

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації
та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Другий (магістерський)
(рівень вищої освіти)

Розроблення автоматизованої складальної ділянки за допомогою методів
доповненої реальності
(тема)

Виконав:
студент 2 курсу, групи КІТПВм-20-1

Мажара Андрій Євгенійович
(прізвище, ініціали)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

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

Освітня програма Комп'ютерно-інтегровані
технологічні процеси і виробництва
(повна назва освітньої програми)

Керівник проф. Новоселов С.П.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАМ

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2021р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ
Кафедра _____ КІТАМ
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми _____ Освітньо-професійна
Освітня програма _____ КІТПВМ, Комп'ютерно-інтегровані технологічні процеси і виробництва
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАМ _____
(підпис)

« _____ » _____ 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Мажарі Андрію Євгенійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розроблення автоматизованої складальної ділянки за допомогою методів доповненої реальності

Затверджена наказом по університету від _____ 08.11.2021 № 1697Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 14.12.2021

3. Вихідні дані до роботи _____ середовище розробки Unity, Visual Studio, середовище моделювання Autodesk Fusion 360, графічний редактор Adobe Photoshop, QR та AR маркери, пристрій для розробки та відладки ПЗ.

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Вступ; 4.2 Аналіз використання доповненої реальності у сучасному виробництві; 4.3 Вибір інструментів для програмної реалізації та розробка алгоритму ПЗ; 4.4 Розробка та тестування додатку з використанням технології доповненої реальності; 4.5 Висновки; 4.6 Додатки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____ Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt). – 14 с. Формату А4.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літератури за темою кваліфікаційної роботи	08.11.21	Виконано
2	Аналіз використання доповненої реальності у сучасному виробництві	12.11.21	Виконано
3	Вибір інструментів для програмної реалізації та розробка алгоритму ПЗ	14.11.21	Виконано
4	Розробка та тестування додатку з використанням технології доповненої реальності	17.11.21	Виконано
5	Оформлення пояснювальної записки	05.12.21	Виконано
6	Подання роботи на перевірку Інтернет-сервісом Unichек	10.12.21	Виконано
7	Подання роботи на рецензію	10.12.21	Виконано
8	Отримання відгуку керівника	10.12.21	Виконано
9	Проходження попереднього захисту	12.12.21	Виконано
11	Подання роботи на підпис зав. кафедри		
12	Подання кваліфікаційної роботи в ЕК		
13			
14			

Дата видачі завдання 08.11.2021

Студент _____
(підпис)

Мажара А.Є.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

проф. каф. КІТАМ Новоселов С.П.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 81 с., 5 табл., 37 рис., 2 дод., 33 джерела.

ДОПОВНЕНА РЕАЛЬНІСТЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,
СЕРЕДОВИЩЕ РОЗРОБКИ, 3D-ОБ'ЄКТ, МІТКА, UNITY, ВИРОБНИЦТВО.

Об'єкт дослідження – методи реалізації доповненої реальності.

Предмет дослідження – програмне забезпечення з використанням методів доповненої реальності для автоматизованої складальної ділянки.

Методи дослідження – теоретичний аналіз методів реалізації доповненої реальності, принципу їх роботи, переваг та недоліків обраних методів реалізації та розробка програмного забезпечення доповненої реальності використовуючи середовище розробки та набори SDK.

Мета кваліфікаційної роботи – аналіз методів роботи доповненої реальності та відображення 3D-об'єктів у її середовищі, дослідження основних можливостей використання для покращення рівня автоматизації промисловості.

У кваліфікаційній роботі проаналізовано сучасний стан використання доповненої реальності на виробництві, досліджено методи реалізації доповненої реальності. Також проведено аналіз сучасних рішень.

На основі проведеного аналізу обрано інструменти для програмної реалізації та розроблено алгоритм ПЗ.

Відповідно до створеного алгоритму розроблено інтерфейс програмного засобу та його програмний код. Проведено експеримент з використання різних маркерів на їх зчитування на базі створеного додатку з використанням технології доповненої реальності. Отримані в ході проведення експерименту дані дозволили впевнитися у тому, що розроблений додаток відповідає усім поставленим вимогам та в подальшому може бути залучений до використання на складальній ділянці або використовуватися у навчальному процесі.

ABSTRACT

Explanatory note: 81 pp., 5 tab., 37 fig., 2 app., 33 sources.

AUGMENTED REALITY, SOFTWARE, DEVELOPMENT ENVIRONMENT, 3D OBJECT, MARK, UNITY, MANUFACTURING.

The object of research – methods of realization of augmented reality.

The subject of research – software using augmented reality methods for automated assembly area.

Research methods – theoretical analysis of augmented reality implementation methods, the principle of their work, advantages and disadvantages of selected implementation methods and augmented reality software development using SDK development environment and kits.

The purpose of the qualification work is to analyze the methods of augmented reality and display 3D-objects in its environment, to study the main possibilities of use to improve the level of industrial automation.

In the qualification work the current state of the use of augmented reality in production is analyzed, the methods of realization of augmented reality are investigated. An analysis of modern solutions was also conducted.

Based on the analysis, tools for software implementation were selected and a software algorithm was developed.

According to the created algorithm the software interface and its program code are developed. An experiment was conducted on the use of various markers and their reading on the basis of the created application using augmented reality technology. The data obtained during the experiment allowed to make sure that the developed application meets all the requirements and can be used for use in the assembly area or used in the learning process.

ЗМІСТ

Перелік скорочень.....	7
Вступ.....	8
1 Дослідження сучасного стану використання доповненої реальності на виробництві.....	10
1.1 Аналіз принципу роботи доповненої реальності.....	10
1.2 Аналіз методів реалізації доповненої реальності	12
1.3 Особливості використання технології доповненої реальності	16
1.4 Аналіз сучасних рішень з використанням доповненої реальності.....	19
1.5 Висновки до першого розділу.....	28
2 Вибір інструментів для програмної реалізації та розробка алгоритму ПЗ..	29
2.1 Опис засобів розробки та підготовка до створення додатку.....	29
2.2 Формування технічного завдання до розроблюваного додатку.....	35
2.3 Розробка структурної схеми програмного засобу.....	35
2.4 Розробка об'єктної моделі програмного засобу.....	37
2.5 Розробка алгоритму програмного засобу.....	39
2.6 Висновки до другого розділу.....	48
3 Розробка та тестування додатку з використанням технології доповненої реальності.....	49
3.1 Розробка інтерфейсу програмного засобу	49
3.2 Розробка програмного коду додатку	55
3.3 Проведення експерименту.....	65
3.4 Охорона праці.....	72
3.5 Висновки до третього розділу.....	75
Висновки.....	76
Перелік джерел посилання.....	78
Додаток А Лістинг програмного коду	82
Додаток Б Демонстраційний матеріал.....	98

ПЕРЕЛІК СКОРОЧЕНЬ

ООП – об’єктно-орієнтоване програмування;

ОС – операційна система;

ПЗ – програмне забезпечення;

САПР – система автоматизованого проектування і розрахунку;

ШІ – штучний інтелект;

API – прикладний програмний інтерфейс;

AR – доповнена реальність;

CAD – система автоматизованого проектування робіт або САПР;

CPU – центральний процесор;

GPU – графічний процесор;

IMU – інерційний вимірювальний блок;

SDK – набір засобів розробки;

SLAM – одночасна локалізація та картографування;

VI SLAM – візуальна інерційна одночасна локалізація та відображення;

VIO – візуальна інерційна одометрія.

ВСТУП

Розробка та реалізація використання методів доповненої реальності є провідним напрямом у розвитку сучасного програмного забезпечення. На сьогоднішній час використання доповненої реальності на виробництві має велике значення з масовим впровадженням Індустрії 4.0.

Можливість відображати об'єкти в доповненій реальності дозволяє детальніше їх дослідити, зрозуміти принцип роботи складних механізмів та пришвидшити процеси збірки об'єктів на виробництві.

Технології доповненої реальності значно підвищують продуктивність та ефективність виробничих компаній. Практичне застосування і переваги можуть бути величезними. Від зниження витрат на навчання та підвищення рівня утримання знань до швидшого виведення продуктів на ринок або підвищення ефективності складальних ліній та завдань з технічного обслуговування. З розвитком даної технології також зростають і можливості її використання, що позитивно впливає на впровадженні у різних напрямках промислового виробництва.

Об'єкт дослідження – методи реалізації доповненої реальності.

Предмет дослідження – програмне забезпечення з використанням методів доповненої реальності для автоматизованої складальної ділянки.

Методи дослідження – теоретичний аналіз методів реалізації доповненої реальності, принципу їх роботи, переваг та недоліків обраних методів реалізації та розробка програмного забезпечення доповненої реальності використовуючи середовище розробки та набори SDK.

Мета кваліфікаційної роботи – аналіз методів роботи доповненої реальності та відображення 3D-об'єктів у її середовищі, дослідження основних можливостей використання для покращення рівня автоматизації промисловості.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз принципу, методів та особливостей використання доповненої реальності;
- провести аналіз сучасних рішень за обраним предметом дослідження;
- обрати засоби розробки для підготовки до створення додатку;
- сформулювати технічне завдання для розробки додатку;
- розробити структурну схему, об'єктну модель та алгоритм роботи ПЗ;
- провести розробку інтерфейсу додатку та його програмного коду;
- використовуючи розроблений додаток провести експериментальні дослідження стосовно якості розпізнавання маркерів на різній відстані та різному освітленні.

Кваліфікаційна робота виконана згідно ДСТУ 3008-15 [1] та керуючись положеннями про вищу освіту [2-3] та стандартами освіти та науки України [4].

Під час навчання на другому (магістерському) рівні матеріали, що були використані для написання кваліфікаційної роботи було апробовано на міжнародних конференціях [5-6].

1 ДОСЛІДЖЕННЯ СУЧАСНОГО СТАНУ ВИКОРИСТАННЯ ДОПОВНЕНОЇ РЕАЛЬНОСТІ НА ВИРОБНИЦТВІ

1.1 Аналіз принципу роботи доповненої реальності

Провідними напрямками будь-якого сучасного виробництва сьогодні є підвищення таких основних показників як якість, швидкість виготовлення та надійність продукції, що виготовляється, а також покращення роботи операторів при взаємодії з процесами управління та контролю рівня безпеки експлуатації. Завдяки підвищенню даних показників виробництво зможе виробляти більш конкурентоспроможну продукцію та буде мати менші витрати в процесі виробництва. Варіантом реалізації поставлених задач може виступати впровадження використання на етапах виробництва доповненої реальності [7].

Доповнена реальність (AR) – це одна з передових технологічних тенденцій в даний час, і вона тільки зростає в міру того, як пристрої з підтримкою AR стають доступнішими по всьому світу. Доповнена реальність включає віртуальне накладання візуальної інформації на навколишній світ для поліпшення досвіду його сприйняття. Тобто завдяки використанню доповненої реальності можна спостерігати за відображенням будь-якої графічної інформації в реальному часі на сумісних електронних пристроях.

Доповнена реальність починається з пристрою, що оснащений камерою, зазвичай це може бути смартфон, планшет або розумні окуляри з встановленим програмним забезпеченням AR. Коли користувач спрямовує камеру пристрою на об'єкт, програмне забезпечення розпізнає його за допомогою технології комп'ютерного зору, яка аналізує відеопотік.

Далі пристрій завантажує інформацію про об'єкт з бази даних подібно до того, як веб-браузер завантажує сторінку через URL-адресу. Основна відмінність полягає в тому, що інформація в доповненій реальності представлена у вигляді тривимірного досвіду, накладеного на об'єкт, а не на двовимірній сторінці на екрані. Доповнена реальність покладається на навколишнє оточення із

проектуванням на нього віртуальних зображень. Таким чином, те, що бачить користувач, є частково реальним, а частково цифровим. Реальний світ при цьому залишається центральним елементом досвіду та доповнюється віртуальними елементами, які зазвичай з ним взаємодіють.

Доповнена реальність може забезпечити перегляд даних, що надходять від продуктів у реальному часі, та дозволити користувачам керувати ними за допомогою сенсорного екрану, голосу чи жестів. Наприклад, користувач може торкнутися кнопки зупинки на цифровому графічному інтерфейсі в доповненій реальності чи просто сказати слово «стоп», щоб відправити голосову команду продукту про зупинення виконання поточної операції. Оператор, що використовує гарнітуру AR для взаємодії з промисловою лінією, може побачити дані про продуктивність лінії та отримати швидкий доступ до її елементів управління.

Коли користувач переміщає камеру, що отримує інформацію з реального світу, розмір і орієнтація даних в середовищі доповненої реальності автоматично підлаштовуються під контекст, що змінюється. З'являється нова графічна чи текстова інформація, тоді як інша інформація, що не потрапила до поля зору, тимчасово або повністю перестає відображатися. У промислових умовах користувачі з різними ролями, такі як оператор верстата та технік з обслуговування, можуть дивитися на той самий об'єкт, але їм можуть бути представлені різні можливості доповненої реальності, адаптовані до їх потреб.

Тривимірна цифрова модель, що знаходиться в базі даних – «цифровий двійник» об'єкта, що об'єднує смарт-об'єкт та доповнену реальність. Ця модель може бути створена за допомогою автоматизованого проектування, що зазвичай виконується під час розробки продукту, за допомогою технології оцифрування фізичних об'єктів, або шляхом створення відповідної тривимірної моделі в спеціалізованому програмному забезпеченні. Потім разом з графічною інформацією, що відображається в доповненій реальності відображається зібрана інформація з продукту, бізнес-систем та зовнішніх джерел, щоб відобразити поточний реальний стан продукту [8].

Важливою частиною при роботі програмного забезпечення з доповненої реальності є апаратне забезпечення. Так, для роботи доповненої реальності на смартфоні, потрібен доступ до CPU, GPU та датчиків, які можуть підтримувати комп'ютерний зір та обробку отриманої інформації. Наприклад, гіроскоп вимірює положення телефону, а датчики приближення визначають, наскільки далеко об'єкти знаходяться відносно камери з якої іде відеопотік. Датчики світла вимірюють яскравість, акселерометри виявляють зміни у русі, а датчики глибини досліджують відстань.

Апаратне забезпечення зазвичай може бути вбудоване в смартфон і використовувати його основну камеру для збору інформації. В якості альтернативи можна також використовувати сторонню камеру з кращим набором камер та датчиків для більш точного відображення об'єктів у доповненій реальності.

1.2 Аналіз методів реалізації доповненої реальності

Значну роль в алгоритмі роботи програмного забезпечення AR має обраний метод реалізації, який визначає принцип розташування об'єктів, що відображаються в доповненій реальності. Загалом для забезпечення бездоганного сприйняття зображеної інформації важливо, щоб цифровий контент точно розміщувався в реальному світі. Існують різні техніки та методи, що допомагають у цьому. Серед таких методів є два основних, що найчастіше використовуються – маркерний та безмаркерний. Робота цих методів базується на використанні алгоритмів комп'ютерного зору для розпізнавання об'єктів з відеопотоку та їх доповнення [9].

Маркерний метод реалізації доповненої реальності був розроблений як основа для AR. Робочий процес алгоритму маркерної AR-навігації складається з 3 етапів:

- розпізнавання пристрою візуалізації;
- розпізнавання маркера;

– реалізація та реєстрація AR у робочому середовищі.

Маркер у доповненій реальності – це зображення або об'єкт, який може бути розпізнаний мобільним додатком з підтримкою AR і використовується для запуску функцій доповненої реальності.

У якості маркеру може виступати об'єкт, наприклад, прямокутник або квадрат з вписаною усередину QR-міткою, розташований у навколишньому просторі, що знаходиться та аналізується спеціальним програмним забезпеченням для подальшого відтворення віртуальних об'єктів.

Маркер прикріплюється до місця поряд з об'єктом, що збігається, яке має мінімальне відображення світла та не буде перекриватися іншими об'єктами. Тримач QR-маркера виготовляють для забезпечення розміщення маркера у тих самих умовах під час перевірки точності реєстрації, де він і буде використовуватися. Щоб звести до мінімуму похибки, що можуть виникнути під час руху, пристрій з встановленим програмним забезпеченням доповненої реальності, що оснащений камерами та датчиками, фіксується у положенні, в якому можна добре розпізнати встановлений маркер.

Запуск програми після налаштування середовища для пристрою та маркера дозволяє оцінити відносне положення та напрямок пристрою у навколишньому середовищі. Цей процес виконується за допомогою алгоритму одометрії. Існують різні типи алгоритмів одометрії в залежності від джерела даних, що використовується для розрахунку. Зазвичай використовують візуальну інерційну одометрію (VIO), що поєднує в собі візуальну одометрію з використанням зображень з камери та інерційну одометрію з використанням інерційного вимірювального блоку (IMU), включаючи використання інформації отриманої від акселерометру та гіроскопу (рис. 1.1).

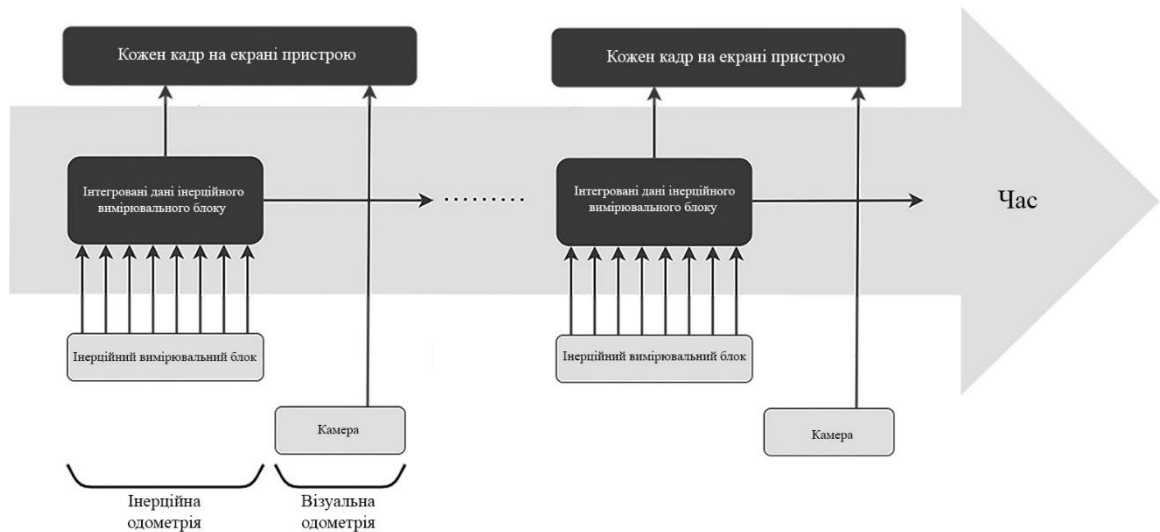


Рисунок 1.1 – Процес роботи алгоритму візуальної інерційної одометрії за допомогою використання камери та інерційного вимірювального блоку

Візуальна інерційна одометрія оцінює розташування пристрою в реальному середовищі, встановлюючи початкове значення розташування пристрою на основі даних інерційного вимірювального блоку, що має високу швидкість збору, а потім калібрує його за допомогою зображень з камери. Сучасні набори засобів розробки підтримують процес закриття циклу, що додатково коригує траєкторію, зіставляючи траєкторію з початковою точкою під час переміщення пристрою та повертаючись до початкової точки під час розрахунку з алгоритмом VIO. Цей процес називається візуальною інерційною одночасною локалізацією та відображенням (VI SLAM). Процес замикання петлі може не використовуватися через характер цієї системи AR, де пристрій працює у фіксованому стані.

Зображення, що використовуються для оцінки положення розпізнаного камерою, також використовуються для алгоритму виявлення ознак зображення, який розпізнає маркер, як правило, з урахуванням кута і перетину ліній або частини зображення з чітким контрастом кольору, наприклад чорного і білого в якості характерної точки відповідного зображення, щоб визначити, що

зображення однакове, на основі положення кожної характерної точки, що залишається незмінною [10].

Хоча у якості маркера може бути використане будь-яке зображення, доцільно буде дотримуватись встановлених рекомендацій при його виборі. Виходячи з обмежень роздільної здатності камери пристрою, особливостей сприйняття кольору, освітлення та обчислювальної потужності обладнання, для швидкої роботи в реальному часі зазвичай обирають чорно-білий маркер простої форми, він повинен мати велику кількість ознак, за якими можна буде встановити відповідність оригіналу.

Алгоритм виявлення особливостей зображення має високу швидкість роботи на зображеннях з багатою текстурою і високою контрастністю, тому зображення маркера повинне мати ті ж характеристики і має бути виготовлене з матеріалу, який погано відбиває світло і не деформується, щоб гарантувати точне розпізнавання зображення камерою.

