсори не за

типами реєстрованих фізичних величин, а за їх призначенням, оскільки подібні за своєю природою сигнали можуть нести різну інформацію. За типом сенсорів можна виділити наступні системи:

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

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

Системи доповненої реальності можна розрізнити за ступенем взаємодії з користувачем. У деяких системах користувач грає пасивну роль, він лише спостерігає за реакцією системи на зміни в навколишньому середовищі. Інші ж системи вимагають активного втручання користувача – він може управляти як роботою самої системи, для досягнення результатів, так і змінювати віртуальні об'єкти.

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

Інтерактивні системи. Такі системи засновані на взаємодії з користувачем, де на різні дії користувач отримує різну відповідь. У подібних системах необхідно мати пристрій введення інформації. В якості такого пристрою може застосовуватися сенсорний екран мобільного телефону, планшет або спеціальний маніпулятор. Вибір пристроїв введення залежить від специфіки системи. У разі простих дій з віртуальним об'єктом, достатньо простого вказівного пристрою. Якщо ж

необхідна імітація будь-яких реальних процесів і виконання складних маніпуляцій з об'єктами використовуються спеціальні маніпулятори, які мають різну кількість ступенів свободи. Прикладом таких маніпуляторів можуть служити пристрої PHANTOM [6].

Інтерактивність систем визначається по-різному. Існують системи, що дозволяють користувачеві активно змінювати віртуальне середовище (зазвичай це системи-стимулятори будь-яких реальних дій) [7]. Вони використовуються у разі, коли використання реальних об'єктів неможливо, наприклад, спеціалізовані медичні тренажери, що дозволяють початківцям лікарям відпрацьовувати необхідні навички [8].

Існують інші системи, де користувачеві не потрібно змінювати віртуальне середовище. Замість цього користувач обирає, які віртуальні об'єкти він хоче побачити. Користувач також має можливість маніпулювати віртуальними об'єктами, але не на рівні структури, а на рівні зображення, тобто застосовувати, наприклад, афінні перетворення типу повороту, переміщення тощо. До даної групи можна віднести різні архітектурні системи, що дозволяють побачити, як впишеться в реально існуючу обстановку нова споруда або його частина, а також навігаційні та геоінформаційні системи. Подібні системи можуть показувати частини об'єктів інтересу, приховані іншими будівлями, додаткову інформацію про обрані об'єктах тощо. За ступенем мобільності системи доповненої реальності можна класифікувати як:

- стаціонарні системи, призначені для роботи в фіксованому місці; переміщення таких систем означає часткове або повне припинення їх працездатності;

- мобільні системи, що можуть без зусиль переміщатися; часто таке переміщення лежить в основі виконуваної ними функції [8].

Належність до того чи іншого типу визначається функціями системи. Так, симулятор хірургічного столу не повинен бути мобільним, оскільки його завдання – відтворити для людини спеціальні умови,

максимально наближені до реальних. У той же час навігаційна система повинна бути якомога більш мобільною, щоб вона могла переміщатися разом з транспортним засобом або людиною, не створюючи додаткових витрат на її переміщення.

2.3 QSAR

Пошук кількісних співвідношень «структура-властивість» в хімічній галузі – це процедура побудови моделей, що дозволяють по структурам хімічних сполучень передбачати їх різноманітні властивості. За моделями, що дозволяють прогнозувати кількісні характеристики біологічної активності, історично закріпилося англomовна назва Quantitative Structure-Activity Relationship (QSAR). Аббревіатура QSAR часто трактується розширено для позначення будь-яких моделей «структура-властивість». Моделі, що дозволяють прогнозувати фізичні і фізико-хімічні властивості органічних сполучень, мають англomовну назву Quantitative Structure-Property Relationship (QSPR). При якісному описі співвідношень між структурами хімічних сполучень і їх біологічною активністю вживають англomовний термін Structure-Activity Relationship (SAR) [11].

Пошук кількісних співвідношень «структура-властивість» заснований на застосуванні методів математичної статистики і машинного навчання для побудови моделей, що дозволяють за описом структур хімічних сполук передбачати їх властивості (фізичні, хімічні, біологічні активності). При прогнозуванні властивостей на якісному рівні (наприклад, чи буде дана хімічна сполука володіти даним видом біологічної активності) використовують рішення класифікаційної задачі, тоді як при прогнозуванні числових значень властивостей мають на увазі рішення регресійної задачі. Опис структур хімічних сполук для цих цілей може бути векторним або неекторним (графовим).

Існуючі набори молекулярних дескрипторів можуть бути умовно розділені на наступні категорії [12]:

- фрагментні дескриптори, що існують в двох основних варіантах – бінарному і цілочисельному [13]. Бінарні фрагментні дескриптори показують, міститься даний фрагмент (підструктура) в структурній формулі (тобто міститься даний підграф в молекулярному графі, що описує дана хімічна сполука), тоді як цілочисельні фрагментні дескриптори показують, скільки раз даний фрагмент (підструктура) міститься в структурній формулі (тобто скільки разів міститься даний підграф в молекулярному графі, що описує дана хімічна сполука);

- фізико-хімічні дескриптори – це числові характеристики, отримані у результаті моделювання фізико-хімічних властивостей хімічних сполук, або величини, які мають чітку фізико-хімічну інтерпретацію. Найбільш часто використовуються такі дескриптори цього типу: ліпофільність (LogP), молярна рефракція (MR), молекулярна вага (MW), молекулярні об'єми і площі поверхонь;

- квантово-хімічні дескриптори – це числові величини, отримані в результаті квантово-хімічних розрахунків. Найбільш часто в якості таких дескрипторів використовуються: енергії граничних молекулярних орбіталей (ВЗМО і НСМО), часткові заряди на атомах і часткові порядки зв'язків, індекси реакційної здатності;

- дескриптори молекулярних полів – це числові величини, що апроксимують значення молекулярних полів шляхом обчислення енергії взаємодії пробного атома, поміщеного в вузол решітки, з поточною молекулою.

Приклади прогнозування фізико-хімічних властивостей органічних сполук:

- фізичні властивості індивідуальних низькомолекулярних з'єднань;
- температура кипіння;
- критична температура;

- в'язкість;
- тиск насиченої пари;
- щільність.

Ці та деякі інші властивості можна прогнозувати з використанням методів математичної статистики та машинного навчання. В цілому сфера прогнозування у області фізико-хімічних наук є дуже складною та потребує використання додаткових інтелектуальних методів та технологій (зокрема, технологій доповненої реальності).

2.4 FAST алгоритм для знаходження кутів

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

Працювати відразу з усіма точками зображення досить складно, отже для мобільних платформ алгоритм пошуку потрібно спростувати.

FAST алгоритм вирішує два завдання – пошук особливих точок зображення і створення їх дескрипторів, інваріантних до масштабу і обертання. Це означає, що опис ключової точки буде однаковим, навіть якщо зразок змінить розмір і буде повернений. Крім того, сам пошук ключових точок повинен бути інваріантним, щоб повернений об'єкт сцени мав той же набір ключових точок, що і зразок [14].

В алгоритмі розглядається коло з 16 пікселів навколо точки кандидата P , як це зображено на рисунку 2.4, побудоване за алгоритмом Брезенхема. У цьому алгоритмі будується дуга кола для першого квадранта, а координати точок кола для інших квадрантів визначаються симетрично. На кожному кроці алгоритму розглядаються три пікселі, з яких вибирається найбільш придатний шляхом порівняння відстаней від центру до обраного пікселя з радіусом кола. Далі вибирається точка

зображення p , для якої буде вирішуватися, чи є вона особливою. Нехай I_p – яскравість точки, а t – значення порога. Розглянемо коло з 16 пікселів навколо обраної точки.

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

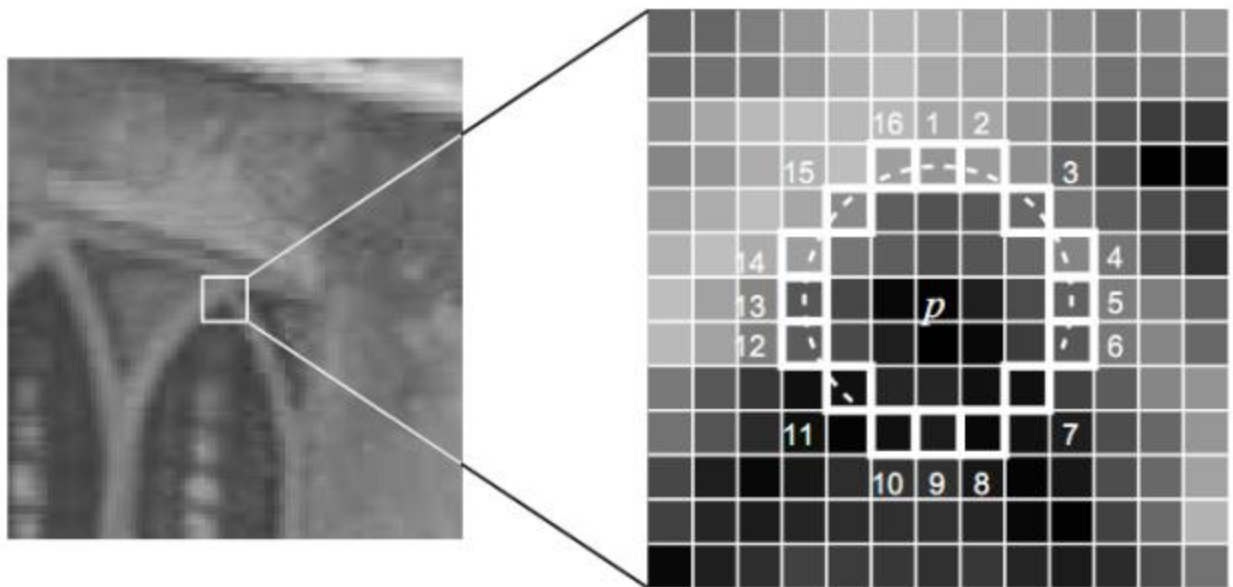


Рисунок 2.4 – Робочий окіл пікселя при використанні FAST-детектора

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

Однак даний алгоритм має ряд недоліків, наприклад, його ефективність залежить від порядку обробки зображення і розподілу пікселів, а поблизу деякого околу може виявитися кілька особливих точок. У [8] було запропоновано використання машинного навчання для виправлення цих недоліків, завдяки чому ми отримали модифікований алгоритм:

- вибирається набір зображень для навчання;
- алгоритм FAST застосовується для пошуку особливих точок на кожному із зображень;
- для кожної особливої точки зберігається 16 пікселів навколо неї у векторному вигляді. Після виконання даного кроку для кожного зображення всі отримані вектори об'єднуються в один вектор P ;
- кожен піксель з 16-ти може знаходитися в одному з трьох станів, які враховано в наступній формулі:

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

- в залежності від станів, наведених вище, P розділяється на три підмножини P_d, P_s, P_b ;
- в розгляд вводиться булева змінна K_p , яка дорівнює 1, якщо p – кут та 0 у зворотному випадку;
- використовується алгоритм ID3 (класифікатор дерев рішень) [9]. Він вибирає піксель x , який дає більше інформації про те, чи є піксель кутом чи ні, обчисленої за допомогою ентропії K_p ;
- ця процедура повторюється рекурсивно до тих пір, поки ентропія K_p не дорівнюватиме нулю;
- створене таким чином дерево рішень використовується для більш швидкого детектування в інших зображеннях.

