МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії
програмного забезпечення

_____ Євген ДАВИДЕНКО

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОТСВІТНЬОГО СТУПЕНЯ МАГІСТРА

Застосунок із доповненою реальністю для планування інтер'єру квартири

Спеціальність 121 Інженерія програмного забезпечення

Освітня програма «Інженерія програмного забезпечення»

Здобувач **Сергій ЧИМБІР**
«__» _____ 2025 р.

Керівник роботи
PhD, доцент

Гліб ГОРБАНЬ
«__» _____ 2025 р.

Миколаїв – 2025

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

Факультет	Комп’ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Другий (магістерський)
Освітній ступінь	Магістр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри інженерії
програмного забезпечення
_____ Євген ДАВИДЕНКО
«____» _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну магістерську роботу здобувача вищої освіти

Чимбір Сергія

1. Тема кваліфікаційної роботи Застосунок із доповненою реальністю для планування інтер’єру квартири затверджена наказом ректора ЧНУ ім. Петра Могили № 182 від «2» липня 2025 р.
2. Строк представлення кваліфікаційної роботи «___» _____ 2025 р.
3. Очікуваний результат роботи та початкові дані якщо такі потрібні.

Очікуваним результатом є реалізований застосунок із доповненою реальністю для планування інтер’єру.

4. Перелік питань, що підлягають розробці:

- аналіз предметної галузі;
- аналіз доступних фреймворків і бібліотек для розробки AR-

застосунків;

- виконати проєктування та моделювання системи планування інтер’єру в режимі доповненої реальності;
- реалізувати планування інтер’єру у мобільному застосунку;
- тестування результату роботи застосунку.

5. Перелік графічних матеріалів:

- презентація.

6. Консультанти:

Консультант	**Кафедра (організація)**	**Частина роботи**

Дата видачі завдання «____» _____ 20___ р

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: **Застосунок із доповненою реальністю для планування інтер'єру квартири**

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	01.09.2025	02.09.2025	виконано
2.	Огляд літератури за темою роботи	03.09.2025	04.09.2025	виконано
3.	Складання календарного плану КМР	05.09.2025	05.09.2025	виконано
4.	Аналіз предметної області	08.09.2025	12.09.2025	виконано
5.	Розробка проєктних рішень	15.09.2025	19.09.2025	виконано
6.	Моделювання та конструювання ПЗ	22.09.2025	26.09.2025	виконано
7.	Кодування, розробка користувацького інтерфейсу, тестування розробленого ПЗ, аналіз результатів тестування	29.09.2025	05.11.2025	виконано
8.	Оформлення КМР та презентації	06.11.2025	23.12.2025	виконано
9.	Попередній захист	24.11.2025	24.11.2025	виконано
10.	Завершення оформлення КМР та презентації	24.11.2025	13.11.2025	виконано
11.	Рецензування			виконано
12.	Відгук керівника КМР			виконано
13.	Захист кваліфікаційної роботи	22.12.2025	22.12.2025	виконано

Здобувач _____ **Сергій ЧИМБІР**

«__» _____ 2025 р.

Керівник роботи

PhD, доцент

кафедри ІПЗ _____ **Гліб ГОРБАНЬ**

«__» _____ 2025 р.

АНОТАЦІЯ

до кваліфікаційної магістерської роботи

Застосунок із доповненою реальністю для планування інтер’єру квартири

Здобувач 608м гр.: Чимбір Сергій

Керівник: канд. техн. наук, доцент Горбань Гліб

Актуальність: у сучасному світі цифрові технології відіграють ключову роль у сфері дизайну інтер’єру та просторового планування. Традиційні методи вибору меблів, кольорової гами та розміщення елементів у приміщенні можуть бути трудомісткими та не завжди дають можливість візуально оцінити кінцевий результат перед здійсненням змін. Застосування технологій доповненої реальності (AR) у цій галузі дозволяє значно спростити процес планування інтер'єру, надаючи користувачам змогу інтерактивно модифікувати приміщення в режимі реального часу без фізичних змін.