Після визначення розташування пристрою та розпізнавання маркера масштаб об'єктів у віртуальному просторі визначається шляхом обчислення відстані між пристроєм і маркером за розміром розпізнаного маркера на попередньому кроці. Далі, коли всі необхідні дані розраховані, віртуальна 3D-модель відображається у вказаному від маркера місці для підтвердження її встановлення користувачем. У разі неточності при розміщенні об'єкта у доповненій реальності користувач може відрегулювавши положення тривимірної віртуальної моделі. Після підтвердження алгоритм розпізнавання зображень завершується, і програма залишається активною без постійного розпізнавання маркерів [11].

Безмаркерний метод схожий з маркерним методом через те, що вони використовують ті самі алгоритми для визначення положення камери та об'єктів відповідно до неї, але різняться алгоритмами, завдяки яким об'єкти в доповненій реальності «прикріплюються» до реального світу.

Безмаркерний метод реалізації доповненої реальності, зазвичай, підтримується датчиком глибини або чимось подібним, та використовуючи

фреймворк, наприклад VI SLAM з використанням VIO, постійно створює віртуальне уявлення реального світу.

Далі ця віртуальна копія реального світу використовується в основному для наступних цілей:

- виявлення горизонтальних та вертикальних поверхонь у реальному світі;
- використання віртуальної сітки реального світу для оклюзії.

Використовуючи просторове відображення та алгоритми обраного набору засобів розробки у реальному просторі визнаються віртуальні якорі, відповідно до яких можна позиціонувати віртуальний контент. В готових рішеннях така інформація, як виявлені поверхні та віртуальні якорі, в основному не відображається для користувача, так як в цьому немає необхідності.

Навіть якщо об'єкти знаходяться поза увагою користувача, вони не переміщаються, коли користувач переміщається, і користувачеві не потрібно сканувати нові зображення. Коли користувач повертається у цей простір, віртуальний контент буде знаходитися у тому місці.

Важливе значення у питанні просторового позиціонування контенту займає такий термін як оклюзія – перекриття віртуального контенту об'єктами реального світу. Щоб створити ідеальну ілюзію віртуального контенту у реальному світі, оклюзія необхідна і буде у центрі уваги розробки. Зараз оклюзії вже можна досягти без використання спеціальних датчиків глибини та створення досить якісної копії реального світу, достатньо лише смартфона з камерою.

1.3 Особливості використання технології доповненої реальності

Ізольовані додатки AR існують вже кілька десятиліть, але тільки недавно стали доступні технології, необхідні для розкриття її потенціалу. По суті, AR перетворює обсяги даних і аналітики в зображення або анімацію, які накладаються на реальний світ. Сьогодні більшість додатків AR надається через мобільні пристрої, але все більше і більше буде поширюватися використання

переносних пристроїв без допомоги рук, таких як дисплеї на голові або розумні окуляри.

Відповідно до цілей Індустрії 4.0, багато виробників намагаються оптимізувати свої процеси, щоб швидко виробляти незмінно якісну продукцію [12]. Розумне виробництво ґрунтується на співпраці людей з провідними технологіями, і ця співпраця процвітає в міру поліпшення користувацького досвіду. Доповнена реальність (AR) доповнює користувацький інтерфейс, який представляє корисну інформацію, реальні ситуації та дані, у вигляді візуальних, звукових або тактильних сигналів.

Інформація в режимі реального часу, що надається за допомогою технологій доповненої реальності, може допомогти виробникам підвищити якість, ефективність та безпеку. AR має усі переваги автоматизованого контролю – економія часу, підвищення точності та виявлення неруйнівних дефектів – при цьому включаючи людський досвід та вибір на кожному кроці. Покрокове керівництво, пристосоване для кожного працівника та продукту, забезпечує вчасне виконання роботи за графіком. Візуальні, звукові та тактильні сповіщення інформують працівників про їх оточення та в разі потреби сповіщають для забезпечення безпеки.

Переваги AR для перевірки та контролю якості особливо яскраво проявляються при навчанні нових, недосвідчених працівників, але AR все ще надає значну користь досвідченим працівникам.

Доповнена реальність змінює поняття про інструкції, тренування та тренінги. Ці важливі функції, що покращують продуктивність праці, є дорогими та трудомісткими, і часто дають неоднакові результати. Наприклад, письмові інструкції по збірці часто важкі та трудомісткі. Стандартні навчальні відео не є інтерактивними та не можуть адаптуватися до індивідуальних потреб у підготовці спеціалістів. Очне навчання коштує дорого і вимагає, щоб студенти і викладачі збиралися в одному місці, іноді неодноразово. І якщо обладнання, роботі з яким навчають студентів, відсутнє, їм може знадобитися додаткове навчання, щоб перенести вивчене в реальний контекст.

Доповнена реальність вирішує ці проблеми, надаючи в реальному часі покрокові візуальні вказівки щодо таких завдань, як збірка продукції, експлуатація машини та вибір матеріалів на складі. Складні двовимірні схематичні уявлення процедури в посібнику, наприклад, стають інтерактивними тривимірними голограмами, які проводять користувача через необхідні процеси. Завдяки таким методам можна ефективно впровадити навчання як нових так і існуючих спеціалістів [13].

Це пояснює, чому доповнена реальність така потужна. Немає кращого графічного інтерфейсу, ніж фізичний світ, який ми бачимо навколо нас, коли він доповнюється цифровим накладенням відповідних даних та вказівок там, де і коли вони необхідні. Доповнена реальність усуває залежність від позаконтекстної і важко оброблюваної двовимірної інформації на сторінках і екранах, значно покращуючи нашу здатність розуміти і застосовувати інформацію в реальному світі.

На виробництві процеси часто бувають складними, вимагають сотень, а то й тисяч кроків, а помилки дорого коштують. Доповнена реальність може доставити потрібну інформацію в той момент, коли це потрібно працівникам на складальних лініях, зменшуючи помилки, підвищуючи ефективність та продуктивність.

Доповнена реальність на підприємствах може збирати інформацію з систем автоматизації і управління, вторинних датчиків і систем управління активами та робити видимими важливі дані моніторингу і діагностики про кожну машину чи процес. Перегляд такої інформації, як ефективність і відсоток браку в контексті, допомагає технічним фахівцям з обслуговування зрозуміти проблеми і спонукає заводських робітників проводити профілактичне обслуговування, що може запобігти дорогим простоям.

За оцінками спеціалістів, складські операції складають близько 20% всіх витрат на логістику, в той час як збір товарів з полиць складає до 65% складських витрат. На більшості складів робітники як і раніше виконують це завдання,

звіряючись з паперовим списком речей, які потрібно зібрати, а потім шукають їх. Цей метод повільний і схильний до помилок.

Інструкції доповненої реальності направляють робітників до місця, де потрібно взяти кожен продукт, а потім пропонують кращий маршрут для таких продуктів.

Додатки доповненої реальності можуть розпізнавати фізичну середу по різному. Щоб точно накласти цифрову інформацію на фізичний світ, технології AR повинні розпізнавати те, на що вони дивляться. Найпростіший підхід – визначити місце розташування пристрою AR, використовуючи, наприклад, GPS, і показати відповідну інформацію для цього місця розташування, не прив'язуючи її до конкретного об'єкта. Це відомо як «незареєстрований» досвід доповненої реальності. Навігаційні дисплеї транспортних засобів зазвичай працюють таким чином.

Інший підхід – "зареєстрований" – отримує інформацію про прив'язку до конкретних об'єктів. Вони можуть робити це за допомогою маркерів, таких як штрих-коди, логотипи або етикетки, які розміщуються на об'єктах і скануються користувачем за допомогою пристрою доповненої реальності. Однак більш потужний підхід використовує технологію, яка розпізнає об'єкти, порівнюючи їх форму з каталогом тривимірних моделей. Це дозволяє техніку з обслуговування, наприклад, миттєво розпізнавати будь-який тип обладнання, за обслуговування якого він або вона відповідає, і взаємодіяти з ним, і робити це під будь-яким кутом. Хоча маркери є гарною відправною точкою, технології розпізнавання форм швидко розвиваються, і організаціям потрібна можливість реалізовувати їх для використання у багатьох новітніх додатках доповненої реальності.

1.4 Аналіз сучасних рішень з використанням доповненої реальності

Стрімкого розвитку та масового використання доповнена реальність здобула декілька років тому. Основні чинники, що цьому посприяли – це

розвиток мобільних систем та інструментів для створення доповненої реальності.

Мобільні пристрої за останні роки досягли значних результатів у можливості роботи з графічним відображенням доповненої реальності, де на зображення з камери накладається не тільки графічна інформація у вигляді тексту, а й у вигляді змодельованих тривимірних об'єктів. У свою чергу такі об'єкти можуть бути досить деталізованими і потребувати значних ресурсів на відображення і взаємодію з реальним світом, наприклад переміщення об'єкту в середовищі доповненої реальності за зображенням маркеру отриманого з камери пристрою.

Протягом останніх кількох років доповнена реальність привернула велику увагу з боку венчурного ринку та отримала значні інвестиції. Завдяки широким можливостям застосування для підприємств, комерційних та промислових просторів AR продовжує залучати інвестиції.

Промисловість стикається з кризою, пов'язаною зі швидким масовим впливом кваліфікованих кадрів із галузі, при цьому кількість нових кваліфікованих спеціалістів є недостатньою. Доповнена реальність являється ефективним інструментом для допомоги промисловим працівникам. Завдяки віддаленому розширеному зв'язку та розширеним операціям AR може допомогти галузі краще вирішити проблему нестачі навичок у нових працівників.

Галузь повинна швидко та ефективно навчати своїх нових працівників. Доповнена реальність також стала ключовим інструментом для навчання. Завдяки здатності AR створювати прозорий ефект і об'єднувати фізичний і віртуальний світи, це може допомогти створити ефективніші інтерактивні навчальні заняття.

Цифрова трансформація руйнує багато аспектів, пов'язаних з організацією та функціонуванням компаній. У багатьох випадках це допомагає прокласти шлях до успішного впровадження AR. По-перше, робоча сила стає все більш відкритою для нових технологій і змін у своїх робочих процедурах. По-друге, усі

технологічні елементи, які ще більше покращують потенціал AR, вже встановлені або будуть встановлені пізніше.

Однією з основних переваг доповненої реальності є те, що вона не потребує спеціального обладнання. Машинне навчання та досягнення в галузі технологій комп'ютерного зору дозволили створювати потужні програми AR для планшетів та мобільних телефонів. Ці мобільні рішення вже стають звичайним явищем у промисловому світі, полегшуючи промисловим організаціям впровадження технологій доповненої реальності. Використання доповненої реальності на промисловості вже являється масовим явищем.

Однією з провідних компаній, що працюють у галузі імерсійних додатків та створюють сучасні ефективні рішення для промисловості є Program-Ace.

Серед розроблених додатків, що компанія може запропонувати, можна виділити Remote AR/MR Assistant.

Remote AR/MR Assistant – це ефективне рішення, яке дозволяє технічним спеціалістам та галузевим експертам працювати разом за допомогою інтерактивного інтерфейсу AR або MR. В результаті чого буде отримана необхідна допомога та вказівки при виконанні завдання на місцях, а також можна поділитися своїм досвідом та ефективніше навчати молодих співробітників. Рішення забезпечує можливість підключення за допомогою голосу, відео та інтерактивного віртуального набору інструментів, що доповнює реальний світ. Незалежно від того, чи це доповнена чи змішана реальність, Program-Ace об'єднає всі необхідні функції та інтерактивні інструкції, щоб надати експертам і працівникам на місцях повний інструмент, який допоможе підвищити продуктивність їхньої роботи [14].

Основні переваги:

- гнучкі та захоплюючі інструменти для отримання знань на запит;
- робочі інструкції та технічна підтримка в режимі реального часу;
- надає можливість мінімізувати час до дозволу та час простою;
- дозволяє мінімізувати витрати на відрядження та максимізувати прибуток від послуг;

– відкриває двері для розширення та масштабування технічних знань.

Ціна готового до використання рішення AR/MR, розробленого за один місяць, з періодом окупності менше ніж рік становить 25000 доларів.

Також Program-Ace може запропонувати Virtual Product Configurator [15]. Ціна такого готового рішення становить 15000 доларів.

Важливо додати, що приведені ціни будуть значно відрізнятися, при внесенні індивідуальних потреб замовника. Аналогічна політика ціноутворення справедлива також для інших відомих виробників, що надають послуги по розробці подібних додатків з AR.

Так як інформація про розроблені додатки з використанням AR, що використовуються на промисловості, не освітлюється у повній мірі, було прийнято дослідити основні напрямки використання сучасних рішень.

До основних напрямків використання AR у промисловості можна віднести наступні:

- навчання;
- віддалена спільна робота;
- складальні операції;
- цифровий двійник.

Навчання нових спеціалістів, як і підтримка навичок існуючих кваліфікованих спеціалістів займає значну роль для будь-якого підприємства. Промисловості необхідно швидко та ефективно проводити навчання нових працівників. Декілька років тому віртуальна реальність рекламувалася як рішення наступного покоління, яке виводить навчання операторів на новий рівень. Віртуальна реальність може створювати ефективні тренувальні програми з великим полем зору та відчуттям присутності. Однак створення якісних та деталізованих подібних рішень з використанням віртуальної реальності займає досить багато часу та коштує відповідно. Доповнена реальність змінює дане положення речей, та надає можливість створювати сучасні інструменти для навчання. Завдяки здатності AR створювати прозорий ефект та поєднувати

реальний та віртуальний світи, вона може допомогти у створенні інтерактивних тренувань.

Багато компаній, які займаються розробкою інструментів для створення програмного забезпечення з використанням AR вже пропонують свої рішення, що дозволяють користувачам створювати власні процедури розширеного навчання. У той час як прості процедури навчання можуть бути створені для смартфонів та планшетів, для програм розширеного навчання потрібне складніше обладнання, таке як окуляри RealWear та Microsoft HoloLens. Створення програм для розширеного навчання потребує виконання різних завдань послідовно, одягнувши інтелектуальне спорядження та записуючи всі свої кроки разом із голосовими інструкціями. Тренери також можуть додавати до своїх навчальних програм розширені об'єкти, інструкції та мітки. Після створення навчального додатку слухачі можуть багаторазово використовувати ту саму програму. Стажери носять обладнання AR та отримують покрокові інструкції. Коли вони завершують кожен крок, AR розпізнає це і дає слухачам наступну інструкцію [16].

Віддалена спільна робота являється однією з областей, у якій доповнена реальність виявляється дуже корисною. Промисловим професіоналам часто доводиться співпрацювати з іншими дистанційними експертами. Використовуючи AR, вони можуть ефективніше вирішувати поставлені питання при дистанційному спілкуванні. Сучасні додатки AR дозволяють використовувати розумні окуляри, телефони та планшети. Це дозволяє користувачам зв'язуватися з віддаленими експертами за допомогою відеодзвінків та отримувати інструкції у режимі реального часу з використанням діаграм, маркерів, інструкцій та анотацій у полі зору користувача в реальному часі. Багато сучасних програм також дозволяють користувачам переглядати важливу інформацію про завдання в межах їхнього зору. З цифровим контентом, накладеним на відеопотік реального часу, доповнена реальність надає віртуальну практичну допомогу.

Виробники обладнання, яким необхідно надавати послуги з технічного обслуговування свого обладнання на різних підприємствах, можуть отримати значну вигоду від віддалених розширених інструментів спільної роботи, щоб знизити витрати на обслуговування та підвищити прибутковість цієї діяльності. Багато виробників обладнання в даний час надають кінцевим користувачам інструменти спільної роботи у доповненій реальності в рамках своїх послуг з технічного обслуговування. Використовуючи ці інструменти, кінцеві користувачі можуть самостійно вирішувати питання по обслуговуванню обладнання з дистанційним залученням експертів їх виробника. Подібні інструменти допомагають кінцевим користувачам усувати неполадки в робочих умовах і вирішувати їх швидше та простіше, ніж при очній присутності відповідних спеціалістів [17].

Доповнену реальність також можна використовувати для спрощення повторюваних операцій та процедур для промислових робітників. Саме тому доповнену реальність охоче використовують при виконанні складальних операцій. Розширені інструкції допомагають збирачам в автомобільній, напівпровідниковій та інших галузях промисловості виконувати завдання ефективно, швидко та з меншою кількістю помилок. Сучасні складальні операції можуть бути складними та потребувати від працівника швидких рішень та дій, коли необхідно збирати сотні складних компонентів.

Традиційно працівникам доводилося покладатися на друковані інструкції та запам'ятовувати необхідні кроки. Додатки з доповненою реальністю допомагають виконувати ці вказівки і робити їх легко видимими в полі зору робітника. У міру того, як співробітники переходять від одного кроку до іншого, програма надає їм необхідні інструкції. Багато програм також дозволяють записувати етапи складання на відео, тому, якщо виникне потреба, працівник може звернутися до раніше записаних відео зі складання поточного обладнання.

Використання доповненої реальності досить широко реалізовано, що дає можливість її впровадження у значну кількість виробничих процесів та допомогти робітникам на виробничих підприємствах при виконанні їх звичних

робочих завдань. Сучасні додатки доповненої реальності використовуються на підприємстві для низки операцій з технічного обслуговування та перевірки. Інтеграція доповненої реальності з іншими рішеннями для управління активами, може принести користь кінцевим користувачам за рахунок спрощення завдань обслуговування. Додатки AR можуть отримувати замовлення на виконання роботи для обслуговуючого персоналу, показувати технічним фахівцям найкращий маршрут до несправного об'єкту, відображати записи технічного обслуговування та отримувати інструкції з експлуатації та показання даних. У разі правильної інтеграції додатки AR також можуть допомогти співробітникам вводити дані про обслуговування та перевірку за допомогою простих голосових команд [18].

Технологія AR вже виявилася дуже корисною для промислового світу. Штучний інтелект (ШІ) може ще більше покращити ціннісні пропозиції. По-перше, ШІ може спростити процес створення контенту AR. Багато програмних засобів для розробки AR вже включають ШІ, щоб користувачам, які не мають великого досвіду при роботі з програмним забезпеченням, було легко створювати контент AR. Штучний інтелект може допомогти програмним інструментам AR швидко аналізувати форму об'єктів, їх краї, криві та симетрію, щоб розпізнавати, відстежувати об'єкти та розміщувати тривимірні об'єкти. По-друге, ШІ можна інтегрувати з додатками доповненої реальності для підвищення ефективності експлуатації та технічного обслуговування. ШІ може допомогти у створенні ефективних розширених процедур та інструкцій, заснованих на ситуації, умовах процесу та досвіді оператора. Оскільки кожен сеанс записується, його можна використовувати для подальшого покращення процедури та пов'язаних із нею розширених інструкцій [19].

Цифровий двійник є цифровим уявленням об'єкта, системи чи процесу. У виробництві цей двійник може відтворити будь-що, навіть найдрібніший компонент, і весь ланцюжок поставок. Як правило, програмне забезпечення двійника аналізує роботу об'єкту/системи/процесу на основі реальних

вимірювань, у той час як функції AR надають візуальні ефекти та відповідні дані, що будуть зрозумілі користувачеві.

Розробка цифрових двійників трохи складніша, ніж для інших типів програмного забезпечення, тому що інтерактивні моделі, які будуть представлені через AR, повинні працювати так само, як об'єкт у реальному житті. Таким чином, до об'єкта застосовуються датчики та вимірювання, а дані, зібрані за певний період функціонування, необхідно перетворити на цифрову форму. В іншому процес створення моделі та надання їй ефекту імерсивності досить стандартний [20].

Інструменти для створення доповненої реальності, що являють собою середовища розробки почали стрімко розвиватися після виходу платформ для мобільних додатків з доповненою реальністю ARCore та ARKit від таких технологічних гігантів, як Google та Apple. Їх популяризація дозволила привернути достатньо уваги, що позитивно вплинуло, як на окремих користувачів, так і на безліч фірм, що тепер мають змогу створювати власні комерційні рішення.

До основних платформ розробки подібних додатків, окрім ARCore та ARKit, можна також віднести і наступні:

- Vuforia;
- AR Foundation;
- Kudan;
- Wikitude;
- MaxST.

Вищезазначені середовища мають свої особливості у реалізації та використанні, а в основі їх роботи лежить обробка відеопотоку з камери пристрою, виявлення горизонтальних та вертикальних поверхонь, відстеження об'єктів, зображень та маркерів.

До головних переваг використання ARCore можна віднести можливість створення додатків для пристроїв на базі ОС Android та iOS і точність розміщення об'єктів у середовищі AR [21]. Головним недоліком є нестабільність

роботи подібних додатків при роботі на ОС Android, що може проявлятися на деяких девайсах.