Для того, щоб уникнути знаходження декількох особливих точок поблизу деякого околу, використовується підхід до пошуку локальних максимумів, що має назву non-maximal suppression. Згідно з ним обчислюється значення функції V для кожної зі знайдених особливих точок (V – сума різниць інтенсивностей p і кожного з 16-ти навколишніх пікселів). Далі розглядаються дві сусідні особливі точки, порівнюються їх значення V , а точка з меншим таким значенням виключається з розгляду.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Перед переходом до експериментального дослідження, перш за все, необхідно вибрати набір програмного API та модулів, які будуть задовольняти усім потребам для реалізації програмного продукту. В якості модуля API, що підтримує технології Augmented reality для забезпечення функції трекінгу, був обраний модуль PTC Vuforia. Оскільки Vuforia підтримує різні 2D і 3D типи мішеней, включаючи маркерні Image Target, це дозволить в подальшому використовувати вкладені функції для реалізації плавного треку маркерів, які будуть реалізовані у вигляді деяких реальних графічних зображень.

Наступним кроком став вибір графічного рушія, який дозволив би перенести додаток на мобільну платформу, оскільки відео потік буде транслюватися безпосередньо через відеокамеру мобільного пристрою, або персонального комп'ютера. Оптимальним ядром можна вважати інструмент для розробки додатків, що працює під операційною системою Windows та дозволяє компілювати проекти під Windows і різні мобільні платформи (зокрема, Android та iOS). Також необхідно вибрати мову програмування, яка б мала широку алгоритмічну базу, можливість модулювання, а також широку підтримку платформ. Цим вимогам задовольняє мова C# 7.0, що легко інтегрується в графічне ядро Unity.

На початковому етапі розробки додатку був реалізований модуль алгоритму FAST для знаходження основних точок на зображенні (рис. 3.1). Для реалізації використано крос-платформу EmguCV, що представляє собою .Net обгортку над OpenCV бібліотекою для обробки зображень і дозволяє визивати OpenCV функції з будь-якої мови яка відноситься до платформи .Net [15]. Однією з таких мов є C#, використана в розроблюваному додатку. Обгортка може компілюватися таким середовищами розробки, як Visual Studio, Xamarin Studio та ігровий рушій Unity.

```

using Emgu.CV;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.Util;
using Project.Scripts.FastFeatureDetection;

namespace Project.Scripts.FastFeatureDetection
{
    [1 usage]
    public class FastFeatureDetection
    {
        public static Image<Bgr, byte> Draw(Image<Gray, byte> modelImage, Image<Gray, byte> observedImage)
        {
            HomographyMatrix homography = null;
            FastDetector fastCPU = new FastDetector( threshold: 16, nonmaxSupression: true);
            BriefDescriptorExtractor descriptor = new BriefDescriptorExtractor();

            const int k = 2;
            const double uniquenessThreshold = 0.8;

            //извлечение особенностей из изображения объекта
            VectorOfKeyPoint modelKeyPoints = fastCPU.DetectKeyPointsRaw(modelImage, mask: null);
            Matrix<byte> modelDescriptors = descriptor.ComputeDescriptorsRaw(modelImage, mask: null, modelKeyPoints);

            //извлечение особенностей из наблюдаемого изображения
            VectorOfKeyPoint observedKeyPoints = fastCPU.DetectKeyPointsRaw(observedImage, mask: null);
            Matrix<byte> observedDescriptors =
                descriptor.ComputeDescriptorsRaw(observedImage, mask: null, observedKeyPoints);
            BruteForceMatcher<byte> matcher = new BruteForceMatcher<byte>(DistanceType.L2);
            matcher.Add(modelDescriptors);

            Matrix<int> indices = new Matrix<int>(observedDescriptors.Rows, cols: k);
            Matrix<float> dist = new Matrix<float>(observedDescriptors.Rows, cols: k);
            matcher.KnnMatch( queryDescriptor: observedDescriptors, indices, dist, k, mask: null);
            Matrix<byte> mask = new Matrix<byte>(dist.Rows, cols: 1);
            mask.SetValue( val: 255);
            Features2DToolbox.VoteForUniqueness(dist, uniquenessThreshold, mask);

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

```

Рисунок 3.1 – Реалізація методу «FAST»

У даній реалізації алгоритм ID3 використовується для оптимізації порядку, в якому випробовуються пікселі, в результаті чого отримується найбільш обчислювально ефективний детектор особливостей.

Для отримання проміжних результатів додано допоміжний алгоритм, який буде виводити початкові зображення і зображення, що пройшли препроцесінг з візуалізованими на них точками. Код допоміжного алгоритму представлений на рисунку 3.2, а проміжні результати на рисунку 3.3.

```

//Отрисовка замеченных ключевых точек
Image<Bgr, byte> result = Features2DToolbox.DrawMatches(modelImage, modelKeyPoints, observedImage,
    observedKeyPoints, indices,
    matchColor: new Bgr( blue: 255, green: 255, red: 255),
    singlePointColor: new Bgr( blue: 255, green: 255, red: 255), mask,
    flags: Features2DToolbox.KeypointDrawType.DEFAULT);

if (homography == null)
    return result;

//Отрисовка проекции
DrawProjection(modelImage, homography, result);

return result;
}

1 usage
private static void DrawProjection(Image<Gray, byte> modelImage, HomographyMatrix homography,
    Image<Bgr, byte> result)
{
    Rectangle rect = modelImage.ROI;
    PointF[] pts =
    {
        new PointF( rect.Left, rect.Bottom), new PointF( rect.Right, rect.Bottom),
        new PointF( rect.Right, rect.Top), new PointF( rect.Left, rect.Top)
    };
    homography.ProjectPoints(pts);

    result.DrawPolyline( pts: Array.ConvertAll(pts, Point.Round), isClosed: true, color: new Bgr(Color.Yellow), thickness: 5);
}
}

```

Рисунок 3.2 – Алгоритм виведення проміжних результатів

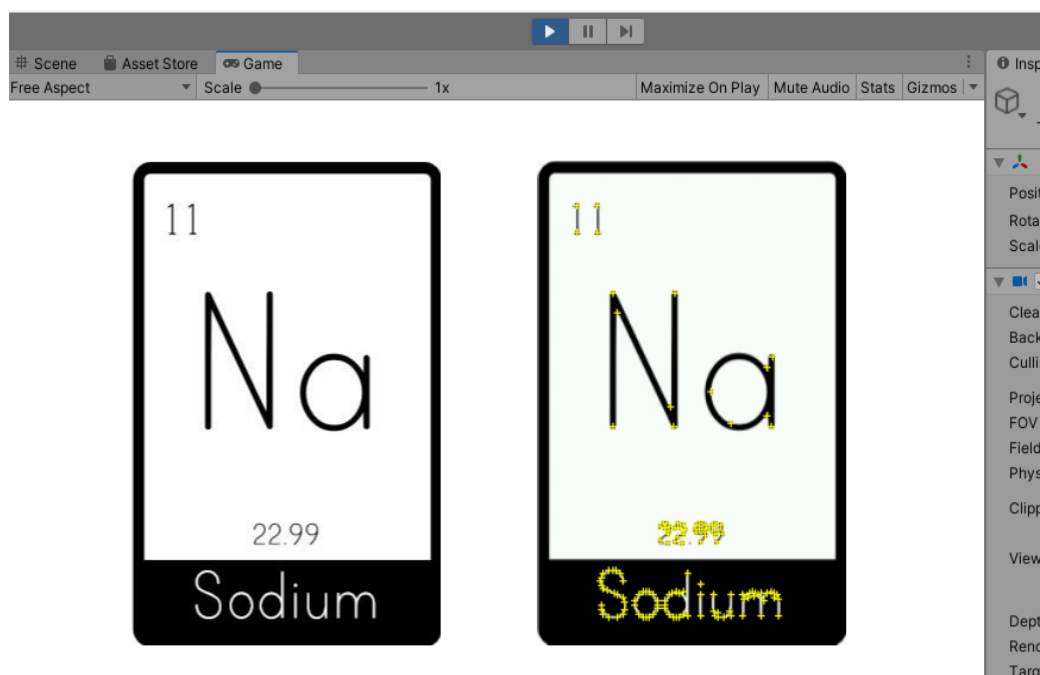


Рисунок 3.3 – Візуалізація особливих точок методу FAST.

Згідно з практичною орієнтацією додатка було обрано API для роботи з хімічними елементами: NCDK (.Net Chemistry Development Kit),

який представляє набір бібліотек для платформи .NET, що забезпечує такі можливості[16]:

- представлення молекулярно-валентної зв'язки у молекулі та реакції;
- читання та запис форматів файлів: SMILES, SDF, InChI, Mol2, CML та ін;
- алгоритми ефективної обробки молекули: пошук кільця, молекулізація, ароматність;
- читання вхідних файлів та баз знань для формування колекцій даних;
- розрахунки дескрипторів для алгоритмів QSAR.

Наступним кроком є формування бази знань хімічних ізотопів для подальшого формування молекул. Для цього будемо використовувати базу знань ізотопів та їх характеристик у форматі XML, яка нараховує більшу кількість відомих активних та не активних ізотопів (приблизно 2 тисячі), та має структуру, приклади якої наведені на рисунку 3.4, та рисунку 3.5.

```
<isotopeList id="H">
  <isotope id="H1" number="1" elementType="H">
    <scalar dictRef="bo:relativeAbundance">99.9885</scalar>
    <scalar dictRef="bo:exactMass" errorValue="0.0001E-6">1.007825032</scalar>
    <scalar dictRef="bo:spin">1/2+</scalar>
    <scalar dictRef="bo:magneticMoment">+2.792846</scalar>
    <scalar dictRef="bo:halfLife" units="siUnits:s">-1</scalar>
    <scalar dictRef="bo:atomicNumber">1</scalar>
  </isotope>
  <isotope id="H2" number="2" elementType="H">
    <scalar dictRef="bo:relativeAbundance">0.0115</scalar>
    <scalar dictRef="bo:exactMass" errorValue="0.0004E-6">2.014101778</scalar>
    <scalar dictRef="bo:spin">1+</scalar>
    <scalar dictRef="bo:magneticMoment">+0.857438</scalar>
    <scalar dictRef="bo:halfLife" units="siUnits:s">-1</scalar>
    <scalar dictRef="bo:atomicNumber">1</scalar>
  </isotope>
</isotopeList>
```

Рисунок 3.4 — Приклад ізотопів водню

```

<isotopeList id="Fe">
  <isotope id="Fe45" number="45" elementType="Fe">
    <scalar dictRef="bo:exactMass" errorValue="240E-6">45.01458</scalar>
    <scalar dictRef="bo:atomicNumber">26</scalar>
  </isotope>
  <isotope id="Fe46" number="46" elementType="Fe">
    <scalar dictRef="bo:exactMass" errorValue="380E-6">46.00081</scalar>
    <scalar dictRef="bo:halfLife" units="siUnits:s">0.02</scalar>
    <scalar dictRef="bo:atomicNumber">26</scalar>
  </isotope>
  <isotope id="Fe47" number="47" elementType="Fe">
    <scalar dictRef="bo:exactMass" errorValue="280E-6">46.99289</scalar>
    <scalar dictRef="bo:halfLife" units="siUnits:s">0.027</scalar>
    <scalar dictRef="bo:atomicNumber">26</scalar>
  </isotope>
</isotopeList>