Доречність описаної теми визначається тим, що використання AR-застосунку для планування інтер’єру може значно зменшити кількість помилок при виборі меблів, заощадити кошти та час, а також зробити процес дизайну доступним для широкого кола користувачів без спеціальних знань.

Розробка мобільного застосунку для планування інтер’єру квартири за допомогою AR дозволить користувачам експериментувати з меблями, змінювати їхній розмір, розташування та колірну палітру, що сприятиме більш обґрунтованому прийняттю рішень. Це особливо актуально для дизайнерів, рієлторів та власників житла, які прагнуть оптимізувати простір і зробити його максимально комфортним. Даний підхід також допоможе зменшити витрати на перепланування, зменшуючи кількість помилок у виборі матеріалів та розташуванні елементів декору.

Об’єктом кваліфікаційної роботи є процес використання технологій доповненої реальності для візуального планування інтер’єру житлових приміщень.

Предметом кваліфікаційної роботи є методи, технології та середовище, що використовуються у розробці застосунків із доповненою реальністю.

Метою кваліфікаційної роботи є планування інтер’єру квартири, яка розробляється із використанням технологій доповненої реальності шляхом розробки мобільного застосунку.

Завданням кваліфікаційної роботи є:

1) розробка мобільного застосунку з інтеграцією доповненої реальності для моделювання дизайну інтер’єру;

2) створення функціоналу для взаємодії з віртуальними об'єктами – можливість додавання, переміщення, зміни кольору та матеріалів меблів та декору у реальному просторі;

3) реалізація алгоритмів візуалізації для коректного відображення 3D-об’єктів у просторі за допомогою камери мобільного пристрою;

4) розробка користувацького інтерфейсу, що забезпечить інтуїтивно зрозумілу взаємодію із застосунком;

5) оптимізація продуктивності застосунку для забезпечення плавної роботи навіть на мобільних пристроях середнього рівня;

6) оцінка ефективності шляхом тестування програми та аналізу зручності використання для кінцевих користувачів.

Кваліфікаційна робота включає вступ, чотири розділи, висновки, список використаних джерел та додаток А із демонстрацією програмного коду.

У вступі подано опис теми, мети, предмета й об’єкта магістерської роботи, визначено її завдання та подано додаткову інформацію про використані матеріали.

У першому розділі описаний аналіз обраної сфери, використання доповненої реальності у бізнес проєктах та житті, аналіз аналогів використання, Unity3D у якості рушія для розробки AR-застосунку.

Другий розділ присвячено аналізу алгоритмів комп’ютерного зору, аналізу моделей та інструментів візуалізації 3D об’єктів, аналізу доступних фреймворків та бібліотек та специфікацію вимог до розробляємого AR-застосунку.

У третьому розділі здійснено проєктування архітектури, сценаріїв використання та багатьох діаграм, для отримання повної картини застосунку який буде розроблятись.

У четвертому розділі описано реалізацію та тестування застосунку, проведено оцінку його ефективності.

У висновках узагальнено результати дослідження та розробки, наведено висновки щодо досягнення мети й завдань.

Кваліфікаційна робота викладена на 92 сторінках, складається із вступу, 4 розділів, загальних висновків, переліку джерел посилання з 29 найменувань та 1 додатків. Праця містить 1 таблицю та 49 рисунків.

Ключові слова: доповнена реальність, інтер'єр, дизайн, мобільний застосунок, AR, C#, Unity3D.

ABSTRACT

to the qualifying master's thesis

Augmented Reality Application for Apartment Interior Planning

Student of 608m group: Chymbir Serhii

Supervisor: PhD, Associate Professor Horban Hlib

Relevance: In the modern world, digital technologies play a key role in the field of interior design and spatial planning. Traditional methods of selecting furniture, color schemes, and arranging elements within a space can be time-consuming and do not always allow for a clear visual assessment of the final result before changes are made. The application of augmented reality (AR) technologies in this field significantly simplifies the interior planning process by enabling users to interactively modify a space in real time without physical changes.