Платформа ARKit не була розглянута як варіант для розробки. Основною засадою слугувало те, що розроблені додатки підтримуються тільки на приладах Apple, що виключає можливість кросплатформної розробки, а також унеможливорює проведення експериментів після завершення розробки додатку.

Набір засобів розробки Vuforia є досить зручним у використанні, швидко та точно позиціонує зображення на маркерах, а також, як і ARCore, підтримується на ОС Android та iOS [22]. Як і для вищезазначених платформ, має достатньо технічної документації для освоєння і розробки власного рішення додатку. Проте Vuforia не задовольняє поставленим вимогам. В основному через те, що безкоштовне використання має обмежену кількість сканувань маркеру, близько 1000 сканувань, а також при роботі створеного додатку на зображенні з камери присутній водяний знак «Vuforia». Дані обмеження можна зняти, але для цього необхідно придбати ліцензію.

Платформа AR Foundation від Unity – це спроба уніфікувати портативну розробку AR, пропонуючи загальний API, що підтримує основні функції від ARCore та ARKit, його дуже легко освоїти і він досить потужний [23]. Це досить зручно та дозволяє використовувати всі функції у рамках єдиного робочого процесу. AR Foundation є інтерфейсом для розробників Unity, але сам не реалізує жодних функцій AR. Щоб використовувати AR Foundation на цільовому пристрої, необхідно встановити пакети для цільових платформ, що офіційно підтримуються Unity. Так для платформи Android необхідно встановити модуль ARCore XR [24].

Набори засобів розробки, такі як Kudan, Wikitude, MaxST безперечно мають свої особливості та відмінності від вищезазначених пакетів. Проте їх не було розглянуто як варіант SDK для розробки. В основному через те, що AR Foundation задовольняє меті кваліфікаційної роботи. До основних переваг даного вибору можна віднести значну кількість технічної документації по налаштуванню та розробці на платформі AR Foundation, можливість створення

додатків для різних платформ, а також оновлення доступних функцій після їх реалізації. Зазначені особливості слугували основним фактором вибору даного набору засобів розробки для створення власного додатку з використанням методів доповненої реальності.

1.5 Висновки до першого розділу

Доповнена реальність може використовуватися на різних пристроях: екранах, окулярах, портативних пристроях, смартфонах, налобних дисплеях. Вона включає в себе такі технології, як SLAM (одночасна локалізація та картографування), depth tracking (відстеження глибини за допомогою даних датчика, що обчислює відстань до об'єктів), а також камери, датчики, обробку та відображення.

AR дозволяє накладати 3D-моделі на фізичний світ, підвищуючи здатність інженерів оцінювати та вдосконалювати дизайн, або накладати моделі САПР на фізичні прототипи, щоб порівняти, наскільки добре вони відповідають стандартам. Це підвищує точність процесу забезпечення якості, в якому раніше інженерам доводилося ретельно порівнювати двомірні креслення з прототипами, і робить його в 5-10 разів швидше.

У виробництві процеси часто бувають складними, вимагають сотень або навіть тисяч кроків, а помилки обходяться дорого. Доповнена реальність може надавати потрібну інформацію в той момент, коли вона необхідна робітникам на складальних лініях, скорочуючи кількість помилок, підвищуючи ефективність і підвищуючи продуктивність.

Отже, головна особливість AR полягає у тому, що ця технологія значно знижує кількість помилок і збільшує продуктивність на підприємствах та і взагалі у всіх сферах її використання, що позитивно відображається на її розвитку і впровадженню у все більшу кількість процесів, що можуть бути автоматизовані.

2 ВИБІР ІНСТРУМЕНТІВ ДЛЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ТА РОЗРОБКА АЛГОРИТМУ ПЗ

2.1 Опис засобів розробки та підготовка до створення додатку

Враховуючи результати аналізу використання доповненої реальності у сучасному виробництві, немає сумнівів, що даний напрям є провідним, а за його реалізацією і впровадженням стоїть майбутнє.

Реалізацію програмного забезпечення у вигляді мобільного додатку можна поділити на наступні етапи:

- аналітика;
- технічне завдання;
- проектування та дизайн;
- розробка;
- тестування та стабілізація;
- впровадження;
- підтримка та розвиток.

Спочатку необхідно встановити як саме і які задачі буде вирішувати додаток. Далі виконують докладний опис функціональності та дизайну, в результаті чого формується перелік функцій, що будуть реалізовані та висуваються певні вимоги до додатку [25].

На етапі розробки додатку з доповненої реальності необхідно обрати програмні засоби які буде використано.

У якості середовища розробки мобільного додатку було обрано Unity та мову програмування C#. До основних переваг обраного середовища розробки можна віднести наступні:

- зрозумілий редактор та інструментарій;
- документація на форумах, а також безліч відео-уроків;
- сучасний рівень графіки;

- надається умовно безкоштовно;
- велике ком'юніті розробників;
- можливість створення фотореалістичної графіки;
- легкий імпорт проекту між ОС Windows, Linux, OS X, Android, iOS на VR- та AR-пристрої.

Виконавши інсталяцію Unity 2020.3.20f1 LTS автоматично була ініціалізована Visual Studio, що буде використовуватися в процесі розробки для написання коду програми. Далі було встановлено модуль Android Build Support (рис. 2.1) з набором SDK, необхідним при розробці додатків для Android.

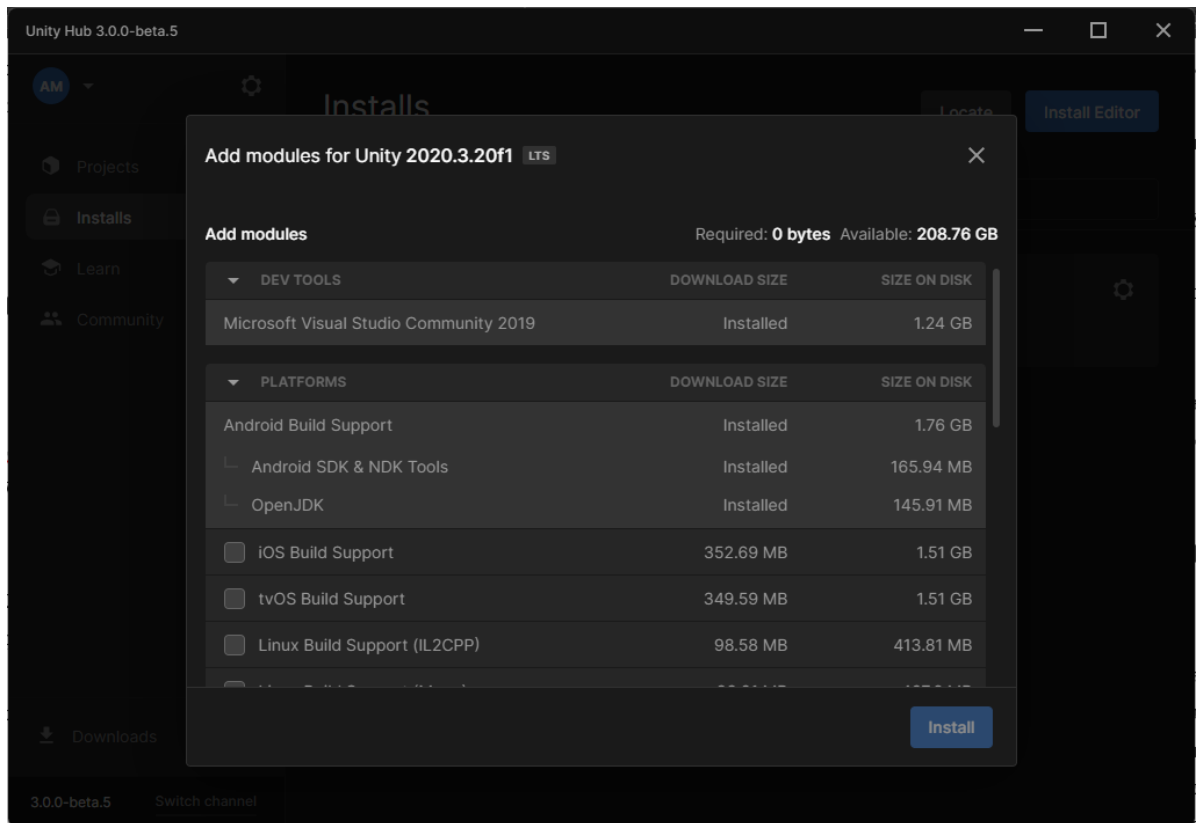


Рисунок 2.1 – Вікно інсталяції модулів в Unity

Без використання готового SDK (набір для розробки програмного забезпечення) важко уявити собі створення якісного додатку з хорошими функціональними можливостями. Саме тому необхідно обрати SDK, що дає можливість використовувати останні розробки та алгоритми при написанні додатку.

Набір для розробки програмного забезпечення Unity AR Foundation було обрано основним при вирішенні поставленої задачі, через те, що він має основні переваги таких SDK як Google ARCore та Apple ARKit. Мета обраного SDK – спростити створення додатку AR, що підтримується сучасними операційними системами, такими як Android та iOS.

У вікні створення нового проекту можна завантажити готовий шаблон для швидкого запуску додатків AR із зразком сцени, ресурсами, рекомендованими пакетами та налаштуваннями (рис. 2.2). Шаблон AR v2.0.1 має необхідний для розробки SDK AR Foundation 4.0.12.

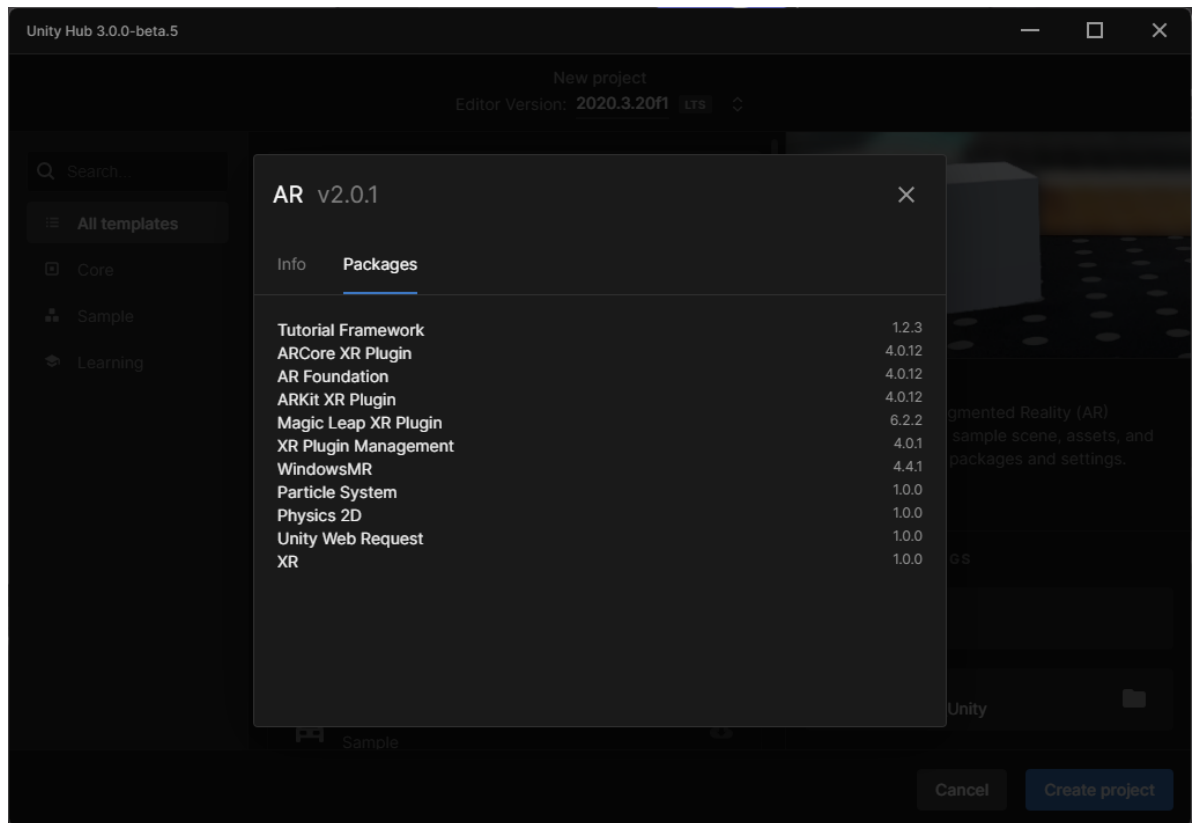


Рисунок 2.2 – Набір пакетів для шаблону AR v2.0.1

Основна задача додатку, що буде розроблятися – відображення у середовищі доповненої реальності 3D-моделі об’єктів збірки. Саме тому, значна кількість часу необхідна також і на моделювання.

Існують програмні забезпечення, що дають змогу виконати автоматично побудову 3D-моделі, як за допомогою камери смартфона, так і за допомогою

спеціального обладнання. Основними недоліками такого рішення є те, що без використання спеціального обладнання при створенні моделі вона має неточності, які можуть залежати від різних факторів. Створена 3D-модель також не повинна являтися суцільним об'єктом і мати можливість відображати як усі, так і окремі елементи.

У якості середовища для створення моделей об'єктів збірки було використано Autodesk Fusion 360 (рис. 2.3). Основними перевагами цього рішення є точність в процесі створення тривимірних об'єктів, що дозволяє якісно відтворювати моделі відповідно до оригінальних. При роботі з текстурами моделей можна використати вбудовані функції даного середовища, або продовжити роботу в середовищі Blender, що має більший набір плагінів, та дозволить більш детально опрацювати дизайн.

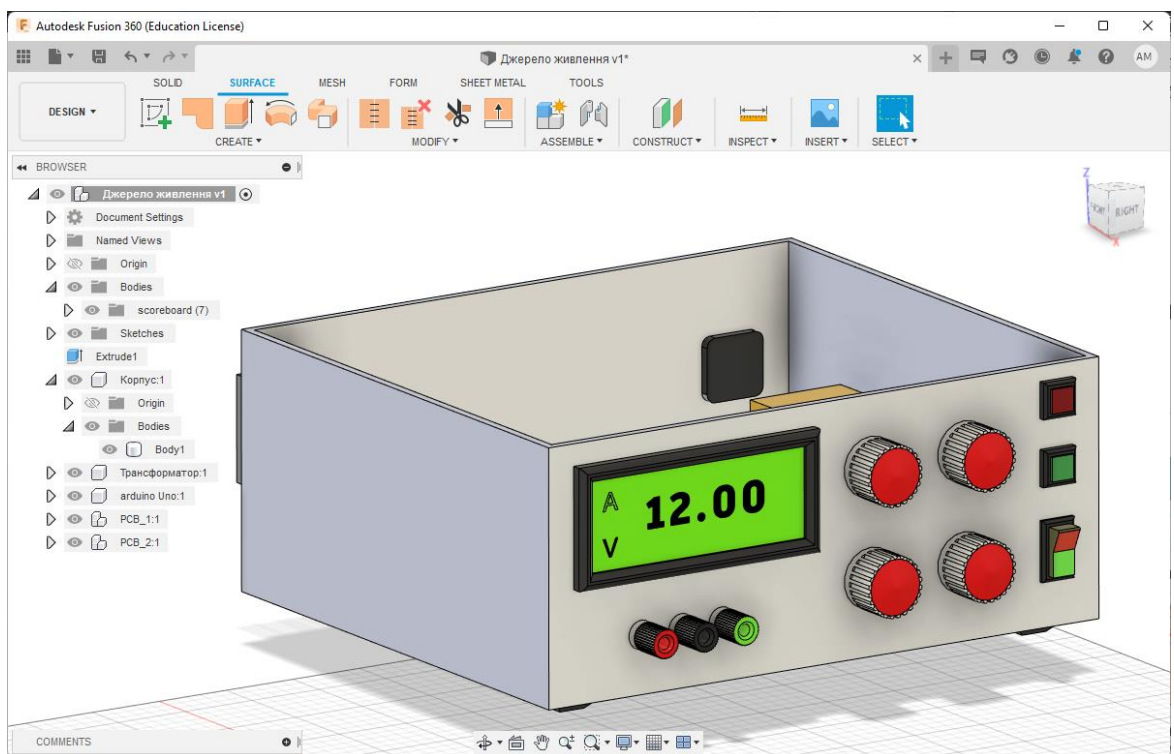


Рисунок 2.3 – Створення моделі об'єкту в середовищі Autodesk Fusion 360

Створивши модель об'єкту «Джерело живлення v1», яку будемо використовувати при розробці (рис. 2.4), експортуємо у форматі *.fbx та

підключаємо до проекту в Unity. Основні елементи, такі як AR Session Origin та AR Camera було автоматично додано при створенні проекту.

У використаному SDK розпізнавання міток виконується скриптом ARTrackedImageManager, до якого підключаємо бібліотеку зі стандартним ім'ям ReferenceImageLibrary, що містить зображення-маркери.

Залишається створити Prefab зі створеної раніше моделі об'єкту та підключити його до ARTrackedImageManager, що буде відображати його у разі розпізнавання мітки.

Prefab – особливий тип асетів, який дозволяє зберігати весь GameObject з усіма компонентами та значеннями властивостей. Виступає в ролі шаблону для створення екземплярів об'єкта, що зберігається в сцені [26].

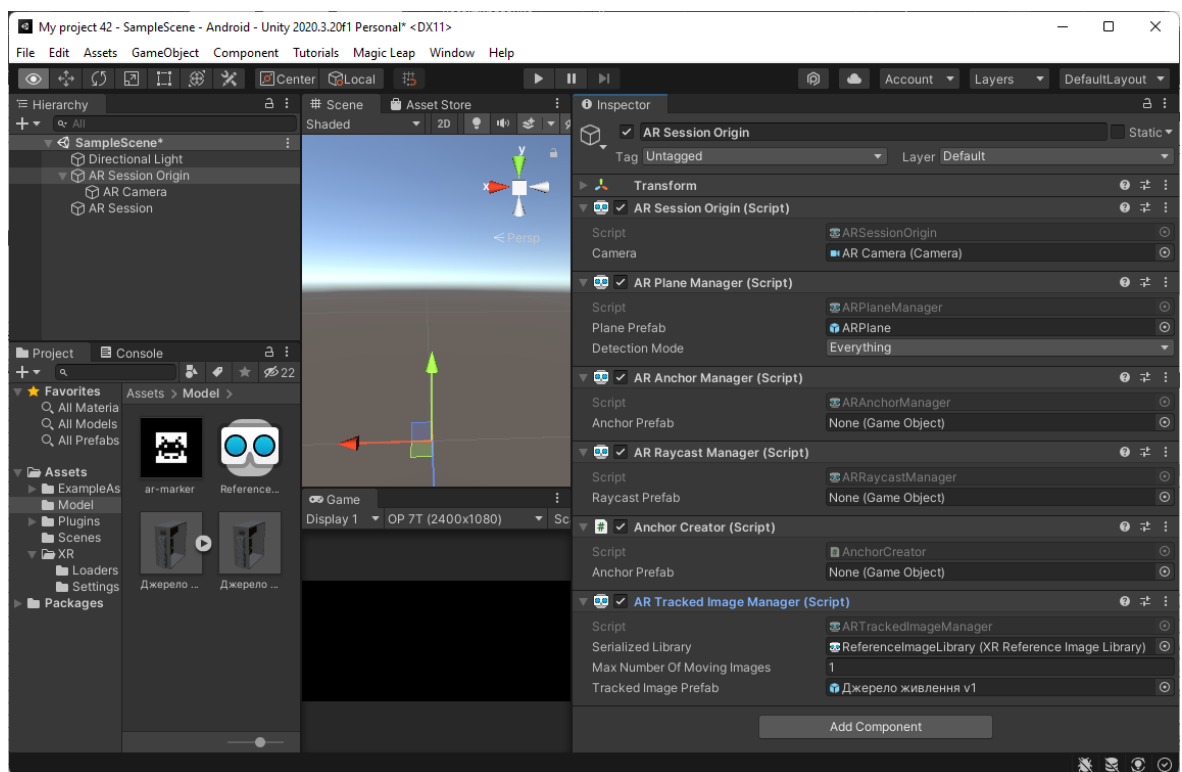
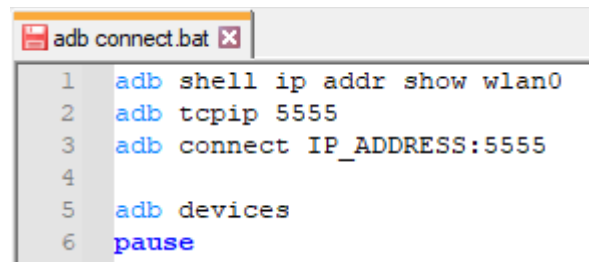


Рисунок 2.4 – Робоче середовище Unity при створенні додатку AR

Підключивши смартфон до ПК по кабелю з увімкненою ADB та натиснувши кнопку «Build and Run» створена програма автоматично буде встановлена та запущена. Використавши скрипт для ADB (рис. 2.5) можна

робити те саме без підключення по кабелю, при умові знаходження ПК та смартфона в одній Wi-Fi мережі.



```

1  adb shell ip addr show wlan0
2  adb tcpip 5555
3  adb connect IP_ADDRESS:5555
4
5  adb devices
6  pause
  
```

Рисунок 2.5 – Скрипт для підключення ADB в мережі Wi-Fi

Після запуску встановленого додатку знадобиться декілька секунд, щоб камера проаналізувала оточуючі поверхні. У разі знаходження заданої мітки, на екрані смартфона можна спостерігати об'єкт, що відображається в середовищі доповненої реальності (рис. 2.6). При переміщенні мітки об'єкт в середовищі доповненої реальності буде рухатися за нею, а при зміні кута повороту маркеру, відносно камери, також буде змінюватися і кут повороту відображуваної моделі. Це дає змогу оглянути об'єкт майже зі всіх сторін, і велику роль у зручності огляду буде мати розположення маркеру, яке можна змінити у будь-який момент.