```

Рисунок 3.5 — Приклад ізотопів заліза

Після отримання усієї потрібної інформації відносно хімічних елементів необхідно обробляти цю базу даних та формувати заходи для одержання інформації з бази за символом хімічного елементу (приклад коду наведено на рисунку 3.6). Для цього був створений клас, який реалізує патерни для повернення результатів у вигляді класу ізотопу. API для отримання класу ізотопу наведено на рисунку 3.7.

```

public class XMLIsotopeFactory
    : IsotopeFactory
{
    public static XMLIsotopeFactory Instance { get; } = new XMLIsotopeFactory();

    /// <summary>
    /// Private constructor for the IsotopeFactory object.
    /// </summary>
    /// <exception cref="IOException">A problem with reading the isotopes.xml file</exception>
    private XMLIsotopeFactory()
    {
        Trace.TraceInformation( message: "Creating new IsotopeFactory");

        var errorMessage (string) = $"There was a problem getting NCDK.Config.Data.isotopes.xml as a stream";
        var configFile = "NCDK.Config.Data.isotopes.xml";
        Debug.WriteLine( message: $"Getting stream for {configFile}");
        using (var reader = new IsotopeReader( input: ResourceLoader.GetAsStream(configFile)))
        {
            var isotopes (ReadOnlyList<Isotope> = reader.ReadIsotopes();
            foreach (var isotope in isotopes)
            {
                Add(isotope);
                Debug.WriteLine( message: $"Found #isotopes in file: {isotopes.Count}");
            }
        }
    }
}

```

Рисунок 3.6 – API для зчитування даних з бази даних хімічних ізотопів

```
2 usages
public static class ChemicalElementsInstanceBuilder
{
    1 usage
    private static XMLIsotopeFactory XMLIsotopeFactory =>
        XMLIsotopeFactory.Instance;

    public static IEnumerable<IIsootope> GetAllIsotopes() => |
        XMLIsotopeFactory.Instance.GetIsotopes();

    1 usage
    public static IAtomContainer GetNewAtomContainer() =>
        CDK.Builder.NewAtomContainer();

    public static IAtom GetNewAtomBySymbol(string symbol) =>
        CDK.IsotopeFactory.Configure(CDK.Builder.NewAtom(symbol), CDK.Builder.NewIsotope(symbol));

    1 usage
    public static IIsootope XMLGetNewIsotopeBySymbol(string symbol) =>
        XMLIsotopeFactory.GetMajorIsotope(symbol);
}
```

Рисунок 3.7 – API для отримання екземплярів хімічних елементів на основі даних з бази даних

Після підготовки бази даних хімічних елементів та написання і формування API для отримання екземплярів хімічних елементів на основі даних у XML, а також успішної інтеграції бази даних зображень хімічних елементів в проект Unity, необхідно, використовуючи RTC Vuforia плагін, співвіднести зображення з елементами хімічних елементів і віртуальними об'єктами, що відповідають хімічним елементам. У якості віртуальних об'єктів будуть розглядатися макети атомів та їх реальне представлення у вигляді якоїсь субстанції з координатами, що відповідають локальним координатам відповідного маркера.

4 РОЗРОБКА ДОДАТКУ

4.1 Засоби розробки

4.1.1 Ігровий рушій

Термін «ігровий рушій» з'явився в 1990-х роках. Перший 3D рушій був розроблений Джоном Кармак в 1996 році в компанії id Software і називався Quake engine. Даний рушій був використаний в більш ніж десяти проектах і дав серйозний поштовх розвитку цієї індустрії [17].

Ігрові рушії були, зокрема, створені для спрощення і прискорення розробки ігор. Існує кілька понять движків, але найчастіше це ігрові та графічні. Ігровий рушій – це той модуль гри, який реалізує ігрову логіку.

Ігровий рушій включає до себе рушій рендеринга (візуалізатор), фізичний рушій, звуковий рушій, систему скриптів, анімацію, штучний інтелект, мережевий код, управління пам'яттю і багато поточність. Так можна заощадити на процесі розробки за рахунок повторного використання одного і того ж рушія для створення різних ігор. Існують чимало різних ігрових рушіїв (від самих популярних брендів, ціни яких є занадто високими для багатьох програмістів, до рушіїв, які створюють для внутрішніх проектів). Вибір потрібного рушія є складним завданням. Технологічні платформи, які вже довели свою ефективність в реальних проектах, є очевидними фаворитами багатьох розробників. Тим не менш, ринок мінливий, а тому питання вибору рушія для розробки свого додатку своєї актуальності найближчим часом не втратить. Розглянемо деякі розповсюджені типи рушіїв.

Unity 3D. Це потужне середовище для розробки 3D ігор і додатків. Головний плюс Unity 3D – простота розробки додатків для мобільних платформ. На даному рушії розробляється чимала кількість ігор під мобільні платформи. Створені за допомогою Unity 3D ігри та програми

працюють в операційних системах Windows, OS X, Android, iOS, Linux, Blackberry, а також на ігрових приставках Wii, PlayStation 3 і Xbox 360. Програми, створені за допомогою Unity, підтримують DirectX і OpenGL.

Unreal Engine. Це один з найбільш популярних рушіїв для розробки AAA-ігор (зокрема, ігри Gears of War, Batman: Arkham Asylum, Mass Effect були створені саме на ньому). Даний рушій написаний на мові C++, розроблений і підтримується компанією Epic Games. Створені ігри та програми на Unreal Engine працюють на різних операційних системах і платформах.

CryEngine 3. Цей популярний рушій, написаний на мові C++, розроблено німецькою компанією Crytek. Даний рушій орієнтований на ПК IBM та на розробку масових багато користувальницьких онлайн ігор. В його комплекті є все, що необхідно для реалізації AAA-класу ігор, у тому числі редактор рівнів, який здатний створювати карти в реальному часі.

3D Rad. Це безкоштовний рушій для створення різних 3D-ігор, інтерактивних додатків і фізичних симуляцій. Створення додатків в 3D Rad ґрунтується на поєднанні в різних комбінаціях об'єктів і налаштування взаємодії між ними. У 3D Rad є функція імпорту моделей в створюваний проект, велика кількість прикладів і зразків тривимірних об'єктів, а також можливість додавання звукових ефектів в форматі WAV або OGG.

За результатами аналізу відомих ігрових рушіїв було прийнято рішення для розробки додатку в магістерській роботі використовувати рушій Unity. Таке рішення базується на таких критеріях:

- вартість (ліцензія безкоштовна);
- можливість використання мови програмування C#;
- мультиплатформеність (Unity дозволяє якісно створювати ігри і додатки для великої кількості відомих платформ).

Unity – це мультиплатформений інструмент для розробки двох і тривимірних додатків та ігор, що працює під операційними системами Windows та OS X. Ігри, створені на цьому інструменті, можна перенести

на: Windows, OS X, Android, Apple iOS, Linux, а також на ігрові приставки Wii, PlayStation, Xbox та інші. Так само можна створювати ігри, що працюють в браузері, для цього треба встановити спеціальний модуль Unity Web Player. Також ігри створені, за допомогою Unity3D підтримують обидві специфікації 3D графіки DirectX і OpenGL.

Unity має дуже зручний та гнучкий інтерфейс, який ніяк не погіршує функціонал [10]. Це означає, що зовнішній вигляд редактора може відрізнятися від одного проекту до іншого, кожен розробник може налаштувати його так, як йому буде зручно (зазвичай це залежить від того, який тип роботи виконує розробник). Цей рушій дає практичний доступ до найбільш поширених вікон. Головне вікно редактора складається з декількох вкладок «Види (Views)». У Unity є кілька типів видів, що призначені для конкретних цілей (рисунок 4.1).

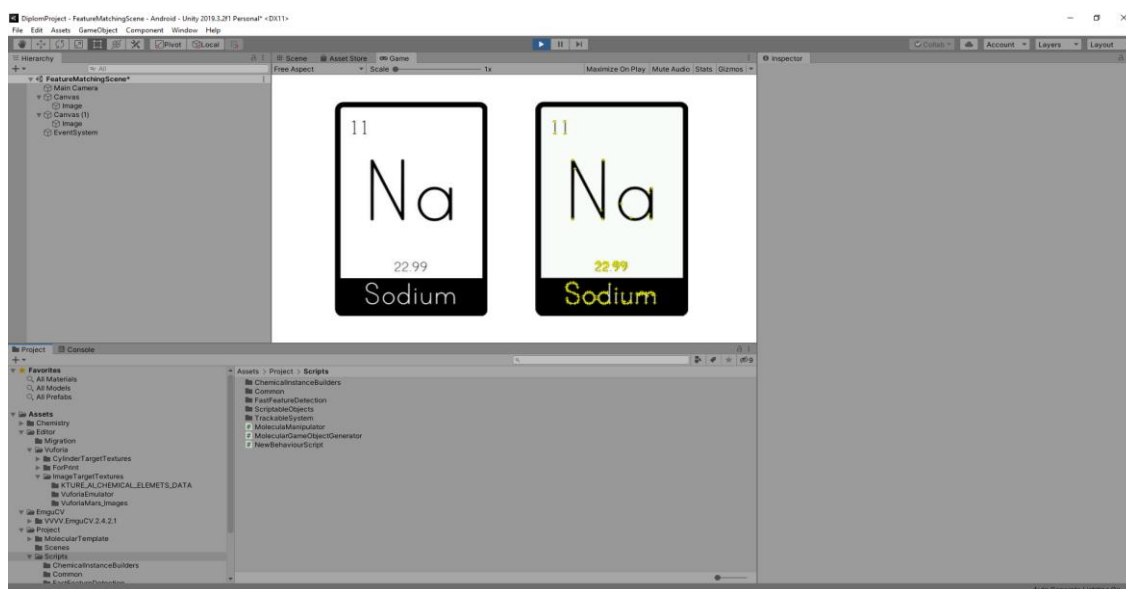


Рисунок 4.1 – Головний інтерфейс ігрового рушія Unity

Вікно проекту (Project Window) відображає бібліотеку активів (Assets), доступних для використання у проекті. Коли користувач імпортує активи у свій проект, вони з'являються в цьому вікні (рисунок 4.2).

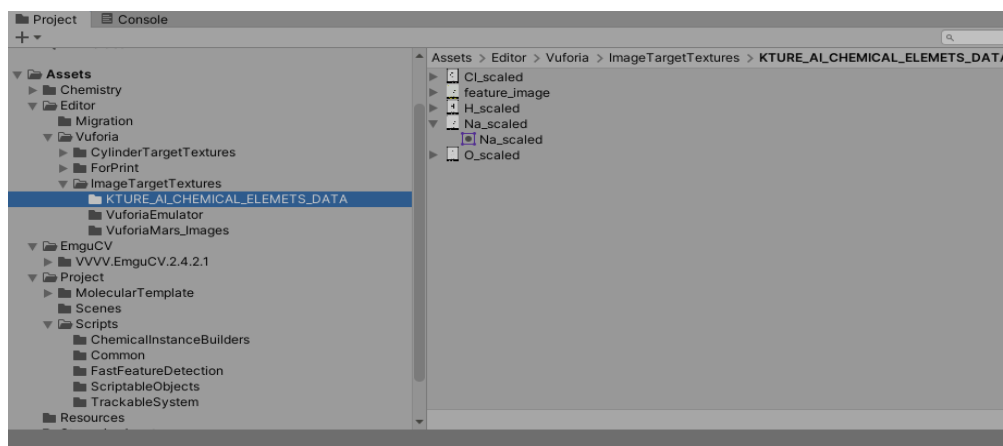


Рисунок 4.2 – Project Window

Вікно сцени (Scene View) – це інтерактивний вид на світ, який використовує користувач (рисунок 4.3). Зазвичай цей вид використовується, щоб вибрати і розташувати декорації, персонажів, камери, освітлення та всі інші об'єкти гри. Крім того, він дає можливість вибирати, керувати та змінювати об'єкти в режимі перегляду сцени.