The relevance of the described topic is determined by the fact that the use of an AR application for interior planning can significantly reduce the number of mistakes when choosing furniture, save time and money, and make the design process accessible to a wide range of users without specialized knowledge.

The development of a mobile application for apartment interior planning using AR will allow users to experiment with furniture, change its size, placement, and color palette, which will contribute to more informed decision-making. This is especially relevant for designers, real estate agents, and homeowners who seek to optimize space and make it as comfortable as possible. This approach will also help reduce remodeling costs by minimizing errors in the selection of materials and the placement of decorative elements.

The object of the qualification work is the process of using augmented reality technologies for the visual planning of residential interiors.

The subject of the qualification work is the methods, technologies, and development environments used in the creation of augmented reality applications.

The purpose of the qualification work is to plan the interior of an apartment using augmented reality technologies through the development of a mobile application.

The objectives of the qualification work are:

1) development of a mobile application with integrated augmented reality for interior design modeling;

2) creation of functionality for interaction with virtual objects, including the ability to add, move, and change the color and materials of furniture and decorative elements in real space;

3) implementation of visualization algorithms for the correct rendering of 3D objects in space using the mobile device camera;

4) development of a user interface that ensures intuitive interaction with the application;

5) optimization of application performance to ensure smooth operation even on mid-range mobile devices;

6) evaluation of effectiveness through application testing and analysis of usability for end users.

The qualification work consists of an introduction, four chapters, conclusions, a list of references, and Appendix A containing a demonstration of the program code.

The introduction presents a description of the topic, purpose, subject, and object of the master's thesis, defines its objectives, and provides additional information about the materials used.

The first chapter presents an analysis of the selected field, the use of augmented reality in business projects and everyday life, an analysis of existing solutions, and Unity3D as a development engine for an AR application.

The second chapter is devoted to the analysis of computer vision algorithms, models, and tools for 3D object visualization, the analysis of available frameworks and libraries, and the specification of requirements for the developed AR application.

The third chapter covers the design of the architecture, use cases, and various diagrams in order to obtain a complete picture of the application to be developed.

The fourth chapter describes the implementation and testing of the application and evaluates its effectiveness.

The conclusions summarize the results of the research and development and present conclusions regarding the achievement of the stated goals and objectives.

The master’s thesis is presented on 92 pages, consists of an introduction, 4 chapters, general conclusions, a reference list of 29 items, and 1 appendice. The work contains 1 table and 49 figures.

Keywords: augmented reality, interior, design, mobile application, AR, C#, Unity3D.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 4
ВСТУП 5
1 АНАЛІЗ ТЕХНОЛОГІЙ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ ПЛАНУВАННЯ ІНТЕР’ЄРУ 7
1.1 Аналіз предметної області 7
1.2 Використання технологій доповненої реальності 8
1.3 Аналіз аналогів використання технологій доповненої реальності 10
1.4 Визначення цільової аудиторії 18
1.4 Unity3D у якості рушія для розробки застосунку доповненої реальності 20
Висновки до розділу 1 22
2 АНАЛІЗ ТА МОДЕЛЮВАННЯ ВИМОГ 24
2.1 Аналіз алгоритмів комп’ютерного зору та AR-технологій для просторового моделювання 24
2.2 Аналіз моделей та інструментів візуалізації 3D-об’єктів 27
2.3 Аналіз доступних фреймворків і бібліотек для розробки AR-застосунків 29
2.4 Специфікація вимог до AR-застосунку для планування інтер’єру 32
Висновки до розділу 2 36
3 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ СИСТЕМИ ПЛАНУВАННЯ ІНТЕР’ЄРУ В РЕЖИМІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ 38
3.1 Сценарії використання системи 38
3.2 Діаграми станів системи 41
3.3 Діаграма прецедентів 43
3.4 Діаграма послідовності 45
3.5 Діаграма класів 47
3.6 Діаграма розгортання 49
Висновки до розділу 3 50
4 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ІЗ ПЛАНУВАННЯ ІНТЕР’ЄРУ В РЕЖИМІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ 52
4.1 Ознайомлення із базовим функціоналом Unity 3D 52
4.2 Створення логіки застосунку 55
4.3 Користувацький інтерфейс 60
4.4 Інтеграція технології доповненої реальності у проєкт 65