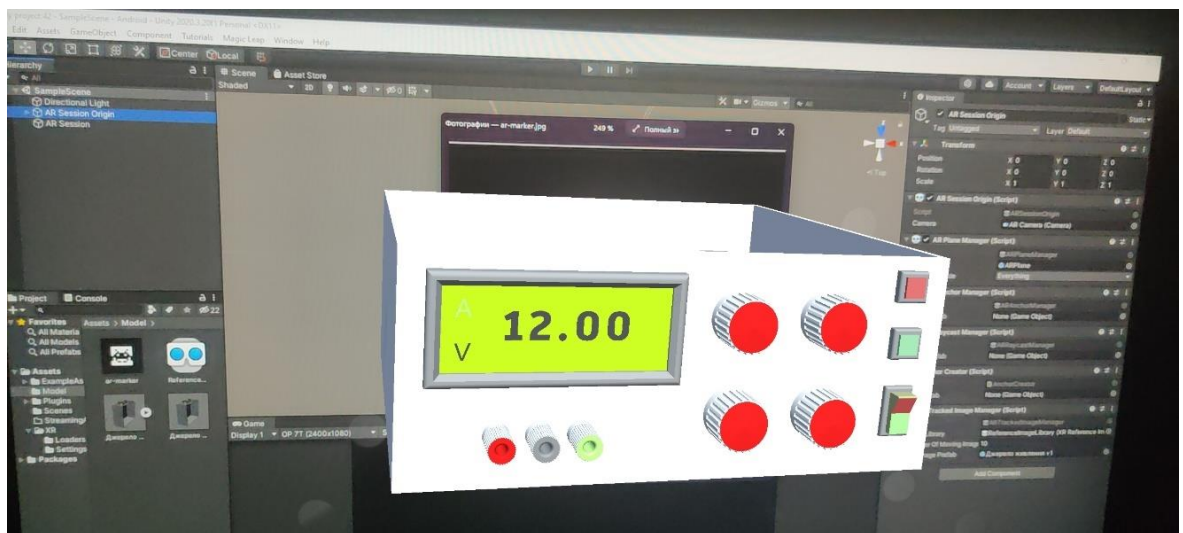


Рисунок 2.6 – Накладання об'єкту на мітку в середовищі AR

2.2 Формування технічного завдання до розроблюваного додатку

У попередньому підрозділі було проведено вибір засобів розробки та проведена підготовка перед повноцінною розробкою додатку. Проаналізувавши сучасні аналоги та спробувавши створити власне рішення було сформовано основні вимоги до функціональних можливостей розроблюваного ПЗ. Таким чином розроблюваний додаток повинен мати наступні функціональні можливості:

- інтуїтивно зрозумілий інтерфейс;
- вибір об'єкту для відображення;
- зображення та назва об'єкту у списку;
- вибір режиму відображення;
- відображати готовий об'єкт;
- перегляд об'єкту на будь-якому етапі збірки;
- покрокове відображення елементів об'єкту;
- виділення обраного елементу об'єкту;
- налаштування параметрів об'єкту;
- зміна прозорості об'єкту;
- збереження поточних параметрів об'єкту;
- завантаження збережених параметрів;
- обмін поточною конфігурацією з іншими пристроями.

2.3 Розробка структурної схеми програмного засобу

Розробка структурної схеми для програмного забезпечення є основою для розробки в цілому. Структурна схема додатку AR представлена на рис. 2.7.

Наведена структурна схема відображає роботу створеного додатку на пристрої, наприклад смартфоні. Блоком обробки програми виступає використаний при розробці SDK AR Foundation та написаний програмний код, що реалізує основні функціональні можливості та логіку роботи додатку.

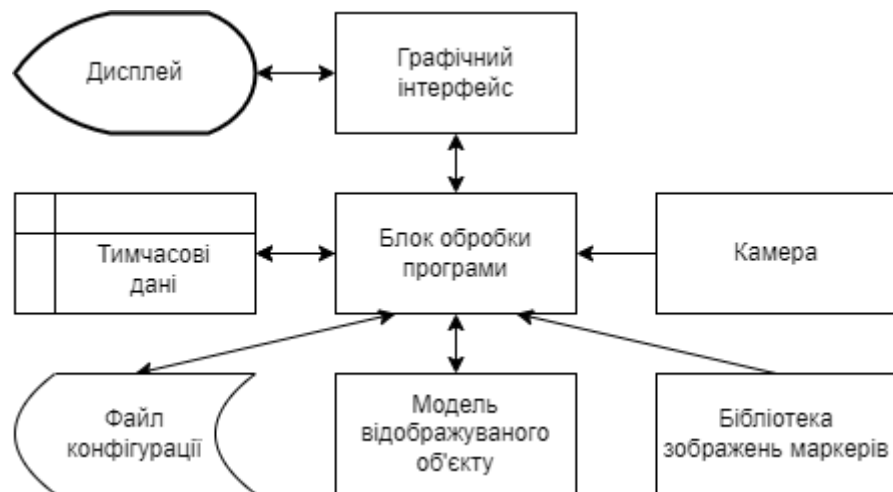


Рисунок 2.7 – Структурна схема роботи додатку AR

В ході роботи пристрій отримує інформацію з камери у вигляді відеопотоку та обробляє її у блоці обробки програми. Даний блок обробляє кожен кадр відеопотоку: виконує пошук заданого маркера заздалегідь, визначає положення маркера в просторі і на основі цих даних обчислює положення віртуальної камери щодо маркера. Після того, як положення та орієнтація камери визначені, блок обробки програми поміщає на сцену задану модель відображуваного об'єкту для візуалізації, використовуючи задані параметри положення, повороту та розміру. Далі відбувається візуалізація моделі на дисплеї пристрою завдяки її накладанню на отриманий відеопотік. При відображенні об'єкту він займає положення маркера або, при відповідному програмному коду, будь-яке встановлене розташування.

Також завдяки використанню дисплею та програмному коду можна створити графічний інтерфейс для взаємодії користувача з відображуваним об'єктом в середовищі доповненої реальності.

Робота пошуку маркерів закладається в тому, що блок обробки програми звертається до бібліотеки зображень маркерів, яка має заздалегідь встановлені зображення, та отримує необхідну інформацію для їх пошуку.

Використання файлу конфігурації дає можливість зберігати встановлені параметри для об'єкту після завершення програми та їх повторне використання при наступних запусках програми. Також таке рішення дозволяє робити резервне

копіювання параметрів налаштувань на пристрої та робить можливим обмін конфігурації між пристроями на яких використовується додаток.

Під час роботи додатку інформація з блоку обробки програми, така як змінні та тимчасові данні, знаходиться в оперативній пам'яті пристрою. Після завершення програми дані з оперативної пам'яті видаляються.

2.4 Розробка об'єктної моделі програмного засобу

Об'єктна модель – це логічний інтерфейс, програмне забезпечення або система, що моделюються за допомогою об'єктно-орієнтованих методів. Це дозволяє створювати архітектурне програмне забезпечення чи модель системи до розробки чи програмування. Об'єктна модель є частиною життєвого циклу об'єктно-орієнтованого програмування (ООП).

Для розробки додатку з AR необхідно спроектувати об'єктну модель ПЗ. Об'єктна модель програмного засобу представлена на рис. 2.8.

На об'єктній моделі зображено основну структуру програми, що складається з таких вузлів, як вузол блоку обробки програми, графічного інтерфейсу для взаємодії користувача з додатком, вибір об'єкту зі списку, отримання та відображення зображення з камери, вибір режиму відображення, робота з файлом конфігурації, зміна налаштувань та кнопка завершення збірки. Вихід з програми виконується закриттям додатку.

Створена об'єктна модель дозволяє краще оцінити основні задачі при подальшій розробці алгоритму ПЗ та має значну роль у вирішенні питання розробки алгоритму програми.

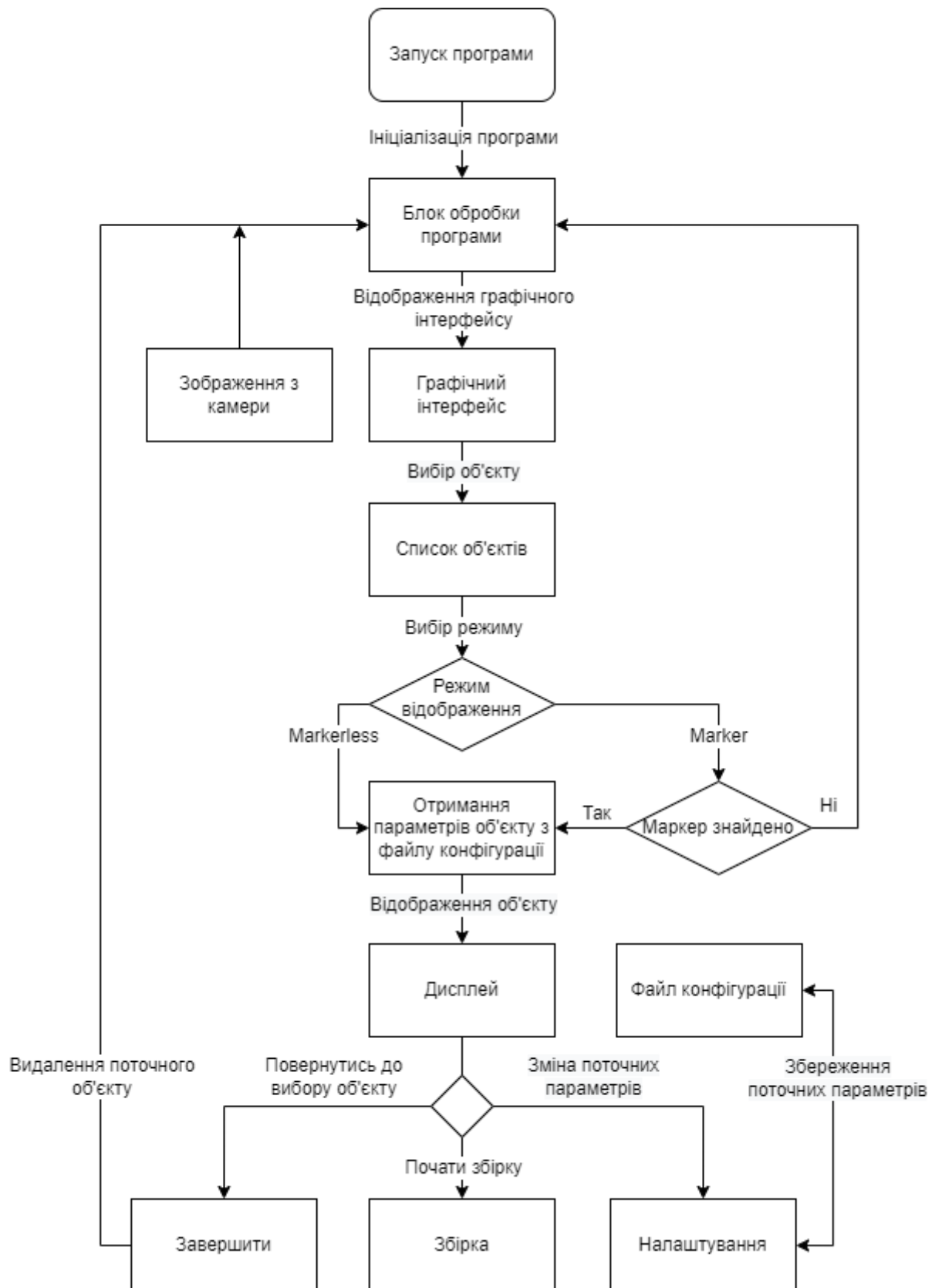


Рисунок 2.8 – Об'єктна модель програмного засобу

2.5 Розробка алгоритму програмного засобу

Складність розробки програмного забезпечення збільшується з кожним днем. Важливою частиною програмування є алгоритми та їх розуміння. Алгоритм програми – представляє собою покроковий опис операцій, які можуть бути виконані під час роботи ПЗ для отримання результату. Далі алгоритми кодуються на комп'ютері за допомогою обраної мови програмування.

Враховуючи можливості існуючих додатків з AR та сформовані основні вимоги до функціональних можливостей розроблюваного ПЗ було розроблено наступний алгоритм, що представлений на рис. 2.9–2.12.

На рис 2.9 зображено алгоритм роботи програми одразу після запуску у розділі «Головне меню», далі наводиться детальний опис цього алгоритму.

При запуску додатку відбувається перевірка на дозвіл роботи з камерою, якщо його не було надано, то буде відсутнє зображення з камери і програма не зможе повноцінно функціонувати, тому потрібно надати відповідний дозвіл. При успішній перевірці дозволу додаток звертається до основної камери пристрою, ініціює її, та починає приймати відеопотік з камери і відображати його на дисплей пристрою. Вивід зображення з камери відбувається на протязі усієї роботи додатку і припиняється по завершенню його роботи.

Після того як додаток було запущено і зображення з камери з'явилося на дисплеї відображається кнопка «Select object», що слугує для виклику списку збережених моделей об'єктів. Коли користувач натисне на цю кнопку відкривається відповідний список, що має зображення об'єктів та їх назву. Переміщення по списку інтуїтивно зрозуміле для користувача, та може відбуватися двома способами завдяки прокручуванню списку після натискання і утримування пальцю на:

- зображенні об'єкту;
- повзункові прокручування.

Користувач має змогу вибрати режим відображення об'єкту у середовищі доповненої реальності, для цього існує кнопка-перемикач «Marker / Markerless»,

що змінює своє зображення відповідно до обраного режиму. Зображення та назва режиму є інтуїтивно зрозумілими та у повній мірі характеризують обраний режим відображення об'єкту. При запуску додатку автоматично обирається перший режим з відображенням об'єкту на маркері, а при зміні на другий режим відображення об'єкту починається одразу після його вибору.

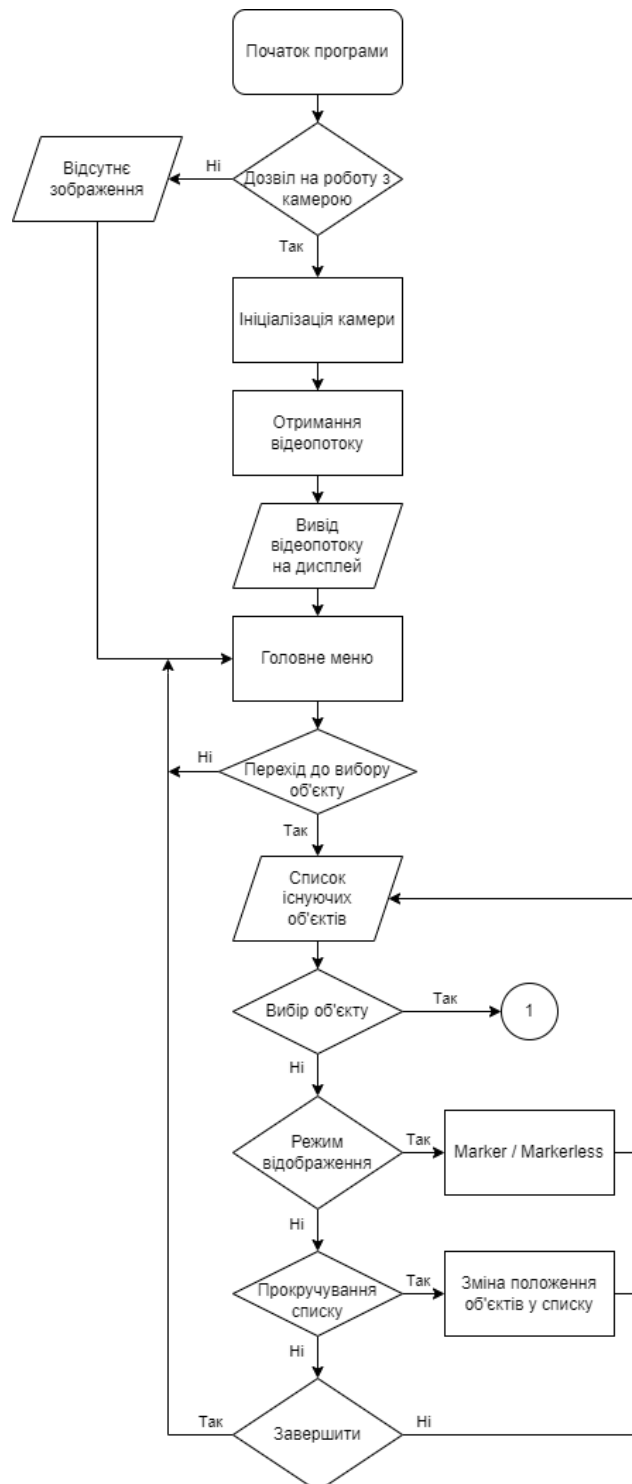


Рисунок 2.9 – Алгоритм роботи програмного засобу «Головне меню»

Кнопка завершити повертає користувача назад до головного меню та закриває відкритий список об'єктів.

На рис. 2.10 зображено алгоритм роботи програми після вибору об'єкту зі списку у розділі «Головне меню», далі наводиться детальний опис цього алгоритму.

Додаток за допомогою використання допоміжного скрипту присвоює об'єкту тег, а далі передає ім'я обраного об'єкту до головної частини програми.

Виконується перевірка встановленого режиму. Якщо було обрано відображення за маркером, то виконується увімкнення компоненту AR Tracked Image Manager, що являє собою скрипт SDK AR Foundation до якого підключено створену у проекті бібліотеку зображень маркерів. Даний компонент виконує пошук маркерів підключеної бібліотеки на отриманому зображенні з камери.

Також, після запуску програми і увімкнення камери, компонент AR Session Origin, що являється частиною обраного SDK, виділяє з отриманого від камери зображення площину та опорні точки у просторі для коректної оцінки положення пристрою користувача та розташування об'єктів у середовищі доповненої реальності.

Отримуючи зображення з камери додаток шукає на ньому відповідні маркери, що має підключена бібліотека. Цей процес відбувається безперервно до виходу в «Головне меню» (компонент деактивується) або завершення програми. При успішному знаходженні маркеру додаток обробляє його розташування у просторі відносно до камери користувача. Після того, як виконується позиціонування маркеру, на його місці з'являється обраний об'єкт. При переміщенні маркеру або камери дані о положенні маркеру знову оновлюються та корегується положення об'єкту.

Якщо було обрано відображення без маркеру, то компонент для пошуку маркеру не активується, а обраний об'єкт одразу відображається у просторі доповненої реальності.

Далі виконується зміна ім'я створеного об'єкту на «Model0» та виклик методу `Preset_manager()` головної частини програми. Для його роботи

необхідний дозвіл на доступ до файлів пристрою. Без надання доступу додаток не буде функціонувати у повній мірі.