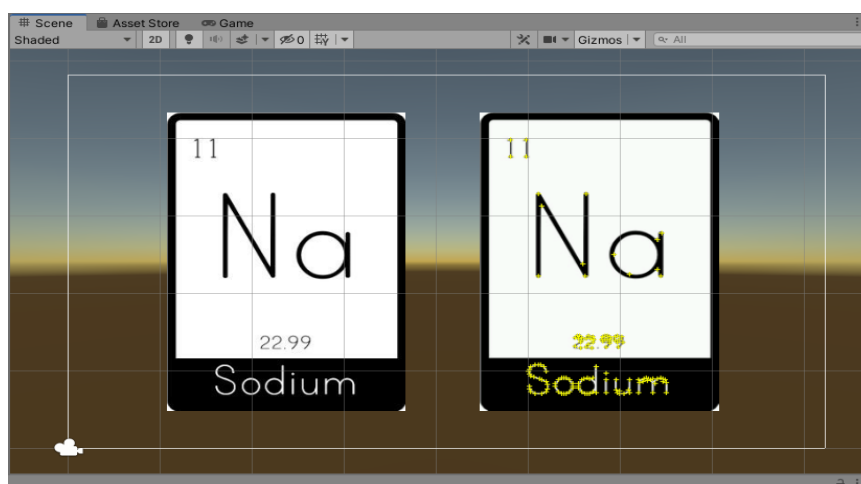


Рисунок 4.3 – Scene View

Вікно гри (Game View) – перегляд ігрового простору з камери у грі (рисунок 4.4).

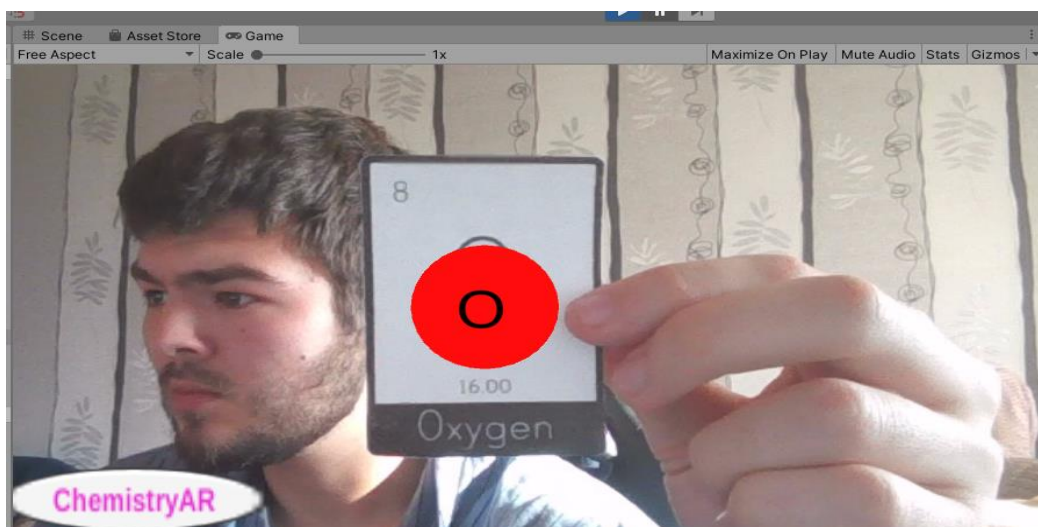


Рисунок 4.4 – Game View

Вікно ієрархії (Hierarchy Window) – це ієрархічне текстове зображення кожного об'єкта у поточній сцені. Деякі з цих зображень є прямими екземплярами Assets-файлів, а інші – екземплярами Prefab. Оскільки об'єкти додаються та видаляються в сцені, вони також з'являються та зникають з ієрархії. За замовчуванням об'єкти відображаються у вікні ієрархії в порядку їх створення. Користувач може змінити їх позицію, перетягуючи об'єкт вгору або вниз, або перетворюючи їх на «дочірній» або «батьківський» об'єкти (рисунок 4.5).

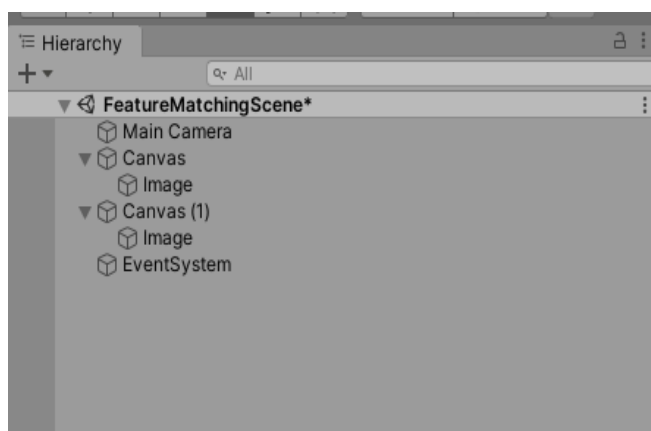


Рисунок 4.5 – Hierarchy Window

Вікно інспектора (Inspector Window) – дозволяє переглядати та редагувати усі властивості поточного виділеного об’єкта. Оскільки різні типи об’єктів мають різні набори властивостей, макет, вміст і компоненти, вікна інспектора будуть різними, рисунок 4.6.

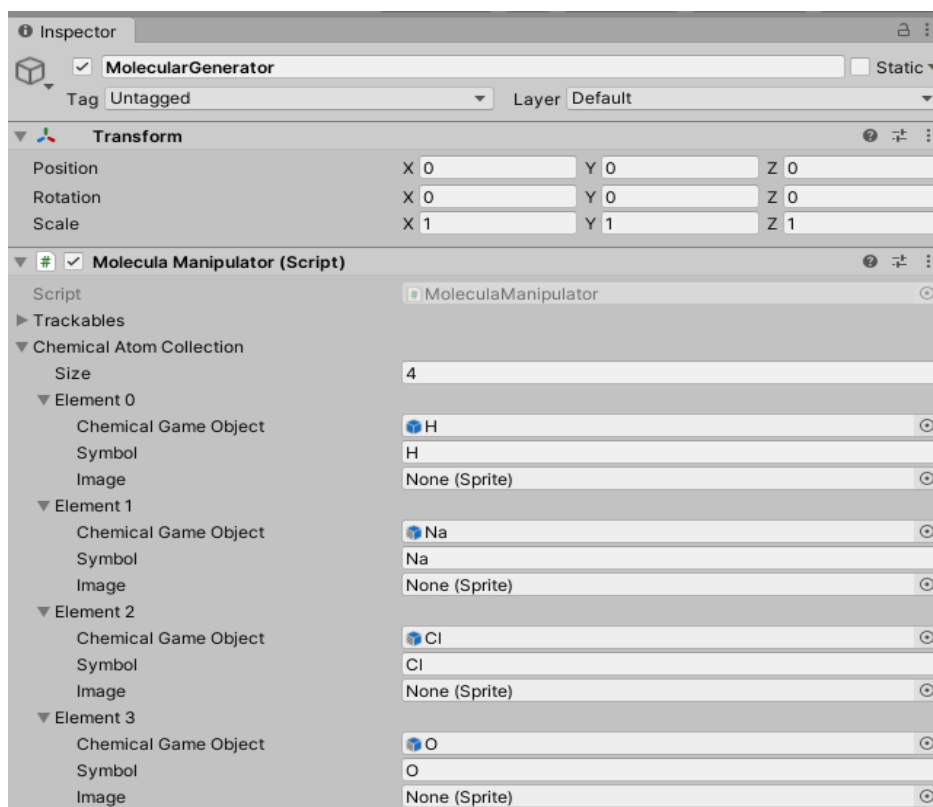


Рисунок 4.6 – Inspector Window

Панель інструментів (Toolbar) надає доступ до найважливіших робочих функцій. Зліва вона містить основні інструменти для роботи з переглядом сцени та об’єктами всередині сцени. У центрі розташовані елементи керування відтворенням, паузою та покадровими кроками. Кнопки праворуч надають доступ до Unity Cloud та Unity Account, далі розташовані меню видимості шарів, меню макета редактора (надає можливість активувати альтернативні види редакторів та зберігати власні). Панель інструментів є єдиною частиною інтерфейсу Unity, яку не можна змінити (рисунок 4.7).



Рисунок 4.7 – Toolbar

Консольне вікно (Console Window) необхідне для оповіщення розробників як в режимі реального часу так і в попередньому режимі збірки проекту (компіляція усіх залежностей програми). Збірка відбувається автоматично при виявленні змін в вихідному коді, що дозволяє виявити помилки на ранній стадії. У Unity існує декілька рівнів логуювання (рисунок 4.8):

- інформаційний рівень – показує хід роботи програми, дозволяє відобразити будь-яку текстову інформацію про поточний стан програми або будь-які інші інформаційні повідомлення;
- рівень попереджень – попереджає про різні упущення в ході розробки, але програма може нормально працювати при такому рівні;
- рівень помилок – вказує на явні помилки, які мали місце під час компіляції або у ході роботи програми.

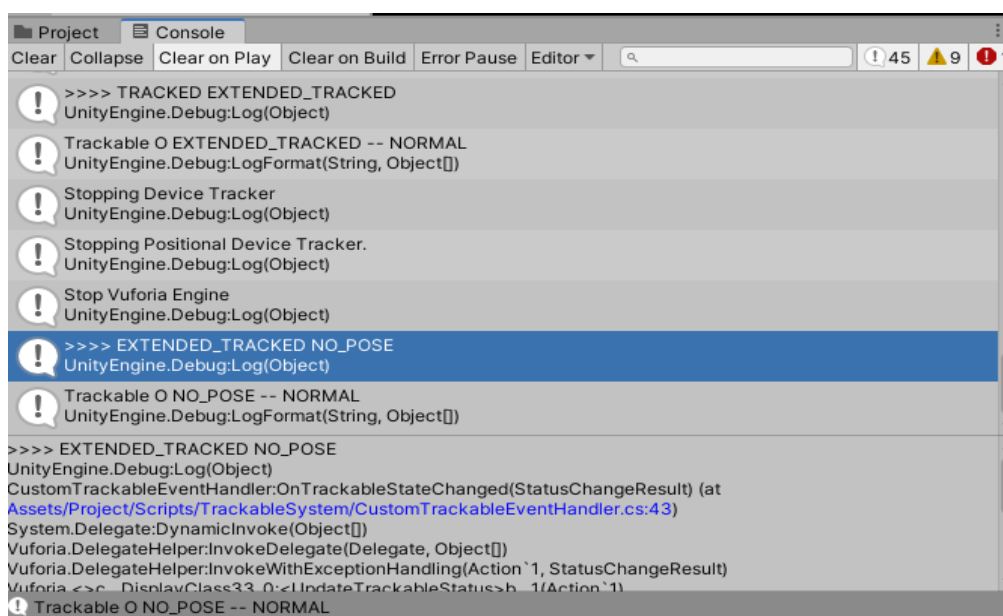


Рисунок 4.8 – Console Window

4.1.2 Бібліотека доповненої реальності

Щоб обрати інструмент для використання у розробці продукту, необхідно спочатку зробити порівняльний аналіз найвідоміших засобів для розробки додатків, які використовують технології доповненої реальності, та знайти інструментарій який буде задовольняти такому набору вимог:

- можливість щодо інтеграції до ігрового рушія Unity;
- можливість трекінгу багатьох зображень одночасно;

ARCore – це набір для розробки програмного забезпечення, який дозволяє будувати додатки доповненої реальності [18]. ARCore використовує три ключові технології для інтеграції віртуального контенту до реального світу, що спостерігається через камеру телефону:

- “Six degrees of freedom“(шість рівнів свободи) – система 3х вимірного простору з кутом обертання для кожного виміру, що дозволяє телефону зрозуміти та відслідковувати його положення стосовно світу;

- розуміння навколишнього середовища, що дозволяє телефону визначити розміри та розташування плоских горизонтальних платформ, таких як земля або стіл;

- оцінка рівня світла дозволяє телефону оцінити поточні умови освітлення навколишнього середовища.