ВИСНОВКИ 75
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 77

ПЕРЕЛІК СКОРОЧЕНЬ

2D – Two-dimensional

3D – Three-dimensional

AR – Augmented Reality

FBX – Filmbox

FPS – Frames Per Second

GLTF – GL Transmission Format

IMU – Inertial Measurement Unit

JSON – JavaScript Object Notation

LiDAR – Light Detection and Ranging

LOD – Level of Detail

OBJ – формат файлу Wavefront OBJ

PBR – Physically Based Rendering

PNG – Portable Network Graphics

RANSAC – RANdom SAmple Consensus

SLAM – Simultaneous Localization and Mapping

UML – Unified Modeling Language

UV-розгортання – процес переведення 3D-моделі з об'ємного (3D) простору у пласке (2D) зображення

ОС – операційна система

ГБ – гігабайт

ВСТУП

Актуальність: у сучасному світі цифрові технології відіграють ключову роль у сфері дизайну інтер’єру та просторового планування. Традиційні методи вибору меблів, кольорової гами та розміщення елементів у приміщенні можуть бути трудомісткими та не завжди дають можливість візуально оцінити кінцевий результат перед здійсненням змін. Застосування технологій доповненої реальності (AR) у цій галузі дозволяє значно спростити процес планування інтер'єру, надаючи користувачам змогу інтерактивно модифікувати приміщення в режимі реального часу без фізичних змін.

Доречність описаної теми визначається тим, що використання AR-застосунку для планування інтер’єру може значно зменшити кількість помилок при виборі меблів, заощадити кошти та час, а також зробити процес дизайну доступним для широкого кола користувачів без спеціальних знань.

Розробка мобільного застосунку для планування інтер’єру квартири за допомогою AR дозволить користувачам експериментувати з меблями, змінювати їхній розмір, розташування та колірну палітру, що сприятиме більш обґрунтованому прийняттю рішень. Це особливо актуально для дизайнерів, рієлторів та власників житла, які прагнуть оптимізувати простір і зробити його максимально комфортним. Даний підхід також допоможе зменшити витрати на перепланування, зменшуючи кількість помилок у виборі матеріалів та розташуванні елементів декору.

Для виконання цього дослідження та отримання потрібних даних для аналізу буде використано створений проєкт з кваліфікаційної магістерської роботи – Unity 3D застосунок із використанням технологій доповненої реальності. У проєкті буде реалізовано функціонал розміщення та маніпуляції віртуальними меблями у реальному просторі за допомогою AR Foundation та бібліотек ARCore/ARKit. Для збору даних будуть використані методи тестування продуктивності, оцінки точності відображення та юзабіліті-

тестування, що дозволить визначити зручність і ефективність застосунку для кінцевих користувачів.

Об’єктом кваліфікаційної роботи є процес використання технологій доповненої реальності для візуального планування інтер’єру житлових приміщень.

Предметом кваліфікаційної роботи є методи, технології та середовище, що використовуються у розробці застосунків із доповненою реальністю.

Метою кваліфікаційної роботи є планування інтер’єру квартири, яка розробляється із використанням технологій доповненої реальності шляхом розробки мобільного застосунку.