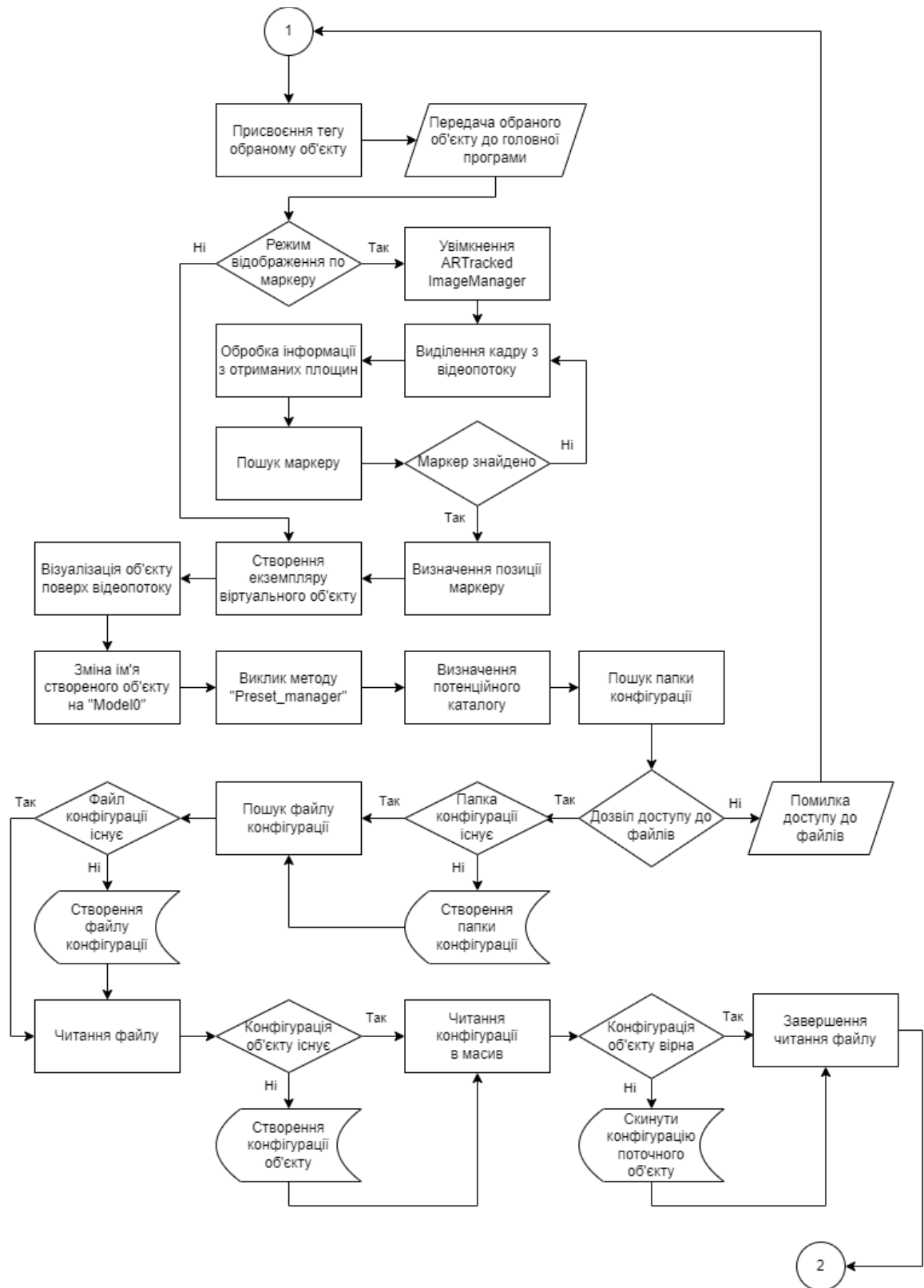


Рисунок 2.10 – Алгоритм роботи програмного засобу при відображенні обраного об'єкту

Основна задача методу `Preset_manager()` полягає у визначенні потенційного каталогу на пристрої, пошуку папки конфігурації або її створенні при відсутності, пошуку файлу конфігурації або його створенні при відсутності, читанні файлу, перевірці обраної конфігурації поточного об'єкту або її створенні при відсутності, читанні конфігурації в масив, перевірці вірності отриманої конфігурації або її скиданні при невірних даних та завершення читання файлу конфігурації.

На рис. 2.11 зображено алгоритм роботи програми після виконання читання конфігураційного файлу, далі наводиться детальний опис цього алгоритму.

Отримавши конфігураційні дані відбувається їх занесення до змінних та початкове позиціонування об'єкту. Далі кнопки з попереднього інтерфейсу, окрім кнопки «Cancel», деактивуються і активуються кнопки «Settings» та «Assembly», а значення слайдерів «X», «Y» та «Z» змінюються згідно до отриманих конфігураційних даних поточного об'єкту.

Розташування об'єкту у просторі доповненої реальності залежить від встановленого користувачем режиму. При виборі відображення об'єкту по маркеру виконується визначення позиції маркеру, згідно з попереднім алгоритмом, далі до отриманих даних о положенні маркеру додаються значення зі слайдерів. Тоді як при відображенні без маркеру на розташування об'єкту у просторі доповненої реальності впливають початкові дані розміщення пристрою після запуску додатку та значення слайдерів. При кожному запуску додатку початкові дані розміщення пристрою визначаються його положенням у реальному світі.

Коли обраний об'єкт буде розташовано відбувається пошук його дочірніх об'єктів з компонентом `Renderer` та їх запис до масиву. До основних дочірніх об'єктів, що не мають компоненту `VoxCollider` додається відповідний компонент.

Далі користувач може почати збірку об'єкту, перейти до налаштувань або повернутися до головного меню.

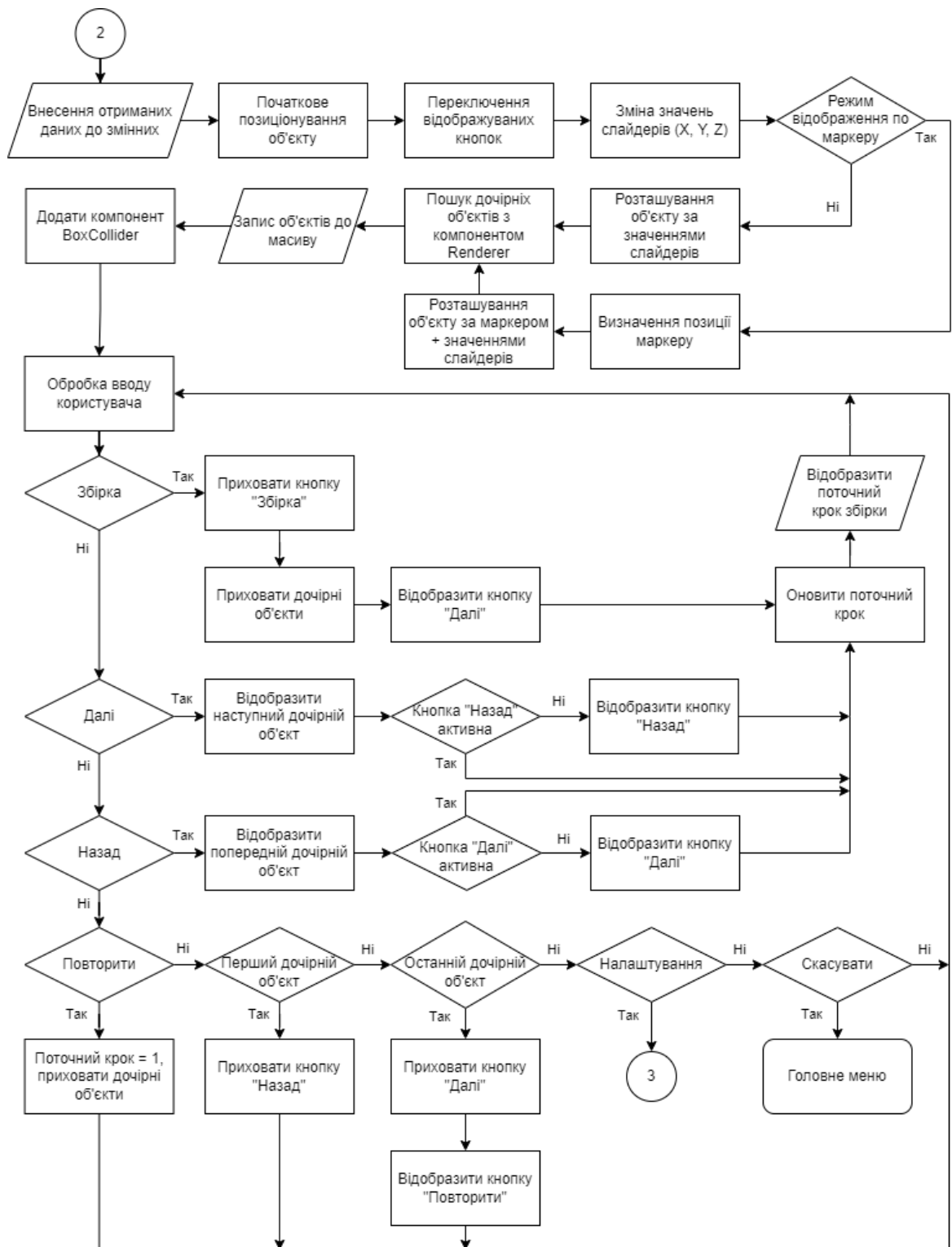


Рисунок 2.11 – Алгоритм роботи програмного засобу при збірці об'єкту

Розглянемо вибір збірки об'єкту після натискання користувачем на кнопку «Assembly». Відповідна кнопка деактивується та миттєво виконується приховання дочірніх об'єктів обраного об'єкту за рахунок вимкнення їх компоненту Renderer. Активується кнопка «Next», що слугує для відображення

наступного дочірнього об'єкту, оновлюється та відображається поточний крок збірки.

Кожне натискання кнопки «Next» буде відображати наступний дочірній об'єкт, а при переході до другого кроку активує кнопку «Back», також відбувається і оновлення та відображення поточного кроку.

Натискання на кнопку «Back» буде приховувати дочірній об'єкт поточного кроку, змінювати поточний крок на -1, оновлювати та відображати поточний крок. При неактивній кнопці «Next» – активує її.

У процесі збірки об'єкту, при досягненні першого кроку – кнопка «Back» буде дезактивована, а при останньому – дезактивована кнопка «Next» та активована кнопка «Repeat».

Якщо поточний крок дорівнює кількості відображених дочірніх об'єктів – буде активовано кнопку «Repeat», що дозволяє розпочати поточну збірку з першого кроку, поточний крок збірки дорівнює 1 а дочірні об'єкти приховуються.

На рис. 2.12 зображено алгоритм роботи при переході до розділу налаштувань та взаємодії користувача з цим розділом, далі наводиться детальний опис цього алгоритму.

Розглянемо перехід до налаштувань після натискання користувачем на кнопку «Settings». Відповідна кнопка має два зображення для інтуїтивного розуміння активності розділу налаштувань, відображення / приховання розділу виконується повторним натисканням кнопки.

Після натискання кнопки, якщо розділ налаштувань не відображався, активуються слайдери та пункти меню налаштувань і кнопка «Presets», дезактивується кнопка «Cancel», замість поточного кроку збірки відображається поточний пресет та пункт налаштувань. У разі вже активного розділу налаштувань, повторне натискання даної кнопки деактивує слайдери, пункти меню налаштувань і кнопку «Presets», активує кнопку «Cancel» та замінює відображення поточного пункту налаштувань на відображення поточного кроку збірки.



Рисунок 2.12 – Алгоритм роботи програмного засобу

Зміна значень слайдерів оновлює поточні дані, що відображаються та змінює розташування об'єкту в середовищі доповненої реальності.

Перехід між пунктами налаштувань виконується натисканням відповідних кнопок:

- «Position»;
- «Rotation»;
- «Scale»;
- «Visibility».

Зображення для зазначених кнопок створені відповідно до налаштувань, що вони виконують. Так «Position» відповідає за розташування об'єкту у середовищі AR, «Rotation» – за кут повороту стосовно простору, «Scale» – за розмір відображуваного об'єкту, «Visibility» – за коефіцієнт прозорості (за замовчуванням об'єкт повністю не прозорий). Поточний пункт налаштувань також відображається при активному розділі налаштувань.

При активному пункті «Rotation» повторне натискання на даний пункт вимикає режим «Rotation Lock», а якщо режим було вимкнено – вмикає. У даному режимі відображуваний об'єкт припиняє змінювати свій кут повороту в залежності від маркеру та приймає значення від слайдерів, інакше – об'єкт змінює кут повороту в залежності від маркеру.

Натискання кнопки «Presets» відображає неактивні пункти вибору пресетів та кнопку «Save» для їх збереження або дезактивує їх, якщо вони були активні.

При виборі користувачем кнопки «1», «2» або «3» відбувається повторний виклик методу `Preset_manager` з запитом на оновлення даних відповідно до обраного пресету, що був детально описаний у другій частині алгоритму для рис. 2.10. Отримані дані одразу застосовуються до відображуваного об'єкту, а значення слайдерів скидаються до значень пресету.

2.6. Висновки до другого розділу

Вибір інструментів для розробки програмного забезпечення є одним із основних етапів на шляху створення якісних сучасних рішень. В залежності від обраного середовища розробки може відрізнятися мова програмування, складність та методи створення ПЗ, що будуть впливати як на швидкість розробки, так і на якість готового продукту. Те саме відноситься і до вибору середовища графічного моделювання та SDK. Обрані засоби розробки у повній мірі відповідають моїм особистим вимогам, а розробка з їх використанням є зручною і зрозумілою.

Формування технічного завдання необхідне для вирішення такого питання, як визначення функціональних можливостей ПЗ, що у повній мірі будуть відповідати потребам користувача. Саме тому було сформовано основні вимоги щодо функціоналу розроблюваного додатку.

Розроблена структурна схема програмного засобу дозволяє зрозуміти елементарний принцип його роботи, та краще орієнтуватися в наступному кроці розробки.

Розробку об'єктної моделі додатку було виконано перед написанням алгоритму, для того щоб детально оцінити зв'язки між основними вузлами програми. Це дозволить краще зрозуміти дії користувача та зобразити на алгоритмі необхідну логіку роботи програми.

Розробка алгоритму програмного засобу являє собою досить важливу частину поставленого завдання, так як у повній мірі відображує всі можливі дії користувача у ПЗ та значно спрощує реалізацію задуманої логіки при написанні коду. Завдяки чому менша вірогідність отримання критичної помилки при роботі готового додатку, так як розробка виконується згідно створеного алгоритму.

3 РОЗРОБКА ТА ТЕСТУВАННЯ ДОДАТКУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

3.1 Розробка інтерфейсу програмного засобу

Важливою частиною в розробці програмного забезпечення є концепція та створення інтерфейсу, що являє собою зв'язок між користувачем та ПЗ. Користувач може взаємодіяти з інтерфейсом, щоб використовувати програмне забезпечення. Сьогодні інтерфейс користувача можна знайти майже скрізь, де є цифрові технології, безпосередньо на комп'ютерах, мобільних телефонах, автомобілях, музичних плеєрах, тощо.

Інтерфейс користувача є частиною програмного забезпечення і розробляється таким чином, щоб забезпечити розуміння користувачем програмного забезпечення. Інтерфейс користувача забезпечує фундаментальну платформу для взаємодії людини з комп'ютером.

Серед основних вимог при розробці інтерфейсу програмного забезпечення можна виділити наступні:

- інтуїтивно зрозумілий;
- не перевантажений;
- швидко реагує на команди користувача;
- відповідає усім функціональним можливостям даного ПЗ;
- адаптований до різних розмірів дисплею.

Враховуючи поставлені вимоги до функціональних можливостей, інтерфейсу та створений алгоритм розроблюваного додатку було визначено основні аспекти та цілі при розробці графічного інтерфейсу.

Розробка додатку велася у середовищі Unity, що має власний вбудований конструктор графічного інтерфейсу (рис. 3.1). В свою чергу це позитивно впливає як на розробку прототипу, так і на розробку фінальної версії інтерфейсу програмного забезпечення.

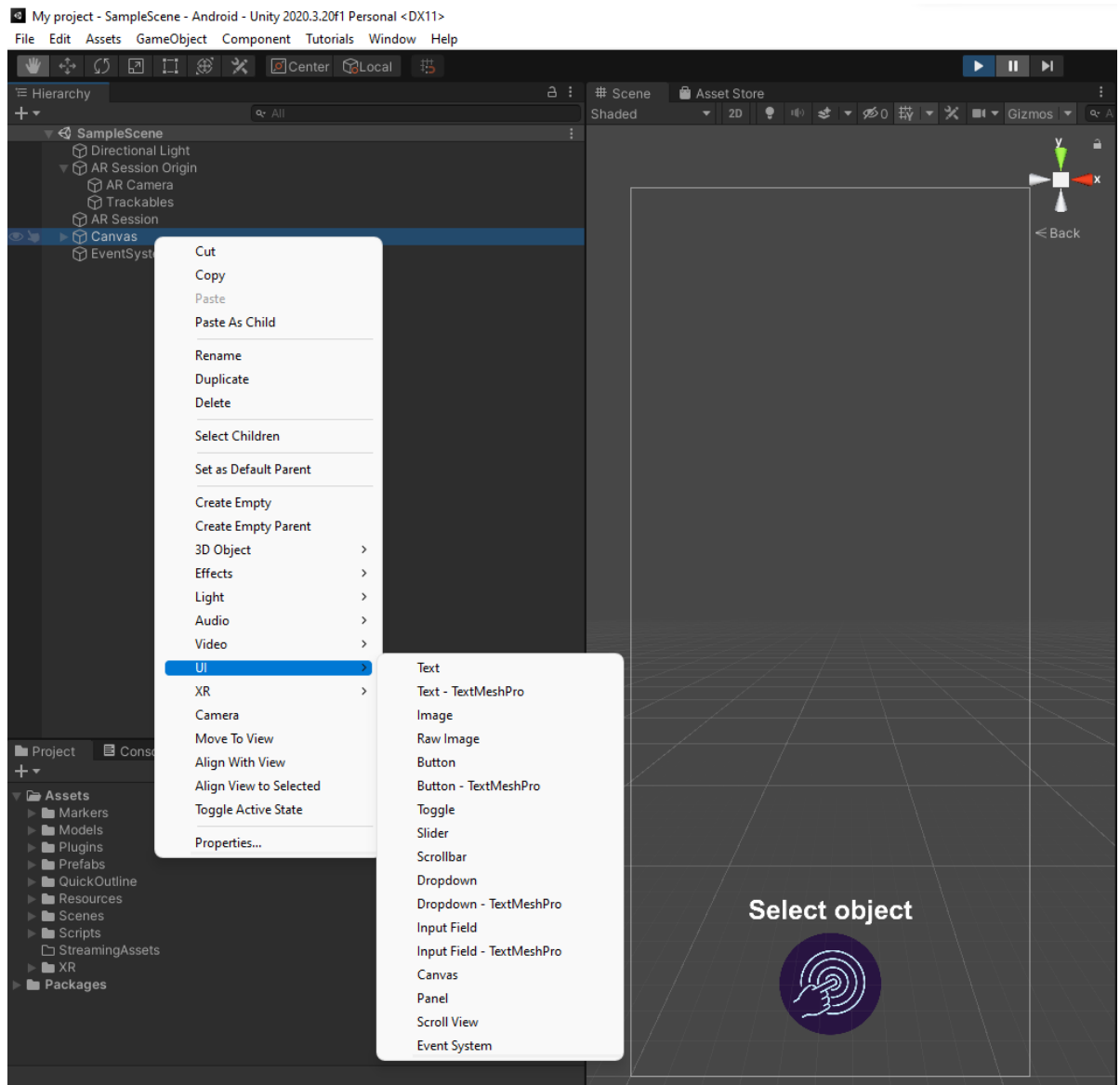


Рисунок 3.1 – Створення графічного інтерфейсу в Unity

Для створення графічних елементів інтерфейсу, таких як зображення для кнопок, було використано Adobe Photoshop (рис. 3.2).

Виконавши розробку алгоритму програмного забезпечення встановлено необхідну кількість іконок, що потрібно створити для повноцінної реалізації запланованого інтерфейсу. Загалом створено 17 іконок (рис. 3.3).