ARCore – це один з багатьох засобів, який був інтегрований у безліч пристроїв різноманітних виробників телефонів. Недоліком платформи є те, що набір для розробки програмного забезпечення працює лише на операційній системі Android 7.0 та більше.

Крім ARCore, набув поширення набір програмного забезпечення від компанії Apple під назвою ARKit, що має такий же функціонал і обмеження (цей набір підтримують пристрої з версією iOS 9.0 та більше) [19].

Wikitude SDK – це кросплатформовий набір програмного забезпечення, який підтримує Android, iOS та Windows та може

використовуватися для декількох смарт-окулярів [20]. Wikitude був першим загальнодоступним додатком, який використовував локальний підхід до розширеної реальності. Основні його особливості та функції:

- доповнена реальність основана на місці розташування (є набір API для геолокації; обчислюється, використовуючи положення користувача (за допомогою GPS або Wifi), напрямом, в якому користувач переміщується (за допомогою компаса) та акселерометр;

- розпізнавання зображень (дозволяє трекарам зображень запускати технологію розширеної реальності в додатку. Програмне забезпечення визначає відповідні точки функції цільового зображення (маркер). Це дозволяє накладати і наклеювати доповнення у певне положення зверху або навколо зображення);

- SLAM (миттєве відстеження, розпізнавання об'єктів та сцен). Унікальна технологія, яка використовується виключно у цьому інструментарії. Дозволяє розробникам легко відображати середовища та відображати зміст доповненої реальності без потреби в цільових зображеннях (маркерах). Розпізнавання об'єктів – це останнє нововведення на основі SLAM, із запуском SDK 7. Ідея розпізнавання об'єктів і відстеження дуже схожа на відстеження зображень, але замість розпізнавання зображень та плоских поверхонь, об'єкт відстеження може працювати з тривимірними структурами та предметами (інструменти, іграшки, техніка тощо).

Vuforia – це набір для розробки програмного забезпечення розширеної реальності (SDK) для мобільних пристроїв, що дозволяє створювати додатки доповненої реальності [18]. Він використовує технологію комп'ютерного зору для розпізнавання та відстеження цільових зображень та 3D-об'єктів у режимі реального часу. Ця можливість реєстрації зображень дозволяє розробникам розміщувати та орієнтувати віртуальні об'єкти, такі як 3D-моделі та інші медіа, стосовно об'єктів реального світу, коли вони переглядаються через камеру мобільного

пристрою. Потім віртуальний об'єкт відстежує положення та орієнтацію зображення в режимі реального часу, щоб погляд глядача на об'єкт відповідав перспективі на ціль. Набір Vuforia підтримує різноманітні 2D та 3D-цілі, включаючи цілі зображення без маркерів, цільову модель 3D та форму адресованого Fiducial Marker, відомого як VuMark (комбінація зображення і QR-коду). Додаткові функції SDK включають шість ступенів локалізації пристрою свободи (Six degrees of freedom) в просторі, локалізоване виявлення оклюзії (віртуальні об'єкти які дозволяють ховати за собою віртуальні об'єкти, імітуючи стіну або будь який статичний та простий об'єкт із реального виміру) за допомогою "Віртуальних кнопок", знаходження цільових зображень у відео потоці, можливість створення та налаштування набору цілей програмно під час виконання програми.

Vuforia надає інтерфейси програмування (API) на мовах C++, Java, Objective-C ++ та .NET через розширення до ігрового механізму Unity. Таким чином, SDK підтримує як вбудовану розробку для iOS, Android, так і UWP (Universal Windows Platform), в той час як він також дозволяє розробляти додатки AR в Unity, яка є міжплатформовою та надаю багато більше можливостей ніж усі інші претенденти.

Огляд альтернатив та визначення основних критеріїв аналізу зображень [21] та урахування найбільш важливих критеріїв показали, що згідно з вимогами оптимальним середовищем для розробки нашої системи є Vuforia. Відзначимо, що ARCore та ARKit можуть бути інтегровані до Unity, але лише для своїх платформ Android та iOS відповідно. Набір Wikitude SDK не підходить через те, що він не підтримує можливість трекінгу декількох об'єктів одночасно, а після знаходження іншого об'єкту він буде буде "забувати" про минулий. Переглянемо детальніше набір функціоналу, який постачає компанія PTC зі своїм продуктом Vuforia:

- Model Targets (дозволяє розпізнавати об'єкти за формою, використовуючи попередньо існуючі тривимірні моделі);
- Area Targets (дозволяє розширяти реальні середовища, які ви

скануєте за допомогою комерційно доступного 3D-сканера);

- Image Targets (дозволяє доповнювати будь які плоскі зображення, наприклад: друківані носії та упаковки продуктів);

- Object Targets (створюється завдяки скануванню об'єкта, що має багато деталей та статичну форму);

- Multi-Targets (створюються за допомогою декількох цільових зображень і можуть бути представлені у вигляді геометричної фігури або в будь-якому довільному розташуванні плоских поверхонь);

- Cylinder Targets (дозволяє розпізнавати зображення, загорнуті на предметі циліндричної форми (наприклад, пляшки для напоїв, чашки для кави, банки з напоєм));

- VuMarks (дозволяє розпізнавати спеціалізовані маркери, які можуть кодувати різноманітні формати даних та підтримують унікальну ідентифікацію та відстеження для програм AR);

- Ground Plane (дозволяє розміщувати вміст на горизонтальних або вертикальних поверхнях навколишнього середовища).

4.2 Розробка додатку

Доцільно провести декомпозицію розробки додатку на інтуїтивно зрозумілі і доступні для контролю етапи, а саме:

- завантаження та інсталяція ігрового рушія;
- інсталяція плагіна NCDK для роботи з хімічними елементами;
- написання парсера для набору даних, що містить хімічні елементи у форматі xml та серіалізація даних в екземпляр класу;

- інсталяція плагіна EmguCV, що є обгорткою над OpenCV для платформи .Net;

- написання класу для формування ключових точок (Features);

- інсталяція плагіну Vuforia до Unity проекту;

- підготовка набору зображень, формування локальної бази даних

зображень та формування дескрипторів для відповідних зображень моделей атомів;

- написання класів для реагування на знаходження зображення у відео потоці, генерація моделі атома та молекул (якщо алгоритм QSAR буде надавати можливість формувати молекули), а також формувати модель молекули та її реальне уявлення.

Розглянемо особливості проектної реалізації визначених кроків розробки додатку.

4.2.1 Інсталяція Unity

Для інсталяції Unity необхідно перейти на офіційний сайт (<http://unity3d.com/unity>) та обрати тип ліцензійної згоди. Unity має два рівні такої згоди, які поділяються декілька підрівнів:

- фізична персона: Student (для студентів, які зараховані в акредитовані освітні установи); Personal (для розробників, які розробляють додатки та у яких оборот або обсяг залучених інвестицій не перевищує сто тисяч доларів за останні дванадцять місяців);

- організація: Plus (для компаній, у яких оборот або обсяг залучених інвестицій не перевищує двісті тисяч доларів за останні дванадцять місяців); Pro (для компаній, оборот та обсяг яких перевищує двісті тисяч доларів за останні дванадцять місяців); Enterprise (спеціальний договір з компанією, який постачає відкритий код ігрового рушія для реалізації необхідного функціоналу).

В нашому випадку було обрано варіант персональної ліцензії для фізичних осіб, тому що оформлення студентської ліцензії є занадто складним. Після інсталяції Unity Hub з контейнеру, що дозволяє інсталювати декілька версій ігрового рушія, обираємо версію ігрового рушія Unity 19.3.2f1, яка на даний момент є самою актуальною із стабільних версій. Створюємо 3D проект і завершуємо перший крок.

4.2.2 Інсталяція .Net Chemistry Development Kit, імплементація

NCDK (.Net Chemistry Development Kit) – це набір бібліотек для платформи .Net з відкритим кодом, що постачає функціонал для роботи з хімією [16]. Існує два варіанта для встановлення цього набору бібліотек:

- клонування відкритого репозиторія до свого проекту (<https://github.com/kazuyaujihara/NCDK>);

- використання NuGet – система управління пакетами для платформ розробки Microsoft, в першу чергу бібліотек .Net Framework.

Більшість інтегрованих середовищ розробки (IDE – комплекс програмних засобів) для платформи .Net має систему для встановлення Nuget Packages під назвою Package Manager Console. Це дозволяє використовувати NuGet PowerShell для пошуку, встановлення та оновлення пакетів, внаслідок чого отримуємо дуже звучний інструментарій для встановлення бібліотек, що були запаковані у необхідний формат та належать усім необхідним стандартам по формуванню пакетів. Приклади інсталяції набору бібліотек NCDK та результат розміщення у проекті наведені на рисунку 4.9.

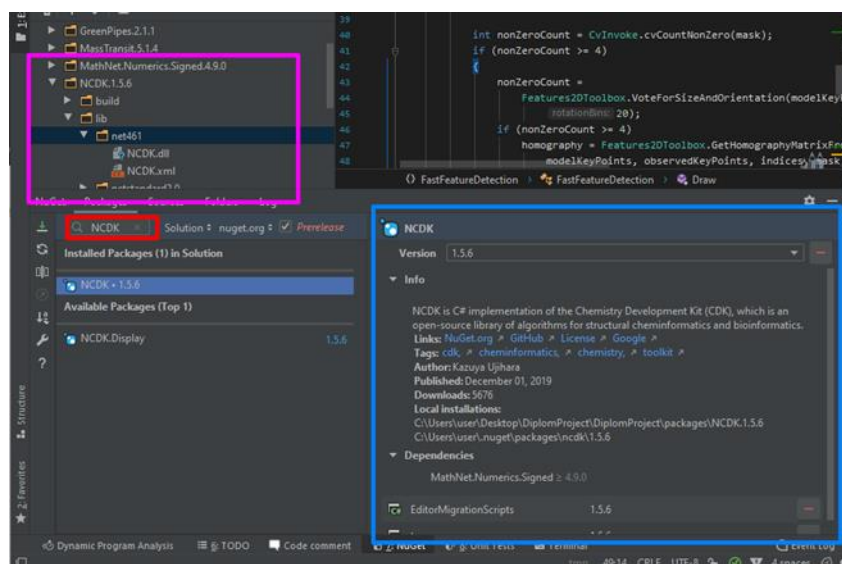


Рисунок 4.9 – Приклад використання Nuget Package Manager у IDE Rider

Далі необхідно написати клас, який дозволяє обробляти набір даних хімічних елементів у форматі xml. Для класу IsotopeReader застосуємо метод, що дозволяє зчитувати та повертати список знайдених та оброблених ізотопів (рисунок 4.10).

```

public IReadOnlyList<IIsootope> ReadIsotopes()
{
    XReader xreader = new XReader();
    IsotopeHandler isotopeHandler = new IsotopeHandler();
    xreader.Handler = (XContentHandler) isotopeHandler;
    try
    {
        XDocument doc = XDocument.Load(XmlReader.Create(this.input, new XmlReaderSettings()
        {
            ValidationType = ValidationType.None,
            ValidationFlags = XmlSchemaValidationFlags.None
        }));
        xreader.Read(doc);
        return isotopeHandler.Isotopes;
    }
    catch (Exception ex)
    {
        Trace.TraceError("message: {0}.GetType().Name + ": " + ex.Message);
    }
    return (IReadOnlyList<IIsootope>) Array.Empty<IIsootope>();
}

```

Рисунок 4.10 – Метод для обробки даних з xml файла

Створимо клас, який буде реалізовувати патерни Singleton та Factory (рисунок 4.11) для отримання однієї точки доступу до списку ізотопів та для повертання екземпляру класу хімічного ізотопа (рисунок 4.12).

```

public class XMLIsotopeFactory : IsotopeFactory
{
    #IL code
    public static XMLIsotopeFactory Instance { get; } = new XMLIsotopeFactory();
    #IL code
    private XMLIsotopeFactory()
    {
        using (IsotopeReader isotopeReader = new IsotopeReader(
            input: ResourceLoader.GetAsStream(name: "NCDK.Config.Data.isotopes.xml")))
        {
            foreach (IIsootope readIsotope in (IEnumerable<IIsootope>) isotopeReader.ReadIsotopes())
            {
                this.Add(readIsotope);
            }
        }
    }
}

```

Рисунок 4.11 – Контейнер для зчитаних даних з Xml

```

public virtual IEnumerable<IIsootope> GetIsotopes()
{
    List<IIsootope>[] isotopeListArray = this.isotopes;
    for (int index = 0; index < isotopeListArray.Length; ++index)
    {
        List<IIsootope> isotopeList = isotopeListArray[index];
        if (isotopeList != null)
        {
            foreach (IIsootope isotope in isotopeList)
            {
                yield return IsotopeFactory.Clone(isotope);
            }
        }
    }
    isotopeListArray = (List<IIsootope>[]) null;
}

```

Рисунок 4.12 – Метод для повернення списку ізотопів

Далі необхідно створити клас (рисунок 4.13), який буде обгорткою над раніше написаним кодом та має концентрувати весь функціонал в одному місці. Це дає можливість отримати данні з xml файлу, перетворити дані у список екземплярів ізотопів за методом, який буде повертати необхідний ізотоп, виходячи з його хімічного символу у періодичній таблиці Менделєєва.

```

public static class ChemicalElementsInstanceBuilder
{
    private static XMLIsotopeFactory XMLIsotopeFactory =>
        XMLIsotopeFactory.Instance;

    public static IEnumerable<IIsootope> GetAllIsotopes() =>
        XMLIsotopeFactory.Instance.GetIsotopes();

    public static IAtomContainer GetNewAtomContainer() =>
        CDK.Builder.NewAtomContainer();

    public static IAtom GetNewAtomBySymbol(string symbol) =>
        CDK.IsotopeFactory.Configure(CDK.Builder.NewAtom(symbol), CDK.Builder.NewIsotope(symbol));

    public static IIsootope XMLGetNewIsotopeBySymbol(string symbol) =>
        XMLIsotopeFactory.GetMajorIsotope(symbol);
}

```

Рисунок 4.13 – Набір інструментарію для роботи з хімічними елементами

Останнім кроком цього етапу є написання функціоналу для формування молекулярних формул, виходячи з вихідного набору та їх валентності (здатності елемента до хімічного об'єднання). На рисунку 4.14 наведено приклад методу який є огорткою, що дозволяє отримувати

примірник ізотопу та перевіряти, чи є у генераторі формули такий ізотоп (якщо є, то він збільшує кількість цього елемента на одиницю, інакше додає новий ізотоп).

```
private IMolecularFormulaSet GetAllPossibleFormulasWithNewIsotope(IIsootope isotope)
{
    if (ChemicalInstanceProvider.MolecularFormulaRange.Contains(isotope))
    {
        ChemicalInstanceProvider.MolecularFormulaRange.AddIsotope(isotope,
            ChemicalInstanceProvider.MolecularFormulaRange.GetIsotopeCountMin(isotope) + 1,
            ChemicalInstanceProvider.MolecularFormulaRange.GetIsotopeCountMax(isotope) + 1);
    }
    else
    {
        ChemicalInstanceProvider.MolecularFormulaRange.AddIsotope(isotope, 1, 1);
    }

    return MolecularFormulaInstanceBuilder.GetNewMolecularFormulaGenerator(ChemicalInstanceProvider.MolecularFormulaRange)
        .GetAllFormulas();
}
```

Рисунок 4.14 – Метод-огортка для формування хімічних формул

4.2.3 Інсталяція EmguCV, імплементація

Наступним кроком є встановлення плагіну EmguCV, що є обгорткою над OpenCV для платформи .Net[15]. Інсталяція плагіну відбувається відповідним шляхом, що й плагін для хімії (NCDK), з використанням NuGet Package Manager.

Після цього було розроблено алгоритм, який має на вході 2 параметри. Перший з них – це двохзв’язний список, де ключ – символ хімічного елемента, а значення – це набір дескрипторів (features) для зображення даного хімічного елемента, які були згенеровані раніше. Другий параметр – це поточний фрейм з безперервного відео потоку (рисунок 4.15).

Відзначимо, що необхідно провести реєстрацію на офіційному сайті розробників продукту компанії РТС – Vuforia (<https://developer.vuforia.com>), після цього кроку розробник має можливість працювати з більшою частиною функціоналу, який постачає цей продукт. Дали ми маємо можливість починати інтегрувати плагіни для розробки

додатків для Android, iOS, та Windows.

Завантаживши пакет з офіційного сайту, його необхідно імпортувати до ігрового рушія або використати систему “Asset Store” ігрового рушія Unity.

```

namespace Project.Scripts.FastFeatureDetection
{
    public class FastFeatureDetection
    {
        public static Dictionary<string, int> Detect(Dictionary<string, VectorOfKeyPoint> modelImage, Image<Gray, byte> observedImage)
        {
            HomographyMatrix homography = null;
            FastDetector fastCPU = new FastDetector(16, true);

            BriefDescriptorExtractor descriptor = new BriefDescriptorExtractor();

            const int k = 2;
            const double uniquenessThreshold = 0.8;

            //перевірка на валідність дескрипторів привдаєих на вхід
            VectorOfKeyPoint modelKeyPoints = fastCPU.DetectKeyPointsRaw(modelImage, null);
            Matrix<byte> modelDescriptors = descriptor.ComputeDescriptorsRaw(modelImage, null, modelKeyPoints);
            //Формирування дескрипторів текущего зображення из видеопотоков
            VectorOfKeyPoint observedKeyPoints = fastCPU.DetectKeyPointsRaw(observedImage, null);
            Matrix<byte> observedDescriptors = descriptor.ComputeDescriptorsRaw(observedImage, null, observedKeyPoints);
            //Сравнение дескрипторов
            BruteForceMatcher<byte> matcher = new BruteForceMatcher<byte>(DistanceType.L2);
            matcher.Add(modelDescriptors);

            Matrix<int> indices = new Matrix<int>(observedDescriptors.Rows, k);
            Matrix<float> dist = new Matrix<float>(observedDescriptors.Rows, k);

            // Возвращение результата
            return matcher.KnnMatch(observedDescriptors, indices, dist, k, null);
        }
    }
}

```

Рисунок 4.15 – Метод повертання результату порівняння відомих дескрипторів з зображенням з відео потоку

Даний плагін буде використовуватися для позиціювання віртуального об’єкта за координатами, які відповідають локальним координатам відповідного маркера і для відповідного розміру.

4.2.4 Підготовка зображень та завершення додатка

На цьому кроці були розроблені зображення карток атомів хімічних елементів, приклад яких наведено на рисунку 4.16.

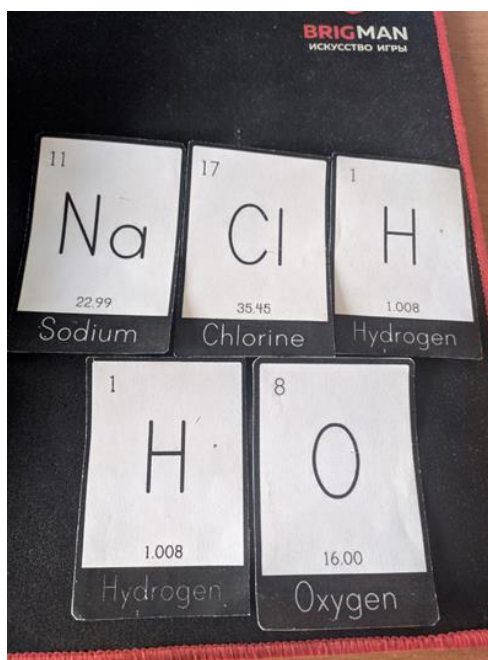


Рисунок 4.16 – Зображення карток атомів хімічних елементів, що будуть застосовані як маркери

Далі був написаний клас, що містить список моделей хімічних елементів, їх назву, набір ключових точок та дескрипторів (рисунок 4.17).

```

using System;
using System.Collections.Generic;
using Emgu.CV;
using Emgu.CV.Util;
using UnityEngine;

[CreateAssetMenu(fileName = "Data", menuName = "ScriptableObjects/ChemicalObjectContainer", order = 1)]
public class ChemicalObjectContainer : ScriptableObject
{
    [SerializeField] public List<ChemicalObject> ChemicalAtomCollection = new List<ChemicalObject>();
    [SerializeField] public List<ChemicalObject> ChemicalMolecularCollection = new List<ChemicalObject>();
}

[Serializable]
public class ChemicalObject
{
    public GameObject ChemicalGameObject;
    public string Symbol;
    public VectorOfKeyPoint modelKeyPoints;
    private Matrix<byte> modelDescriptors;
}
  
```

Рисунок 4.17 – Scriptable Object, локальна база даних хімічних елементів і їх віртуальних моделей

Наступним кроком є заповнення локальної бази даних дескрипторами відповідних зображень хімічних елементів (відповідні результати наведено у розділі експериментальних досліджень).