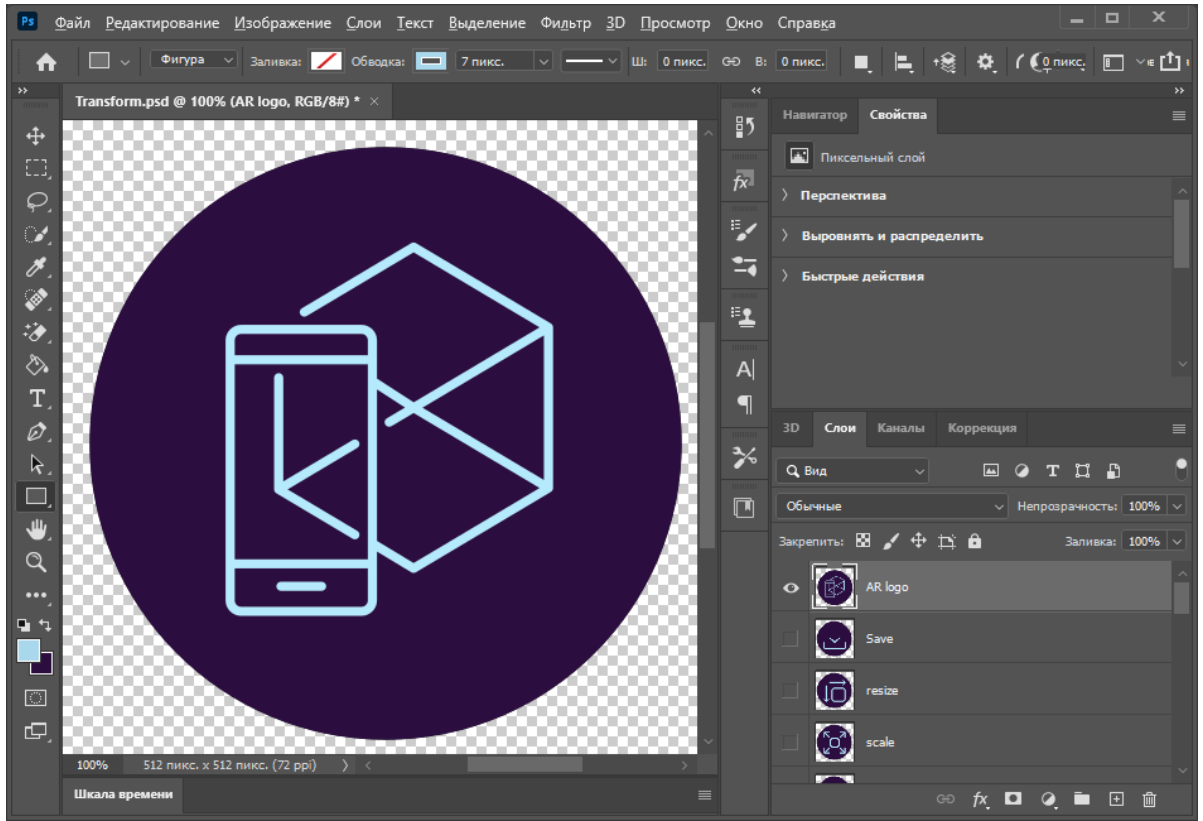


Рисунок 3.2 – Створення іконок для кнопок в Adobe Photoshop

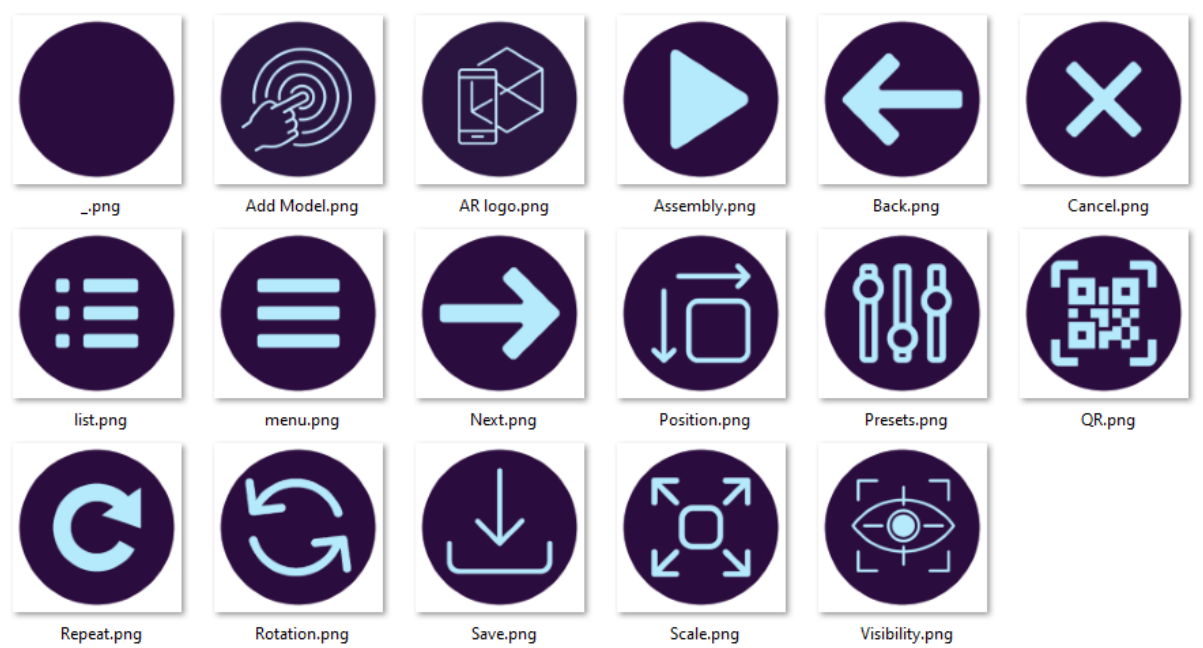


Рисунок 3.3 – Усі створені іконки для реалізації графічного інтерфейсу

Окрім створених іконок для графічного інтерфейсу необхідно також створити зображення для списку об’єктів, що буде реалізовано в додатку. Для

цього було виконано моделювання об'єктів в середовищі Autodesk Fusion 360 використавши знання отримані під час проходження професійної практики.

Далі проведено детальну візуалізацію об'єкту (рис. 3.4) та його збереження до файлу з розширенням *.jpg. Візуалізація виконується завдяки CPU ПК та займає певний час, від 30 секунд до 3 хвилин, в залежності від складності об'єкту та потужності процесору. Загалом було створено 6 об'єктів (рис. 3.5).

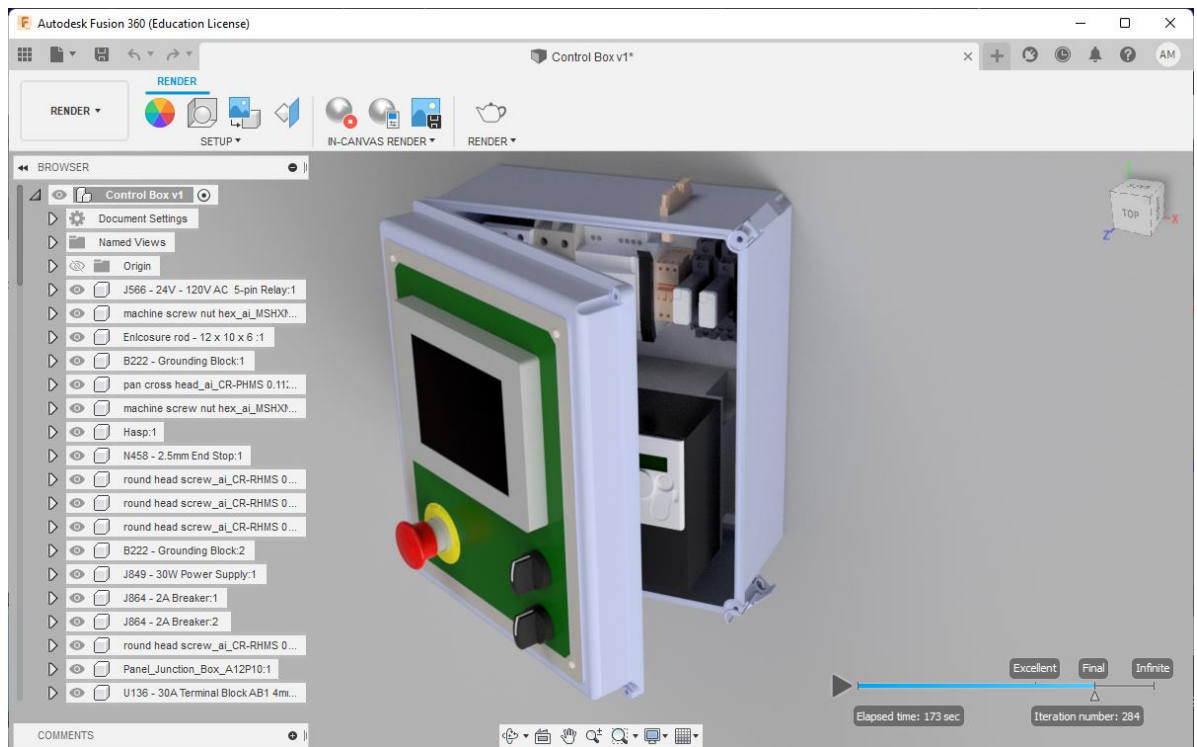


Рисунок 3.4 – Autodesk Fusion 360. Вікно візуалізації об'єкту

Завершивши підготовку основних елементів, необхідних для створення графічного інтерфейсу, їх було додано до проекту в Unity. Створені моделі було занесено до папки «Assets/Models», іконки – до папки «Assets/Resources/Icons», а зображення об'єктів – до папки «Assets/Resources/Img» (рис. 3.6).

При створенні даного додатку AR розробка інтерфейсу та коду програми виконувалась паралельно. Тобто по мірі доповнення алгоритму та написанні коду програми вносилися необхідні зміни до інтерфейсу.



Рисунок 3.5 – Зображення змодельованих об’єктів для проекту

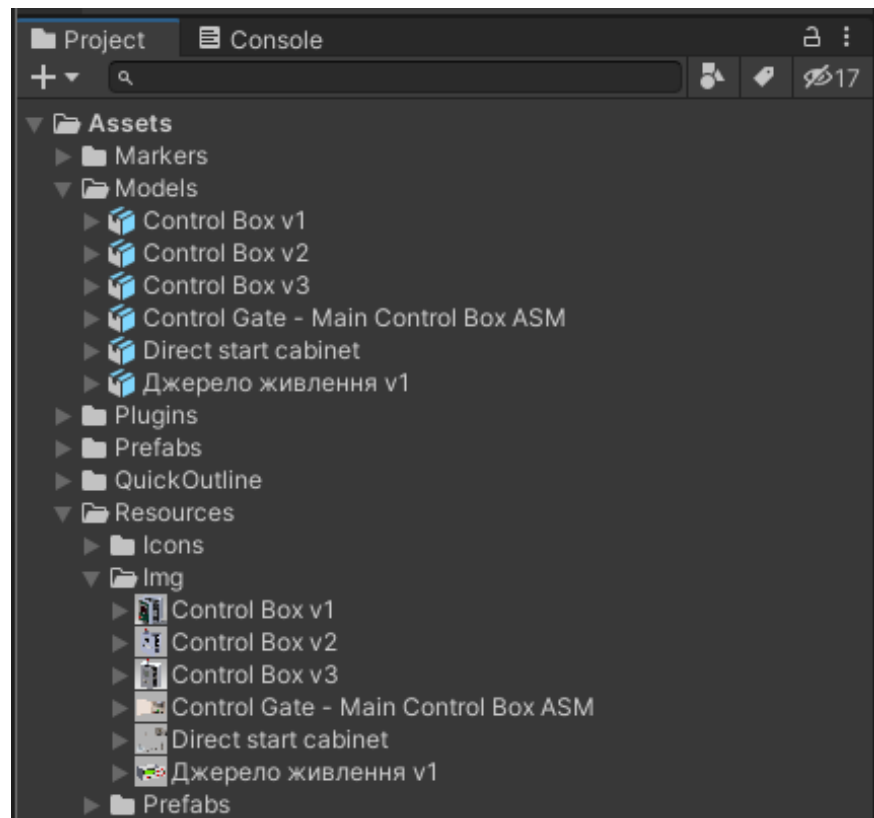


Рисунок 3.6 – Створенні об’єкти та їх зображення у середовищі проекту

Розробка інтерфейсу полягала у створенні в основній сцені проекту «SampleScene» такого об'єкту як «Canvas», що слугує середовищем для розміщення графічного інтерфейсу. Значення висоти і ширини було обрано відповідно до роздільної здатності пристрою на якому проводилося тестування програми на протязі усього життєвого циклу розробки, та становить 2400 та 1080 пікселів відповідно.

Перед використанням зображень, підготованих до використання на кнопках та списку моделей, необхідно налаштувати їх в інспекторі. Для цього, виділивши усі додані зображення, було змінено тип текстури на «Sprite (2D and UI)» (рис. 3.7).

Спрайт – це двовірний графічний об'єкт, що використовується для персонажів, реквізиту, та інших елементів двовимірного процесу Unity. Графіка отримана з растрових зображень – Texture2D. Клас Sprite в першу чергу визначає частину зображення, яка має використовуватись для певного спрайту. Потім ця інформація може використовуватися компонентом SpriteRenderer GameObject для фактичного відображення графіки [27].

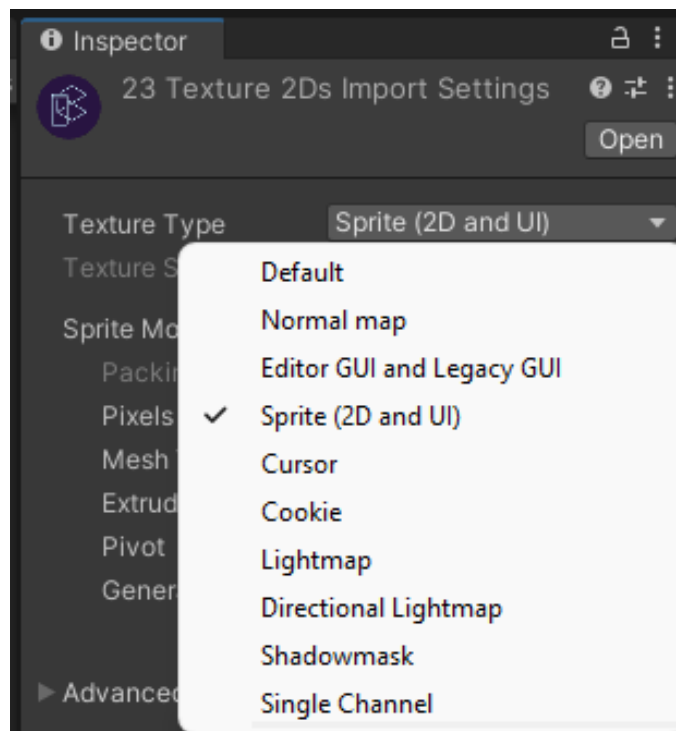


Рисунок 3.7 – Зміна типу текстури зображення на UI

Створивши необхідну кількість кнопок, слайдерів, а також об'єкту списку, їх було розміщено таким чином, щоб розроблений інтерфейс відповідав основним вимогам, що були висунуті при розробці інтерфейсу ПЗ (рис. 3.8). Важливо пам'ятати, що елементи інтерфейсу не будуть відображатися одночасно, а будуть залежати від дій користувача, тобто можуть активуватися та дезактивуватися в залежності від написаного коду програми.

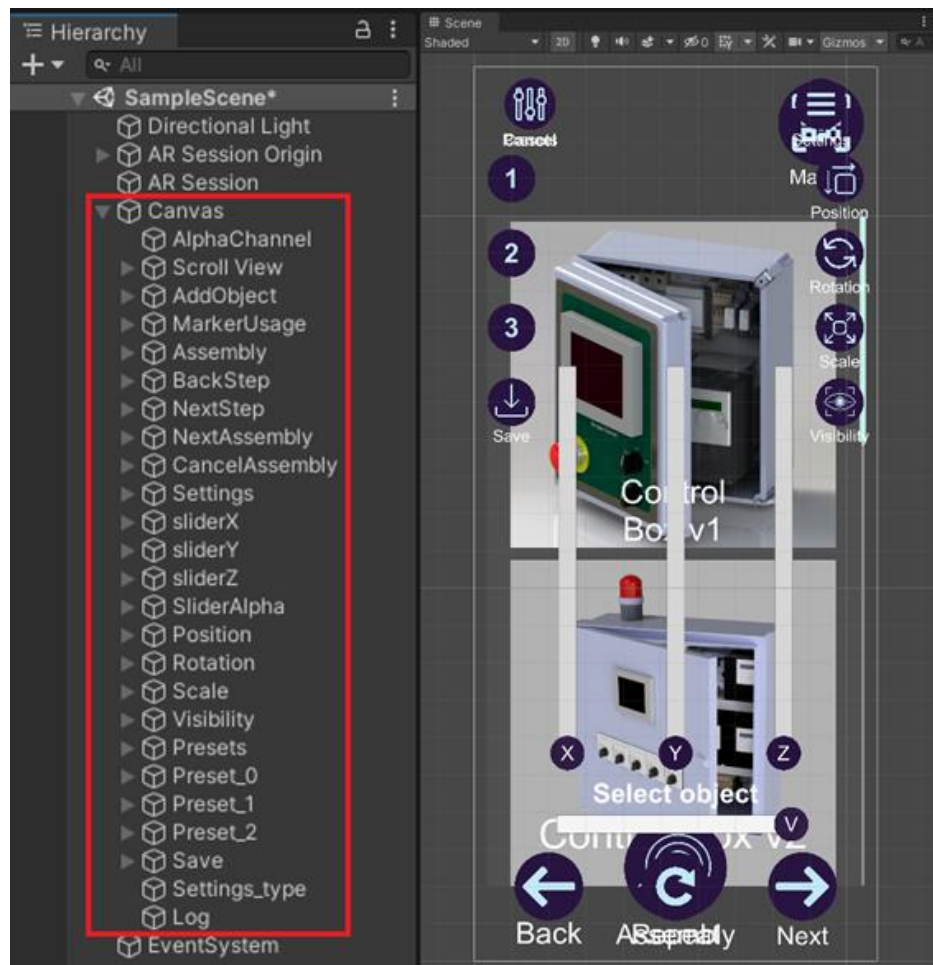


Рисунок 3.8 – Загальний вигляд створених об'єктів кнопок, слайдерів та списку у вікні ієрархії проекту та у вікні сцени

Завершений вигляд інтерфейсу представлено на рис. 3.9-3.10. Розроблений інтерфейс, на мою думку, повністю задовольняє вимогам сформованим у даному підрозділі. Поліпшити створений інтерфейс можна, наприклад додавши анімації при натисканні кнопок та переходу до розділу налаштувань, а також можна додати підтримку декількох мов інтерфейсу.



Рисунок 3.9 – Інтерфейс додатку при виборі об'єкту та його відображенні

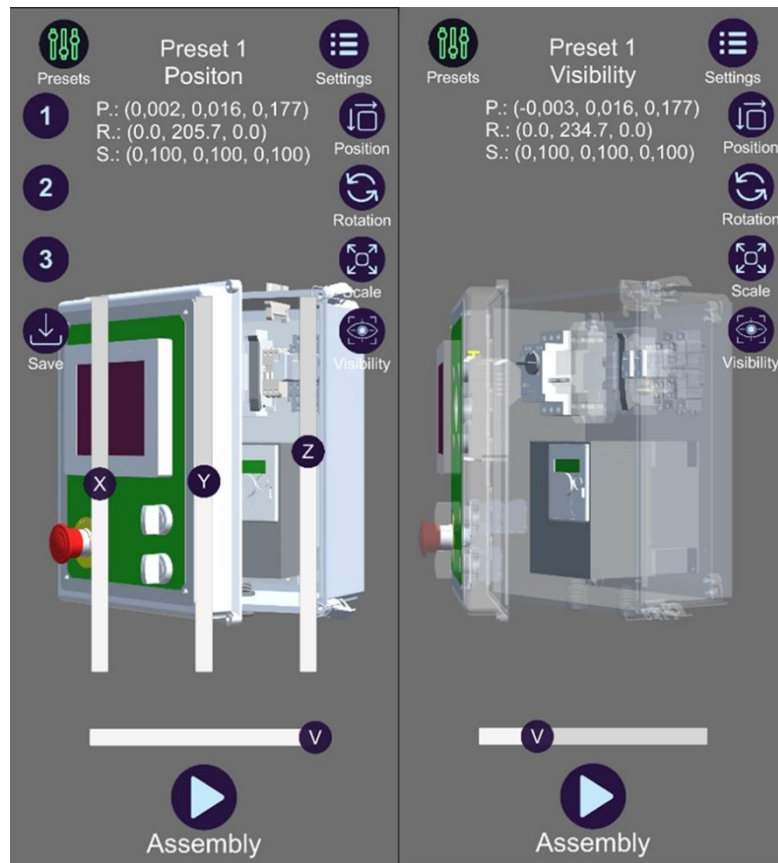


Рисунок 3.10 – Інтерфейс додатку у розділі налаштувань

3.2 Розробка програмного коду додатку

Програмний код додатку написаний мовою C#. До створеного проекту Unity підключено SDK AR Foundation для розробки додатку з використанням доповненої реальності.

Основну частину роботи програми по відстеженню маркерів реалізовано у ARSessionOrigin з підключеного SDK. ARSessionOrigin є батьківським об'єктом сцени для налаштування AR. Він містить камеру та будь-які ігрові об'єкти, створені з виявлених об'єктів, таких як площини чи хмари точок [28].

Коли Unity створює програму, він автоматично генерує файли маніфесту Android, виконавши наступні дії [29]:

- Unity використовує LibraryManifest.xml або AndroidManifest.xml як основний маніфест;
- потім він знаходить усі маніфести Android доданих плагінів (файли .aar та бібліотеки Android);
- він використовує клас `manifmerger` від Google для злиття маніфестів плагінів у основний маніфест;
- Unity змінює файли маніфесту в модулях запуску та бібліотеки та автоматично додає дозволи, параметри конфігурації, використовувані функції та іншу інформацію.

Для того щоб створений додаток мав можливість запиту до зовнішнього сховища файлів пристрою – до проекту було додано файл «AndroidManifest.xml», так як Unity автоматично не створює необхідні дозволи. Зміст файлу зображено на рис. 3.11.

Код програми реалізовано у вигляді трьох файлів скриптів *.cs (рис. 3.12), перший необхідно підключити до кожної кнопки списку моделей, а інші два підключені як компоненти до ARSessionOrigin.

```

<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.unity3d.player"
xmlns:tools="http://schemas.android.com/tools">
<application android:requestLegacyExternalStorage="true">
    <activity android:name="com.unity3d.player.UnityPlayerActivity"
        android:theme="@style/UnityThemeSelector">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
    </activity>
</application>
</manifest>

```

Рисунок 3.11 – Модифікована частина AndroidManifest.xml для надання можливості запиту дозволу на доступ до файлів

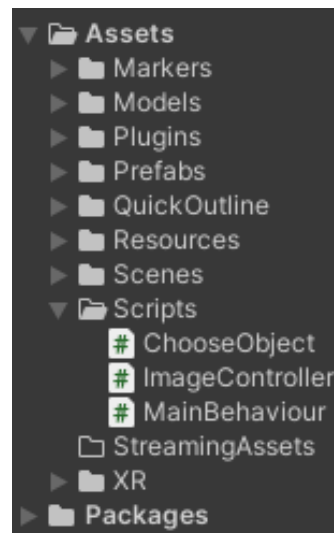


Рисунок 3.12 – Написані скрипти в ієрархії проекту

У коді програми такі класи, як кнопки, слайдери та об'єкти були створені наступним чином (рис. 3.13). Для їх роботи необхідно також підключити до них створені об'єкти – в основному це виконано у тілі методу Start(), приклад наведено на рис. 3.14, що викликається перед першим викликом будь-якого методу Update().

```

public Button AddObject_btn;
public Slider sliderX;
public GameObject ScrollView;

```

Рисунок 3.13 – Створення кнопки, слайдеру та об'єкту у коді ПЗ

```

AddObject_btn = GameObject.Find("AddObject").GetComponent<Button>();
AddObject_active = GameObject.Find("AddObject");
AddObject_btn.onClick.AddListener(AddObject_task);

sliderX = GameObject.Find("sliderX").GetComponent<Slider>();
sliderX_active = GameObject.Find("sliderX"); sliderX_active.SetActive(false);

ScrollView = GameObject.Find("Scroll View"); ScrollView.SetActive(false);

```

Рисунок 3.14 – Пошук об’єктів, компонентів, дезактивація, та обробка натискання кнопок

Завдяки функції `GameObject.Find()` знаходимо об’єкти в інтерфейсі програми, що мають відповідні імена. Для кнопок необхідно знайти об’єкт та компонент. Об’єкти у ході роботи програми будуть активуватися та дезактивуватися завдяки публічному методу `SetActive()` [30], а компоненти кнопок будуть використані при виклику функції, що викликає заданий метод при натисканні на кнопку [31]. Таким чином для кожної кнопки було створено окремий метод, частину створеної структури методів зображено на рис. 3.15.

```

573  +  ссылка: 8  void Settings_title()...
580
581  +  ссылка: 1  void Save_task()...
633
634  +  ссылка: 4  void Apply_Preset()...
661
662  +  ссылка: 4  void Preset_manager()...
777  }

```

Рисунок 3.15 – Частина структури методів головної програми

Після створення та підключення об’єкту «`ScrollView`», його було налаштовано відповідно до розміру екрану пристрою так, щоб використання списку було комфортним при користуванні обома чи однією рукою. Тобто в основному список був налаштований так, щоб великим пальцем руки, у якій лежить пристрій, якщо це смартфон, міг дотягнутися до верхнього елементу списку. Далі імпортовані до проекту моделі об’єктів було збережено як префаби

до розділу «Assets/Prefabs», та додано до кожної кнопки списку. Відповідно до обраного об'єкту також внесена його назва та додано його зображення.

Виходячи з цього, щоб додати у розроблену програму нову модель об'єкту для збірки, необхідно виконати її моделювання та зберегти у форматі *.fbx, імпортувати файл до проекту, зберегти його префаб, копіювати останню кнопку зі списку та змінити у ній префаб на новий, при необхідності відповідно змінивши зображення та текст кнопки.

Відповідно до розробленого алгоритму налаштовано відображення елементів інтерфейсу, для його відображення та приховування в залежності від дій користувача. Тобто після запуску програми користувач бачить тільки кнопку виклику списку, а в розділі налаштувань усі основні елементи розробленого інтерфейсу.

Після вибору користувачем об'єкту зі списку скрипт «ChooseObject» передає цю інформацію до головного скрипту «MainBehaviour» а також допоміжного «ImageController». Робота зазначених скриптів полягає в наступному (рис. 3.16):

- додати з моделі всі об'єкти з компонентом `Renderer`;
- додати компонент `BoxCollider`, якщо об'єкт його не має;
- сортувати створений масив за ім'ям.

```

if (Model0 != null & Check_Object)
{
    //===== Додати з моделі всі об'єкти з компонентом Renderer =====//
    for (int i = 0; array.Length < Model0.GetComponentInChildren<Renderer>().Length; i++)
    {
        array = array.Concat(new Renderer[] { Model0.GetComponentInChildren<Renderer>()[i] }).ToArray();
        //===== Додати BoxCollider =====//
        if (!Model0.GetComponentInChildren<Renderer>()[i].gameObject.GetComponent<BoxCollider>())
        {
            Model0.GetComponentInChildren<Renderer>()[i].gameObject.AddComponent<BoxCollider>();
        }
    }
    Check_Object = false;

    //===== Сортування за ім'ям =====//
    var sortedArray = array.OrderBy(go => go.name).ToArray();
    array = sortedArray;
}

```

Рисунок 3.16 – Принцип отримання масиву дочірніх об'єктів

Змінна «Model0» несе інформацію про обраний об'єкт, а «Check_Object» необхідна для того, щоб операція проводилася лише один раз після вибору об'єкту. Сортвання масиву дає можливість контролювати задану послідовність відображення дочірніх об'єктів, та встановити її за допомогою додання індексу перед назвою елементів при їх моделюванні. Подібне найменування дозволяє швидко адаптувати існуючі моделі або проводити налаштування моделей, що знаходяться в розробці для використання у створеному додатку.

Дочірні об'єкти, з яких складається модель можуть називатися подібним чином (3.1):

$$\text{number_Name}, \quad (3.1)$$

де `number` – числовий індекс об'єкту відмінний від 0;

`Name` – ім'я дочірнього об'єкту моделі.

Обраний об'єкт, згідно з розробленим алгоритмом, розміщується у середовищі доповненої реальності завдяки коду представленому на рис. 3.17.

Якщо користувач обрав роботу додатку з відображенням по маркеру, то відповідний компонент «ARTrackedImageManager» буде активовано. Даний код (рис. 3.18) необхідний для корекції розташування об'єкту відносно маркеру та виконується тільки у разі його знаходження. Для зміни положення моделі її новими значеннями локальної позиції та кута повороту стають значення маркеру та / або значення слайдерів.

```

if (ChooseObject)
{
    if (MarkerUsage) ARTrackedImageManager.enabled = true;

    if (!MarkerUsage || tracked)
    {
        Instantiate(ObjectToSpawn, new Vector3(0, 0, 0), Quaternion.Euler(0f, 0f, 0f));
    }

    if (GameObject.FindWithTag("Model0"))
    {
        GameObject.FindWithTag("Model0").name = "Model0";
        ImageControllerScript.Model0 = GameObject.Find("Model0");
        Preset_manager();

        positionChange = new Vector3(preset_values[0], preset_values[1], preset_values[2]);
        rotationChange = new Vector3(preset_values[3], preset_values[4], preset_values[5]);
        scaleChange = new Vector3(preset_values[6], preset_values[7], preset_values[8]);

        sliderP = positionChange;
        sliderR = rotationChange;
        sliderS = scaleChange;

        if (!MarkerUsage)
        {
            ImageControllerScript.Model0.transform.localPosition += positionChange;
        }
        ImageControllerScript.Model0.transform.eulerAngles += rotationChange;
        ImageControllerScript.Model0.transform.localScale = scaleChange;

        Assembly_active.SetActive(true);
        CancelAssembly_active.SetActive(true);
        MarkerUsage_active.SetActive(false);
        Settings_active.SetActive(true);
        ChooseObject = false;
    }
}

```

Рисунок 3.17 – Блок обробки розміщення нового об'єкту

Сообщение Unity | Ссылка: 0

```

void Awake()
{
    ARTrackedImageManager = GetComponent<ARTrackedImageManager>();
}

```

Сообщение Unity | Ссылка: 0

```

void OnEnable() => ARTrackedImageManager.trackedImagesChanged += OnChanged;

```

Сообщение Unity | Ссылка: 0

```

void OnDisable() => ARTrackedImageManager.trackedImagesChanged -= OnChanged;

```

Ссылка: 2

```

void OnChanged(ARTrackedImagesChangedEventArgs eventArgs)
{
    if (eventArgs.updated.Count > 0)
    {
        tracked = true;
        ARTrackedImage trackedImage = eventArgs.updated[0];

        if (ImageControllerScript.Model0)
        {
            ImageControllerScript.Model0.transform.localPosition = trackedImage.transform.position + sliderP;
            ImageControllerScript.Model0.transform.eulerAngles = trackedImage.transform.eulerAngles;
            if (rotationLock) ImageControllerScript.Model0.transform.eulerAngles = sliderR;
            ImageControllerScript.Model0.transform.localScale = sliderS;
        }
    }
}

```

Рисунок 3.18 – Код для корекції розміщення об'єкту відносно маркера

Наступний код (рис. 3.19) реалізує зміну значень відповідно до поточного пункту налаштувань, таким чином керування 9 значеннями об'єкта відбувається за допомогою 3 слайдерів.

```

if (ImageControllerScript.Model0 != null)
{
    if (Settings_set == 1) sliderP = new Vector3(sliderX.value, sliderY.value, sliderZ.value);
    if (Settings_set == 2) sliderR = new Vector3(sliderX.value, sliderY.value, sliderZ.value);
    if (Settings_set == 3) sliderS = new Vector3(sliderX.value, sliderY.value, sliderZ.value);
    if (scaleLock) sliderS = new Vector3(sliderZ.value, sliderZ.value, sliderZ.value);

    if (!MarkerUsage)
    {
        ImageControllerScript.Model0.transform.localPosition = sliderP;
        ImageControllerScript.Model0.transform.eulerAngles = sliderR;
        ImageControllerScript.Model0.transform.localScale = sliderS;
    }
}

```

Рисунок 3.19 – Блок обробки значень компонентів Slider

Так як було використано 3 слайдери для 9 значень, то при кожній зміні пункту налаштувань або збереженні пресету для кожного з параметрів (позиція, кут повороту, розмір) виконується створений метод. На рис. 3.20 представлений метод для налаштування параметру Position.

Ссылка: 4

```

void Position_task()
{
    Settings_set = 1;
    Settings_title();
    SliderShow_task();

    scaleLock = false;

    sliderX.minValue = (positionChange.x * -10);
    sliderX.maxValue = (positionChange.x * 10);
    if (positionChange.x < 0.5f) { sliderX.minValue = -1f; sliderX.maxValue = 1f; }
    sliderX.value = sliderP.x;

    sliderY.minValue = (positionChange.y * -10);
    sliderY.maxValue = (positionChange.y * 10);
    if (positionChange.y < 0.5f) { sliderY.minValue = -1f; sliderY.maxValue = 1f; }
    sliderY.value = sliderP.y;

    sliderZ.minValue = (positionChange.z * -10);
    sliderZ.maxValue = (positionChange.z * 10);
    if (positionChange.z < 0.5f) { sliderZ.minValue = -1f; sliderZ.maxValue = 1f; }
    sliderZ.value = sliderP.z;

    sliderP = new Vector3(sliderX.value, sliderY.value, sliderZ.value);
}

```

Рисунок 3.20 – Реалізація адаптивної зміни значень слайдерів при активному пункті налаштувань «Position»

У процесі розробки було виявлено, що для активації альфа-каналу (RGBA) необхідно попередньо розмістити будь-який об'єкт зі створеним прозорим матеріалом. Інакше розміщений об'єкт не змінював своєї прозорості. Для адаптивної зміни прозорості було створено також ще один слайдер. Залежно від його значення буде змінюватися значення альфа-каналу об'єкту.

При налаштуванні параметрів об'єкту реалізовано відображення поточного пресету, обраного параметру та положення, повороту і масштабу об'єкту. При збірці замість цих значень відображається поточний крок.

Об'єкти для яких створено BoxCollider є придатними для взаємодії. Код, що представлений на рис. 3.21, необхідний для отримання ім'я об'єкту по якому відбулося натискання.

```

if (Input.touchCount > 0)
{
    Touch touch = Input.GetTouch(0);

    if (touch.phase == TouchPhase.Began)
    {
        Ray ray = ARCamera.ScreenPointToRay(touch.position);
        RaycastHit hitObject;

        if (Physics.Raycast(ray, out hitObject))
        {
            if (hitObject.transform.name != null) temp = GameObject.Find(hitObject.transform.name);
        }
    }
}

```

Рисунок 3.21 – Реалізація взаємодії з об'єктом

Для виділення контуру об'єкту використовується підключений плагін «QuickOutline». Було встановлено необхідні параметри контуру, а логіку роботи реалізовано у скрипті «ImageController» (рис. 3.22).

```

if (temp.GetComponent<Outline>() == null)
{
    var outline = temp.AddComponent<Outline>();
    outline.OutlineMode = Outline.Mode.OutlineAll;
    outline.OutlineColor = Color.green;
    outline.OutlineWidth = 4f;
    temp = null;
}

```

Рисунок 3.22– Встановлення контуру для обраного об'єкту

3.3 Проведення експерименту

Виходячи з теми кваліфікаційної роботи, предметом дослідження буде розроблене ПЗ з використанням технології доповненої реальності.

Проведений експеримент полягає у наступному:

- дослідження використання QR та AR маркерів у розробленому додатку при різних розмірах та різному освітленні, вибір оптимального маркеру на основі отриманих даних;
- дослідження роботи додатку на пристроях з різними розмірами дисплею та значеннями продуктивності системи.

У першій частині експерименту було проведено заміри відстані на якій маркери будуть зчитуватися залежно від їх розмірів та освітлення. Отримані результати занесено до відповідних таблиць та побудовано діаграму, а також визначено оптимальний розмір та тип маркеру залежно від отриманих даних.

Для проведення експерименту було надруковано наступні маркери рис. 3.23 та виміряно їх реальні розміри, що наведені в табл. 3.1.



Рисунок 3.23 – Обрані зображення QR та AR міток

Таблиця 3.1 – Розміри міток

№	Назва маркеру	Висота, мм	Ширина, мм
1	1_100dpi	69	69
2	1_600dpi	180	180
3	2_100dpi	82	82
4	2_600dpi	180	180
5	3_100dpi	96	96
6	3_600dpi	180	180
7	ar-marker_1	19	19
8	ar-marker_1	39	39
9	ar-marker_1	59	59
10	ar-marker_1	79	79
11	ar-marker_2	54	54
12	ar-marker_2	72	72

Згідно з ДБН В.2.5-28:2018 [32] до освітлення приміщень виробничих підприємств висувають наступні вимоги (табл. 3.2).

Таблиця 3.2 – Вимоги до освітлення приміщень виробничих підприємств

Монтажні операції	Норми освітленості $E_{\text{екс}}$, Лк
грубі	300
середньої точності	500
високої точності	750
надточні	1000
Виробництво електронної техніки, випробування і контроль	1500

Перевіримо розпізнавання маркерів при інтенсивності освітлення для монтажних операцій середньої точності та виробництва електронної техніки, значення яких становлять 500 Лк та 1500 Лк (табл. 3.3-3.4).

Таблиця 3.3 – Мінімальна відстань відстеження маркеру при 500 Лк

№	Назва маркеру	$E_{\text{екс}}$, Лк	Відстань до маркеру, мм
1	1_100dpi	500	405
2	1_600dpi	500	1000
3	2_100dpi	500	470
4	2_600dpi	500	890
5	3_100dpi	500	482
6	3_600dpi	500	840
7	ar-marker_1	500	105
8	ar-marker_1	500	200
9	ar-marker_1	500	304
10	ar-marker_1	500	456
11	ar-marker_2	500	280
12	ar-marker_2	500	378

Таблиця 3.4 – Мінімальна відстань відстеження маркеру при 1500 Лк

№	Назва маркеру	$E_{\text{екс}}$, Лк	Відстань до маркеру, мм
1	1_100dpi	1500	408
2	1_600dpi	1500	1110
3	2_100dpi	1500	496
4	2_600dpi	1500	1080
5	3_100dpi	1500	535
6	3_600dpi	1500	970
7	ar-marker_1	1500	98
8	ar-marker_1	1500	194
9	ar-marker_1	1500	275
10	ar-marker_1	1500	432
11	ar-marker_2	1500	270
12	ar-marker_2	1500	364

При виборі оптимального маркеру необхідно враховувати значення відстані розпізнавання маркеру відносно до його ефективної площі. Це відношення можна розрахувати наступним чином (3.2):

$$K_{\text{еф}} = \frac{l}{s} \cdot 100, \quad (3.2)$$

де $K_{\text{еф}}$ – коефіцієнт ефективності маркеру;

l – відстань до маркеру;

s – площа маркеру.

Розраховані значення занесено до табл. 3.5.

Таблиця 3.5 – Розраховані значення коефіцієнту ефективності

№	Назва маркеру	S, мм ²	500 Лк, $K_{\text{еф}}$	1500 Лк, $K_{\text{еф}}$
1	1_100dpi	4761	8,507	8,570
2	1_600dpi	32400	3,086	3,426
3	2_100dpi	6724	6,990	7,377
4	2_600dpi	32400	2,747	3,333
5	3_100dpi	9216	5,230	5,805
6	3_600dpi	32400	2,593	2,994
7	ar-marker_1	361	29,086	27,147
8	ar-marker_1	1521	13,149	12,755
9	ar-marker_1	3481	8,733	7,900
10	ar-marker_1	6241	7,307	6,922
11	ar-marker_2	2916	9,602	9,259
12	ar-marker_2	5184	7,292	7,022

Завдяки отриманим даним було побудовано діаграму ефективності маркерів (рис. 3.24). Отримана діаграма дає зрозуміти, що розпізнавання QR та

AR маркерів залежить від інтенсивності освітлення, так при освітленості в 500 Лк QR мітки мають дещо гіршу ефективність, ніж при освітленості в 1500 Лк. AR мітки навпаки, краще розпізнаються при освітленості в 500 Лк.