Далі з'явилася необхідність реалізувати клас `CustomTrackableEventHandler`, який для кожного кадру викликає алгоритм “Fast”, що буде знаходити зображення, викликати відповідні івенти та встановлювати стан знайденого об'єкта у необхідний. Коли метод пошуку зображень знаходить знайшов одне з цільових зображень за його дескриптором, він має створювати примірник класу `TrackableBehaviour` та починати відстежувати зміни його стану. Разом з цим буде викликано метод `HandleTrackableStatusChanged` (приклад наведено на рисунку 4.18). При реєстрації івенту знаходження цільового зображення викликається метод `OnTrackingFound` (рисунок 4.19), який передає символ знайденого елементу до класу, що формує формули молекул та генерує моделі хімічних молекул (рисунок 4.20).

```
protected virtual void HandleTrackableStatusChanged()
{
    if (!ShouldBeRendered(PreviousStatus) &&
        ShouldBeRendered(NewStatus))
        OnTrackingFound();
    else if (ShouldBeRendered(PreviousStatus) &&
            !ShouldBeRendered(NewStatus))
        OnTrackingLost();
    else
        if (!m_CallbackReceivedOnce && !ShouldBeRendered(NewStatus))
            OnTrackingLost();

    m_CallbackReceivedOnce = true;
}

protected bool ShouldBeRendered(TrackableBehaviour.Status status)
```

Рисунок 4.18 — Метод реагування на знаходження цільового зображення за його поточним станом

```

protected void OnTrackingFound()
{
    if (mTrackableBehaviour)
    {
        var rendererComponents = mTrackableBehaviour.GetComponentsInChildren<Renderer>(true);
        foreach (var component in rendererComponents)
            component.enabled = true;
    }

    MoleculaManipulator.AddChemicalElement(mTrackableBehaviour);

    OnTargetFound?.Invoke();
}

protected void OnTrackingLost()
{
    if (mTrackableBehaviour)
    {
        var rendererComponents = mTrackableBehaviour.GetComponentsInChildren<Renderer>(true);
        foreach (var component in rendererComponents)
            component.enabled = false;
    }

    OnTargetLost?.Invoke();
}

```

Рисунок 4.19 — Метод реагування на знаходження та втрату зображення у відео потоці

```

private void CreateChemicalInstanceProvider() =>
    ChemicalInstanceProvider = new ChemicalInstanceProvider(ChemicalElementsInstanceBuilder.GetNewAtomContainer(),
        MolecularFormulaInstanceBuilder.GetNewMolecularFormulaRange());

public void AddChemicalElement(TrackableBehaviour trackableBehaviour)
{
    if (!Trackables.Contains(trackableBehaviour))
        Trackables.Add(trackableBehaviour);

    var allDisabledTrackables = Trackables.Where(track =>
        track.CurrentStatus == TrackableBehaviour.Status.NO_POSE
        || track.CurrentStatus == TrackableBehaviour.Status.EXTENDED_TRACKED).ToList();

    var allActiveTrackables = Trackables.Where(track =>
        track.CurrentStatus == TrackableBehaviour.Status.DETECTED
        || track.CurrentStatus == TrackableBehaviour.Status.TRACKED
        || track.CurrentStatus == TrackableBehaviour.Status.EXTENDED_TRACKED).ToList();

    IIsotope newIsotope =
        ChemicalElementsInstanceBuilder.XMLGetNewIsotopeBySymbol(trackableBehaviour.TrackableName);
    IMolecularFormulaSet molecularFormulas = GetAllPossibleFormulasWithNewIsotope(newIsotope);

    if (molecularFormulas.Count < 0)
        return;

    ChemicalInstanceProvider.MolecularFormulaRange.GenerateMolecularFormula(molecularFormulas.First());
}

```

Рисунок 4.20 — Метод конфігурації знайдених на даний момент зображень та виклик функціонала для генерації та відображення молекули

5 АНАЛІЗ РЕАЛІЗОВАНОГО ПРОДУКТУ

Після завершення реалізації додатка було проведено його апробацію, виявлено та виправлено деякі недоліки. Результати атестаційної роботи були, зокрема, апробовані на Міжнародній науково-практичній конференції та опубліковані у науковій статті [22], [23].

Додаток був протестований на пристроях різних операційних систем, таких як Android, Windows та iOS. Під час тестування була виявлена така проблема: через використання метода пошуку цільового зображення на кожному фреймі додатку кількість зайнятої оперативної пам'яті суттєво збільшилась, через що прийшлося примусово зменшити частоту викликання методу з кожного фрейму додатку при роботі графічного процесора. Інших недоліків не було виявлено, додаток працює стабільно і не викликає системних помилок, які могли би призвести до збоїв.

Розроблений додаток, як додаток для освітнього процесу відповідає усім необхідним потребам:

- прийнятний розмір додатку для мобільних операційних систем;
- кросплатформеність;
- стабільна робота на різних процесорах смартфонів, навіть на версії Android 4.0 та iOS 8.0;
- відсутність необхідності з'єднання з інтернетом;
- безперервна робота у режимі обробки відео потоків;
- освітній зміст.

Інтерфейс додатку використовує виключно зображення з камери пристрою та кнопки, які необхідні лише на етапі розробки для перезавантаження додатку. Приклад візуалізації інтерфейсу наведено на рисунку 5.1.

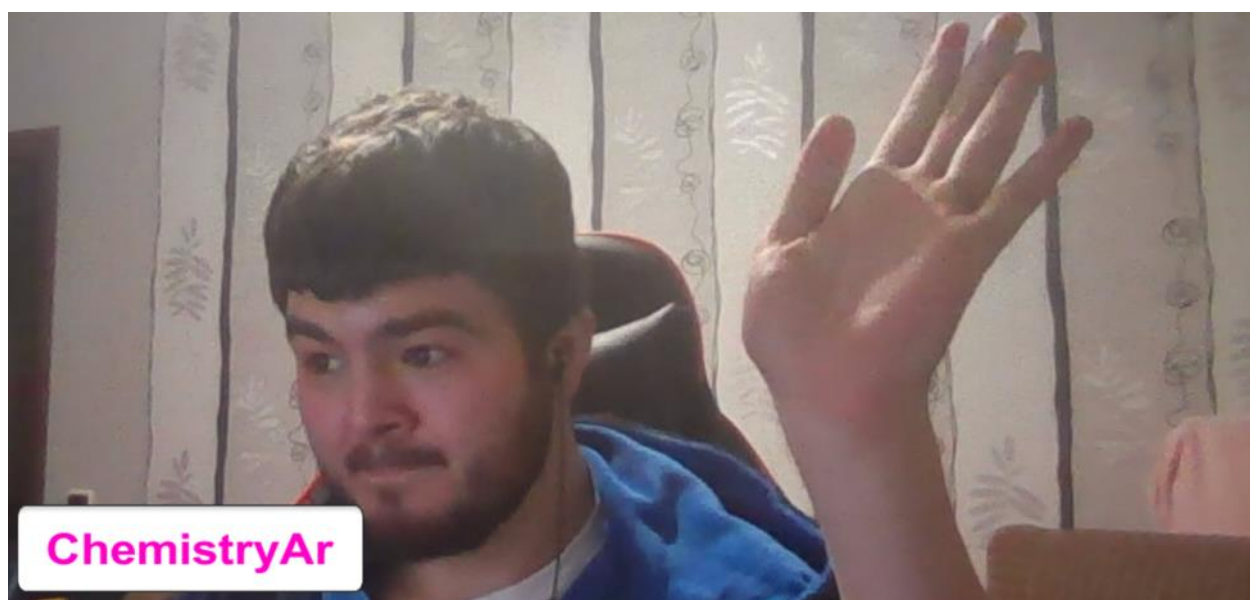


Рисунок 5.1 – Візуалізація інтерфейсу додатку

У разі знаходження у відео потоці цільового зображення починає працювати алгоритм, який візуалізує зображення хімічного елемента (рисунок 5.2, рисунок 5.3).



Рисунок 5.2 – Візуалізація роботи додатка з одним атомом

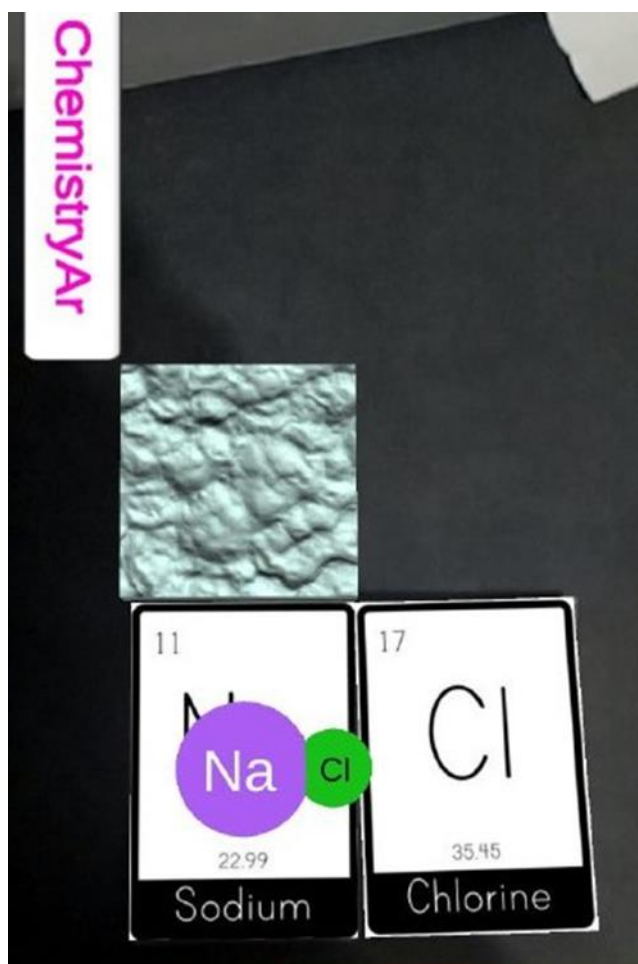


Рисунок 5.3 – Візуалізація роботи додатку з формуванням молекули «NaCl» на смартфоні з операційною системою Android

Через те, що додаток розроблявся не тільки для ПК з операційною системою Windows, а також і для смартфонів, то необхідно було здійснити відповідну оптимізацію додатку. Оцінити оптимізацію та знайти “вузькі” точки додатка дозволив один із інструментаріїв Unity – Profiler, що можна використовувати для отримання інформації про ефективність розробленого додатку. Існує можливість підключення його до пристроїв, які знаходяться у одній мережі, або пристроїв, підключених до персонального комп’ютера, щоб перевірити, як працює додаток на необхідній версії операційної системи. На рисунку 5.4 наведено приклад графіку використання CPU (процесору).

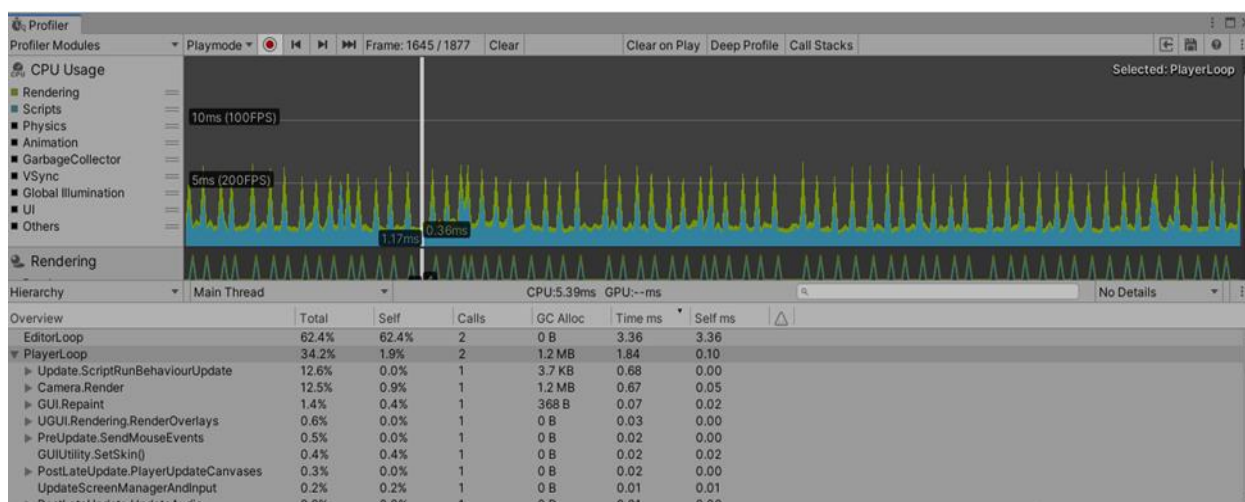


Рисунок 5.4 – Аналіз навантаження на центральний процесор

Як можна бачити з графіку, то оптимізація роботи (з викликанням методу знаходження цільового зображення один раз на кожні три кадри) дещо поліпшила роботу додатка, так як зменшилась кількість алокацій. Через те, що додаток на постійній основі працює з відео потоком з камери та рендерингом створених об'єктів, то робота з відео пам'яттю має відповідати потрібному рівню. На рисунку 5.5 наведено графік навантаження на графічний процесор.

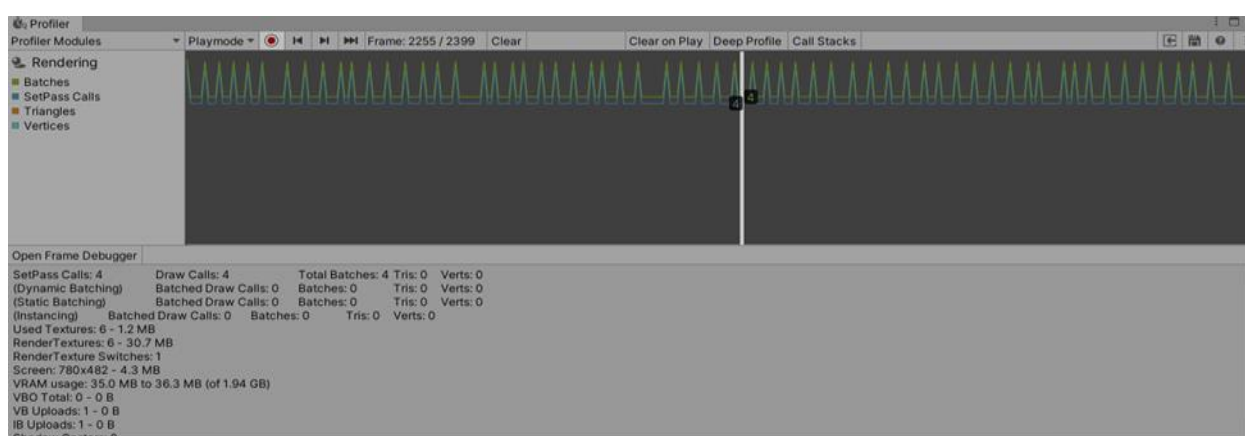


Рисунок 5.5 – Аналіз навантаження на графічний процесор

Через те, що кількість віртуальних об'єктів зводиться до мінімуму,

то навантаження на графічний процесор дуже низьке, а на кожному кадрі центральний процесор відправляє запит на обробку лише чотирьох об'єктів, що не завжди достатньо. Оскільки на мобільних платформах необхідно дотримуватися мінімальних затрат фізичної пам'яті, то було здійснено оцінку кількості пам'яті, що виділяється для стабільної роботи додатку, її мінімізація. Графік та інформація відносно кількості зайнятої фізичної пам'яті для роботи додатку наведено на рисунку 5.6.

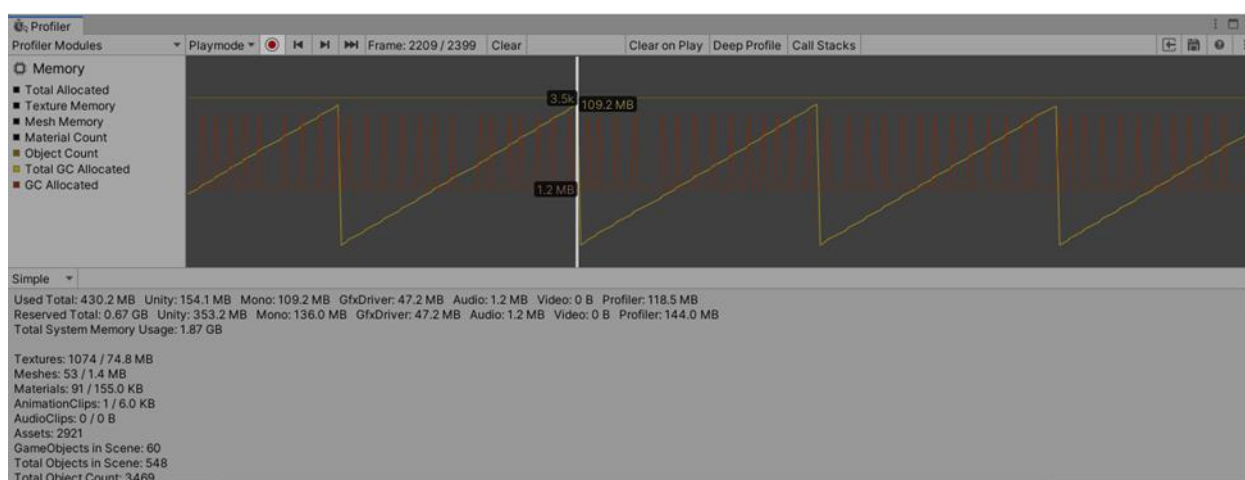


Рисунок 5.6 – Аналіз кількості необхідної фізичної пам'яті для роботи додатку

ВИСНОВКИ

Згідно з завданням в даній атестаційній роботі було розглянуто існуючі засоби розробки додатків, які стосуються використання доповненої реальності в хімічній галузі; зокрема, ігрових рушіїв, алгоритмів опису та знаходження зображень Fast.

Показано перспективність використання технології доповненої реальності при побудові моделей QSAR, що дозволяють по структурам хімічних сполук передбачати їх різноманітні властивості, включаючи властивості фізико-хімічного прогнозування.

За результатами аналізу можливостей відомих ігрових рушіїв для розробки додатку в магістерській роботі обрано ігровий рушій Unity, що має такі переваги: безкоштовність використання; можливість взаємодії з програмами на мові C#; мультиплатформеність.

Проаналізовано основні аспекти використання технології доповненої реальності у різних сферах, зокрема, в освітньому процесі. Були класифіковані системи, які використовують доповнену реальність, розглянуті та порівняні пристрої та необхідний набір апаратних засобів для функціонування доповненої реальності на різних пристроях.

Було проаналізовано декілька інструментів реалізації доповненої реальності та прийнято рішення використання платформи Vuforia з застосуванням віртуальної камери для аналізу відео потоку.

Для реалізації алгоритмів комп'ютерного зору було здійснено аналіз дескрипторних алгоритмів та прийнято рішення зупинитися на алгоритмі Fast, та його реалізації на мові C#, через його перевагу у швидкості над SIFT та SURF, що дуже важливо для оперативного аналізу відео потоку.

У результаті виконання магістерської роботи був розроблений додаток, який дозволяє аналізувати зображення із відео потоку та знаходити цільове зображення, для якого існує заздалегідь сформований

набір ключових точок (дескрипторів).

Доповнена реальність є однією з найбільш перспективних технологій сучасності, що дозволяє користувачеві сприймати більше корисної інформації про навколишній світ і інтерактивно впливати на нього за допомогою використання мобільних пристроїв, лептопів, окулярів віртуальної реальності тощо.

ПЕРЕЛІК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Brian X. Chen. If You're Not Seeing Data, You're Not Seeing. Wired, 2009. 213 p.
2. Azuma R., A Survey of Augmented Reality. Cambridge University Pres, 2005. 254 p.
3. Amit Y. 2D Object Detection and Recognition: Models, Algorithms and Networks. The MIT Press, 2002. 113 p.
4. Bay H., Tuytelaars T., Gool L. Surf: Speed up robust features // European Conference on Computer Vision, 2006. P. 404–417.
- 5 Hartley R., Zisserman A. Multiple view geometry in computer vision. 2nd edition. Cambridge University Press, 2004. 112 p.
6. Lepetit V., Fua P., Pilet J. Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation // Conference on Computer Vision and Pattern Recognition, 2004. P.125–243.
7. Mikolajczyk K., Schmid C. An Affine Invariant Interest Point Point Detector // European Conference on Computer Vision, 2002. P. 128–142.
8. Rosten E. Faster and better: a machine learning approach to corner detection, 2008. 96 p.
9. David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, 2004. 195 p.
10. Ronald T. Azuma A. Survey of Augmented Reality // In Presence: Teleoperators and Virtual Environments. 1997. No 4. P. 355–385.
11. I. Baskin, A. Varnek. «Fragment Descriptors in SAR/QSAR/QSPR Studies, Molecular Similarity Analysis and in Virtual Screening». In: Chemoinformatic Approaches to Virtual Screening, A. Varnek, A. Tropsha, eds., RCS Publishing, 2008, P. 1–43
12. N. S. Zefirov, V. A. Palyulin. Fragmental Approach in QSPR J. Chem. Inf. Comput. Sci. 2002 P. 1112–1122.

13. I. Baskin, A. Varnek. Building a chemical space based on fragment descriptors *Combinatorial Chemistry & High Throughput Screening* 2008. P. 661–668.

14. OpenCV Feature Detection Documentation URL: https://docs.opencv.org/3.4/db/d27/tutorial_py_table_of_contents_feature2d.htm (дата доступу – 31.03.2020)

15. EmguCV description and data API documentation URL: http://www.emgu.com/wiki/index.php/Main_Page (дата доступу – 07.04.2020)

16. NCDK description and data API documentation URL: <https://github.com/kazuuyajihara/NCDK> (дата доступу – 14.04.2020)

17. Ардаева А.А., Макаричева А.І. Огляд ігрових рушіїв для створення ігор. Наукове товариство студентів: міждисциплінарні дослідження: зб. ст. по мат. XXXI міжнар. студ. наук.-практ. конф. № 20 (31). URL: <https://sibac.info/archive/meghdis/20.pdf> (дата звернення: 02.05.2020)

18. ARCore description and data API documentation <https://developers.google.com/ar/reference> (дата доступу – 16.04.2020)

19. ARKit description and data API documentation <https://developer.apple.com/documentation/arkit> (дата доступу – 16.04.2020)

20. Wikitude SDK description and data API documentation <https://www.wikitude.com/documentation/> (дата доступу – 16.04.2020)

21. Чалая, Л.Э. Поиск неполных дубликатов в системах анализа цифровых изображений Л.Э. Чалая, П.Ю. Попаденко // Вісник КрНУ імені Михайла Остроградського. Вип. 5/2014 (88). – 2014. – С.42 – 47.

22. Берестовий, О.О. Системи когнітивного розвитку з використанням платформи VUFURIA // О.О. Берестовий, Д.І. Карпенко, С.Г. Удовенко // Обчислювальний інтелект (результати, проблеми, перспективи) (ComInt-2019): матеріали V Міжнар. науково-практичної конференції, 15-20 квітня. 2019 р. – Ужгород, 2019. – С. 165 – 166.

23. Чала Л.Е. Розробка інтерактивного музейного додатку з використанням технології доповненої реальності/ Л.Е. Чала, Д.І. Карпенко, О.О. Берестовий // Біоніка інтелекту. – 2019. – Вып. 2 (93). – С. 39-46