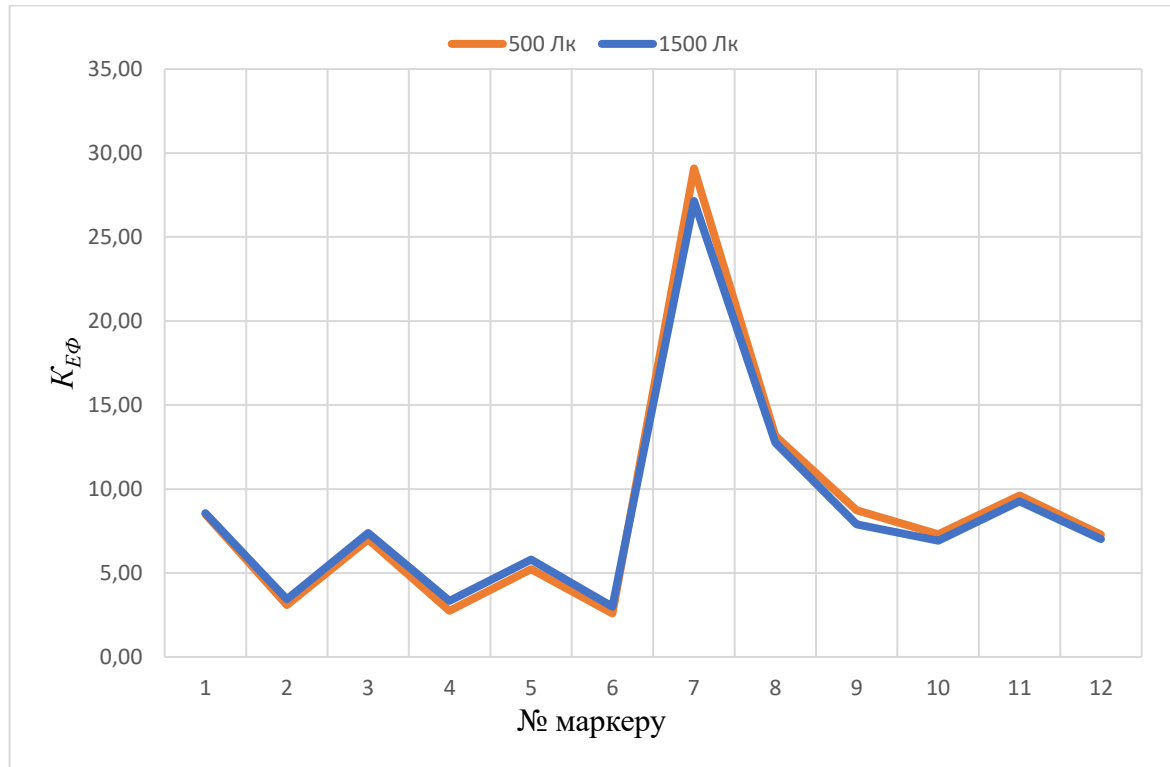


Рисунок 3.24 – Діаграма ефективності використаних маркерів

Судячи з отриманого коефіцієнту ефективності, можна побачити, що AR мітка з найменшою площею має найбільшу ефективність. Проте, до недоліку такої мітки можна віднести те, що вона має найгіршу відстань розпізнавання, 105 та 98 мм відповідно до інтенсивності освітлення.

Враховуючи отриману ефективність для 1 та 12 маркерів, а також їх приблизно однакові розміри, 69 мм та 72 мм відповідно, можна побачити, що вони мають приблизно однакову ефективність. Однак QR мітка з меншими розмірами розпізнається краще, ніж AR з дещо більшими. Аналогічна ситуація з результатами 3 та 10 маркерів.

Стосовно результатів 2, 4 та 6 маркерів. Не зважаючи на те, що дані маркери мали найбільші розміри та отримали найгіршу ефективність – вони

мають найкращу швидкість відображення тривимірної моделі у середовищі доповненої реальності. Це було перевірено та доведено експериментально.

Дослідивши використання QR та AR маркерів у розробленому додатку за допомогою проведеного експерименту було встановлено, що 7 та 2 маркер мають найбільш вагомій відмінності та їх можна рекомендувати для подальшого використання. В залежності від потреб, доцільно буде використовувати відповідний маркер.

Для невеликих об'єктів – однозначно це буде 7 маркер, але слід враховувати, що додаток не одразу відображає на ньому модель у середовищі віртуальної реальності. Якщо більш точно, то це стосується першого відображення об'єкту у активній сесії, подальше відображення об'єкту, наприклад при переміщенні маркеру відбувається досить добре.

Для великих об'єктів – доцільно буде використати 2 маркер через найбільшу відстань зчитування.

Інші перевірені маркери лише дещо відрізняються своїм коефіцієнтом ефективності в залежності від освітлення у робочій зоні.

До головних факторів похибки у проведеному експерименті можна віднести наступні:

- нерівномірність світлового потоку на маркері, що може негативно або позитивно вплинути на зчитування, в залежності від того QR чи AR мітка використовується;

- під час робочого процесу, якщо в приміщенні є пил або щось інше, що може забруднити лінзу камери пристрою, можна отримати різну ефективність розпізнавання, від -10 мм до +10 мм при оптимальній відстані зчитування близько 400 мм.

У якості другої частини експерименту розроблений додаток було встановлено загалом на 4 пристрої:

- OnePlus 7T;
- Realme 7 Pro;
- Xiaomi Mi 5S;

– Redmi 5 Plus.

Серед даних пристроїв усі, окрім третього, мають офіційну операційну систему. На третьому пристрої встановлено модифіковану операційну систему.

На перших двох пристроях додаток працює повноцінно. До відображуваного інтерфейсу та функціоналу зауважень немає.

На третьому пристрої додаток якийсь час не запускався, а коли це вдалося – виявилось, що зображення з камери не відображається. Необхідний для роботи пакет «Сервіси Google Play для AR» було встановлено та оновлено, а дозвіл на роботу з камерою надано. Проте це не виправило становище і додаток на даному пристрою працює без зображення з камери.

Останній пристрій, як виявилось після запуску додатку, не підтримує пакет для роботи з доповненою реальністю.

Стосовно перших двох пристроїв, другий, не зважаючи на більш скромні технічні характеристики (приблизно у 1,5 рази) та трохи довший час запуску додатку – швидше опрацьовує виділення обраного об'єкту контуром. Скоріше за все справа у оптимізації операційної системи. При передачі конфігураційного файлу з одного пристрою на другий, результат був очікуваний, конфігурація почала відображатися тільки після того, як додатку було надано дозвіл на роботу з файлами. Це може викликати деякі незручності у майбутніх користувачів, тому перед повноцінним впровадженням необхідно зробити можливим автоматичний запит на доступ до файлів.

Також розроблений додаток має високі вимоги до продуктивності пристрою. Було виявлено, що запис відео у 4K, що становить 3840 на 2160 пікселів, при 60 кадрах в секунду менше навантажує графічний та центральний процесори, ніж додаток, в якому відеопотік має роздільну здатність 640 на 480 пікселів та частоту 30 кадрів в секунду.

3.4 Охорона праці

Розробка велася за робочим місцем у приміщенні, розміри якого становлять $1,9 \text{ м} \times 2,8 \text{ м} \times 4,1 \text{ м}$, ширина, висота та довжина відповідно. Також було дотримано забезпечення достатнім для здійснення роботи рівнем освітлення (природного та штучного – у темну пору доби) та звукоізоляції. Для регуляції рівня освітлення природним світлом застосовано ролети.

При організації та обладнанні робочого місця було дотримано рекомендовану площу, відведену на одне робоче місце, що має становити не менше 6 м^2 та об'єм – не менше 20 м^3 . Отримані показники становлять $7,8 \text{ м}^2$ та $21,8 \text{ м}^3$ відповідно.

Конструкція робочого місця повністю забезпечує підтримання оптимальної робочої пози, що дозволяє виконувати роботу з мінімальним напруженням тіла та уникнути перевтоми в ході і після закінчення робочого процесу. Раціональна робоча поза має важливе значення для збереження здоров'я. Установлено, що робота в зігнутому положенні збільшує затрати енергії на 20 %, а при значному нахиленні – на 45 % порівняно з прямим положенням корпусу.

Робоче місце розташовано відносно джерела природного світла таким чином, що в денну пору світло потрапляє з лівого боку. Також робоче місце відповідає сучасним вимогам ергономіки:

- стіл має висоту поверхні 730 мм, ширину 1400 мм і глибину 820 мм. Дані параметри відповідають нормам та забезпечують можливість виконання операцій в зоні досяжності користувача;

- використовується підйомно-поворотний робочий стілець з можливістю регулювання висоти з напівм'якою нековзкою поверхнею сидіння, що легко чиститься і не електризується;

- екран комп'ютера розташований на оптимальній відстані від користувача, що становить 620 мм;

– персональний комп'ютер та його комплектуючі захищені від потрапляння під прямі промені сонячного світла та під дію інших джерел тепла завдяки використанню ролетів на вікні.

Далі наведено основні вимоги безпеки під час роботи з комп'ютером [33].

Щодня перед початком роботи оператор повинен:

- оглянути своє робоче місце, та у разі виявлення ознак пошкодження обладнання інформувати свого безпосереднього керівника;
- відрегулювати освітленість на робочому місці, переконатися в відсутності відблисків на екрані комп'ютера, відсутності зустрічного світла;
- перевірити правильність підключення обладнання ПК до електромережі;
- очистити екран комп'ютера від пилу та інших забруднень;
- перевірити правильність організації робочого місця й за необхідності провести відповідні коригування.

Оператор під час роботи зобов'язаний:

- виконувати тільки ту роботу, яку йому було доручено;
- підтримувати порядок і чистоту на робочому місці;
- тримати відкритими всі вентиляційні отвори обладнання;
- коректно закрити всі активні завдання у разі припинення роботи з комп'ютером;
- негайно відключити комп'ютером від електричної мережі у разі виникнення аварійної ситуації.

При виконанні робіт оператор персонального комп'ютера повинен:

- витримувати відстань від очей до екрану ПК в межах 60 – 70 см;
- дотримуватися внутрішньозмінного режиму праці та відпочинку, регламентованих перерв у роботі, а саме (при 8-годинній денній робочій зміні):
 - для розробників програм – тривалістю 15 хвилин через кожну годину роботи;
 - для інших категорій працівників – тривалістю 15 хвилин через кожні дві години роботи;

– для операторів комп'ютерного набору – тривалістю 10 хвилин, після кожної години роботи.

Під час регламентованих перерв рекомендується виконувати комплекси вправ для очей, рук, хребта, поліпшення мозкового кровообігу тощо. Про виявлення несправності обладнання або інших факторів, які створюють загрозу для життя або здоров'я працівників, необхідно негайно інформувати свого безпосереднього керівника.

При роботі на персональному комп'ютері не допускається:

- виконання ремонту та налагодження комп'ютерної техніки безпосередньо на робочому місці оператора;
- зберігання біля комп'ютера паперу та інших носіїв інформації, запасних блоків, деталей тощо, якщо вони не використовуються для поточної роботи;
- відключення захисних пристроїв, самочинні зміни в конструкції комп'ютера;
- використання комп'ютерів, на екранах яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на екрані тощо;
- доторкання до задньої панелі системного блоку при включеному живленні;
- вимикання живлення під час виконання активного завдання;
- попадання вологи на поверхню системного блоку, монітора, клавіатури, дисководів, принтерів та інших пристроїв;
- приймання напоїв та їжі на робочому місці.

Після закінчення роботи з використанням ПК необхідно дотримуватися такої послідовності вимикання обладнання:

- закрити всі активні завдання;
- переконатися у відсутності дисків у дисководах;
- використавши опцію "Завершення роботи" у меню "Пуск", вимкнути живлення системного блоку;
- вимкнути живлення всіх комп'ютерів;
- вимкнути блок аварійного живлення (за наявності);

– відключити комп'ютер від електромережі, при цьому забороняється тягнути штепсельну вилку за дріт.

3.5 Висновки до третього розділу

В ході виконання третього розділу розроблено інтерфейс програмного засобу. Для цього було сформовано основні вимоги та проведено їх втілення у проект в середовищі розробки Unity. Іконки для кнопок графічного інтерфейсу виконано завдяки використанню графічного редактору Adobe Photoshop. Дизайн іконок був розроблений надихнувшись мінімалістичним підходом та змістовним сенсом їх дій. Зображення для кнопок зі списку виконано за допомогою візуалізації створених об'єктів у середовищі моделювання Autodesk Fusion 360.

Отримані зображення та моделі було додано до об'єкту та відповідно налаштовано. В результаті виконаних дій було сформовано необхідний графічний інтерфейс додатку.

Програмний код ПЗ був виконаний згідно з поставленими вимогами до розроблюваного додатку, що були сформовані у ході виконання кваліфікаційної роботи. Також наведено методи, що реалізують деякі дії, які було описано в алгоритмі програмного засобу.

Результатом розробки ПЗ став створений додаток з використанням технологій доповненої реальності.

Роботу додатку перевірено завдяки проведенню експерименту, що полягав у дослідженні використання QR та AR міток при різних розмірах та різному освітленні, виборі оптимального маркеру на основі отриманих даних та дослідженню роботи додатку на пристроях з різними розмірами дисплею та значеннями продуктивності системи.

Відповідно до отриманих даних встановлено оптимальний розмір та тип використовуваного маркеру, в залежності від розміру об'єкту збірки та інтенсивності освітлення у приміщенні, а також перевірено роботу додатку на різних пристроях.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розроблено додаток з використанням технології доповненої реальності, якому була дана дослівна назва «Assembly Guide», тобто «Керівництво по збірці» або «Керівництво по складанню». Основна задача при виконанні завдання полягала у використанні знань отриманих у першому та другому розділах проведеної роботи, а також освоєнні та втіленні нових навичок, набутих у третьому розділі.

Розробка додатку у повній мірі залежала від виконання усіх етапів, що були виконані та описані у розділах даної роботи. Таким чином проведене дослідження сучасного стану використання доповненої реальності, а саме аналіз принципу роботи, методів реалізації, особливостей використання та сучасних рішень, вплинуло на загальне розуміння роботи та використання доповненої реальності на виробництві.

Також сформовано основні вимоги до розроблюваного додатку, опираючись на які були обрані середовища розробки, що вплинули на досвід, отриманий в процесі виконання поставлених задач. Формування технічного завдання допомогло висвітлити основні вимоги до розроблюваного додатку, виходячи із яких було розроблено структурну схему, об'єктну модель та алгоритм програмного засобу. Розроблений алгоритм у повній мірі допоміг сформувати основні логічні зв'язки, що далі були реалізовані у додатку.

Виходячи з аналізу подібних рішень, розробленого алгоритму та поставлених вимог було створено інтерфейс програмного засобу. Розроблений інтерфейс задовольняє поставленим у ході виконання роботи цілям, що було перевірено проведеним експериментом.

В ході розробки програмного коду втілено всі ідеї, вимоги та рекомендації зі створення додатку та підтверджено це проведеним експериментом. Також можу зазначити, що використаний при написанні ПЗ SDK AR Foundation не підтримує роботу з Web-камерою ПК, що створило деякі складності під час

розробки. Та не зважаючи на все, розробка додатку була проведена до кінця, а його успішна робота була підтверджена проведенням експериментом.

Розроблення автоматизованої складальної ділянки за допомогою методів доповненої реальності має в собі на увазі використання розробленого під час виконання кваліфікаційної роботи додатку на реальному виробництві. Додаток було перевірено при максимально наближених, до виробничих, умовах де він показав себе з позитивної сторони.

Завдяки основним перевагам розробленого додатку, таким як можливість оцінити об'єкт збірки на будь-якому етапі, від стадії його проектування до перевірки на контролі якості, вважаю, що розроблене рішення є готовим для впровадження і подальшого використання на підприємстві, а саме на автоматизованих складальних ділянках.

Використання доповненої реальності на виробництві дозволить швидше навчати нових фахівців та допоможе в роботі існуючим фахівцям.

Застосування доповненої реальності надає краще сприйняття візуальної інформації, дозволяє підтримувати високий рівень безпеки на підприємстві, а також призване знизити витрати підприємств на навчанні співробітників, підвищити швидкість і надійність їх роботи на складальній ділянці.

Таким чином, використання доповненої реальності відкриває нові можливості для підприємств, та здатне зробити їх більш ефективними та надійними, а також позитивно впливає на автоматизацію виробництва.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП “УкрНДНЦ”, 2016. – 30 с.

2. Методичні вказівки до розробки й оформлення кваліфікаційної роботи студентів другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп’ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп’ютерно-інтегровані технологічні процеси і виробництва», «Комп’ютеризовані та робототехнічні системи» / Упоряд. І.Ш. Невлюдов, В.В. Євсєєв, С.П. Новоселов, О.М. Цимбал., – Харків: ХНУРЕ, 2021. – 50 с.

3. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. № 143. – Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi -universitetu.pdf

4. Невлюдов І.Ш. Основи наукових досліджень: Навч. посібник / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривий Ріг: Криворізький коледж НАУ, 2019. – 396 с.

5. Мажара А. Є. Індустрія 4.0 як промислове виробництво майбутнього [Текст] / Мажара А. Є., Павленко В. І., Бурма О. М., // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED2019): збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2019. – Вип. 2. – 199 с.

6. Мажара А. Є. Використання доповненої реальності у сучасному виробництві [Текст] / Мажара А. Є., // Міждисциплінарні наукові дослідження та перспективи їх розвитку: матеріали I Міжнародної студентської наукової

конференції (Т. 2), м. Київ, 7 травня, 2021 р. / Молодіжна наукова ліга. – Вінниця: Європейська наукова платформа, 2021. – 18-20 с.

7. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.

8. M. C. Augmented Reality and Virtual Reality: New Trends in Immersive Technology / C. M., T. H. Jung, S. M. C. Loureiro. – Progress in IS. – Springer, Cham, 2021. – 314 p.

9. Технології, системи та додатки доповненої реальності. Мультимедійні інструменти і додатки / Дж. Карміньяні, Б. Фурт, М. Анісетті та ін. – 2021. – 302–314 с.

10. Чандар Дж. Безмаркерне відстеження камери в реальному часі для додатків доповненої реальності і віщання / Г.А. Томас, Д. Стрікер // Журнал обробка зображень в реальному часі. Проект MATRIS. – 2017. – С. 54–62.

11. Підхід доповненої реальності для інтеграції практичних занять в системи електронного навчання / Н. El Kabtane, M. Sadgal, M. El Adnani, Y. Mourdi. – 2016. – 114–119 с.

12. Fite-Georgel P. Is There A Reality in Industrial Augmented Reality, Mixed and Augmented Reality (ISMAR) – 2021. – 181–195 p.

13. Chen C.J. Автоматизоване позиціонування віртуальної 3D-сцени в системі керування збіркою та демонтажем на основі AR // C.J. Chen, J. Hong, Wang, S.F. // Міжнародний журнал передових технологій виробництва. – 2014. – С. 682-690.

14. Remote AR/MR Assistant [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://program-ace.com/solutions/remote-ar-mr-assistant/>. – 06.12.2021.

15. Virtual Product Configurator [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://program-ace.com/solutions/virtual-product-configurator/>. – 05.12.2021.

16. Potentials of Augmented Reality in Training [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://www.sciencedirect.com/science/article/pii/S2351978919303774>. – 05.12.2021.

17. How Augmented Reality Can Help You with Remote Collaboration [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://www.bairesdev.com/blog/augmented-reality-remote-collaboration/>. – 05.12.2021.

18. Augmented Reality in Assembly Systems: State of the Art and Future Perspectives [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : https://link.springer.com/chapter/10.1007/978-3-030-72632-4_1. – 05.12.2021.

19. Augmented Reality Meets Artificial Intelligence in Robotics: A Systematic Review [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://www.frontiersin.org/articles/10.3389/frobt.2021.724798/full>. – 05.12.2021.

20. DIGITAL TWIN PRODUCT DEVELOPMENT [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://program-ace.com/expertise/digital-twin-development/>. – 05.12.2021.

21. Overview of ARCore and supported development environments [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://developers.google.com/ar/develop>. – 05.12.2021.

22. Vuforia Engine developer portal [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://developer.vuforia.com/>. – 06.12.2021.

23. AR Foundation [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : https://wiki.etc.cmu.edu/index.php/AR_Foundation. – 06.12.2021.

24. About AR Foundation [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>. – 06.12.2021.

25. Wells R. Unity 2020 By Example: A project-based guide to building 2D, 3D, augmented reality, and virtual reality games from scratch / Robert Wells., 2020. – 676 p.

26. Prefabs [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/Manual/Prefabs.html>. – 07.12.2021.

27. Sprite [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/ScriptReference/Sprite.html>. – 07.12.2021.

28. Class ARSessionOrigin [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@1.0-preview.8/api/UnityEngine.XR.ARFoundation.ARSessionOrigin.html>. – 07.12.2021.

29. Android Manifest [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/Manual/android-manifest.html>. – 07.12.2021.

30. GameObject.SetActive [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/ScriptReference/GameObject.SetActive.html>. – 08.12.2021.

31. Button.onClick [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://docs.unity3d.com/2018.3/Documentation/ScriptReference/UI.Button-onClick.html>. – 08.12.2021.

32. ПРИРОДНЕ І ШТУЧНЕ ОСВІТЛЕННЯ [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : https://www.minregion.gov.ua/wp-content/uploads/2018/09/DBN_Osvitlennya-ostatochna.pdf. – 09.12.2021.

33. ВИМОГИ ОХОРОНИ ПРАЦІ ПРИ ВИКОНАННІ РОБІТ НА ПЕРСОНАЛЬНОМУ КОМП'ЮТЕРІ [Електронний ресурс]. Електрон. текстові дані. – Режим доступу : <https://selidovo-rada.gov.ua/novini/vimogi-okhoroni-pratsi-pri-vikonanni-robit-na-personalnomu-komp-yuteri>. – 09.12.2021.