Завданням кваліфікаційної роботи є:

1) розробка мобільного застосунку з інтеграцією доповненої реальності для моделювання дизайну інтер’єру;

2) створення функціоналу для взаємодії з віртуальними об'єктами – можливість додавання, переміщення, зміни розміру та кольору меблів та декору у реальному просторі;

3) реалізація алгоритмів візуалізації для коректного відображення 3D-об’єктів у просторі за допомогою камери мобільного пристрою;

4) розробка користувацького інтерфейсу, що забезпечить інтуїтивно зрозумілу взаємодію із застосунком;

5) оптимізація продуктивності застосунку для забезпечення плавної роботи навіть на мобільних пристроях середнього рівня;

6) оцінка ефективності шляхом тестування програми та аналізу зручності використання для кінцевих користувачів.

1 АНАЛІЗ ТЕХНОЛОГІЙ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ ПЛАНУВАННЯ ІНТЕР’ЄРУ

1.1 Аналіз предметної області

Доповнена реальність – це технологія, що дозволяє інтегрувати віртуальні об’єкти у реальний світ за допомогою пристроїв із камерою та сенсорами. У сфері дизайну інтер’єрів AR надає користувачу можливість візуально оцінити результат розміщення меблів чи елементів декору в приміщенні ще до того, як буде прийнято остаточне рішення про покупку чи монтаж [1].

Сучасні методи реалізації AR у задачах планування інтер’єру можна умовно поділити на кілька груп [2]:

- методи візуалізації у реальному часі;
- методи просторового вимірювання та побудови кімнати;
- методи інтерактивної взаємодії з об’єктами;
- методи оптимізації та перевірки;
- комбіновані методи.

Кожен із методів має свої особливості та сферу застосування. Наприклад, прості AR-застосунки для візуалізації меблів у приміщенні не потребують складної моделі кімнати, тоді як професійні дизайнерські рішення вимагають точних вимірів та перевірки відповідності об’єктів ергономічним нормам. Опишемо кожен метод окремо.

Методи візуалізації у реальному часі ґрунтуються на використанні камери мобільного пристрою для зчитування простору та накладення 3D-моделей. Використовуються інструменти на зразок ARKit (Apple) та ARCore (Google), які забезпечують виявлення площин, освітлення, масштабування та інтерактивне розміщення об’єктів.

Методи просторового вимірювання та побудови кімнати дозволяють користувачеві створити базову модель приміщення (стіни, вікна, двері) за

допомогою камери та сенсорів. Це дає змогу точніше співставляти віртуальні об’єкти з реальними параметрами простору.

Методи інтерактивної взаємодії з об’єктами передбачають додавання, переміщення, масштабування та обертання меблів чи інших елементів інтер’єру в реальному середовищі. Також застосовуються методи зміни текстур та кольорових палітр для оцінки різних варіантів дизайну.

Методи оптимізації та перевірки використовуються для оцінки ергономіки простору, виявлення колізій між об’єктами, перевірки достатності відстаней для проходу, а також для оцінки освітлення та стилістичної гармонії.

Комбіновані методи поєднують візуалізацію у реальному часі, просторове вимірювання та аналітику. Вони дозволяють користувачу не лише бачити об’єкти у кімнаті, а й отримувати рекомендації щодо їх оптимального розташування чи поєднання.

Під час створення мобільного застосунку необхідно враховувати цільову аудиторію та завдання, які він має виконувати.

1.2 Використання технологій доповненої реальності

Технології доповненої реальності сьогодні активно використовуються в різних сферах людської діяльності. Найбільш відомими прикладами є мобільні ігри (Pokémon Go), інтерактивні фільтри в соціальних мережах (Snapchat, Instagram), інструменти для навігації та освіти.

Окрім розважальної сфери, AR також широко застосовується у таких галузях [3]:

- наука – візуалізація експериментів, симуляція фізичних та хімічних процесів;
- освіта – інтерактивні навчальні матеріали, симулятори, 3D-посібники;
- медицина – візуалізація органів та систем людини для підготовки хірургів, створення тренажерів та навчальних моделей;

– військова сфера – симулятори, системи візуалізації тактичних карт, тренажери для пілотів і солдатів;

– медіа – інтерактивна реклама, AR-каталоги товарів, маркетингові кампанії;

– промисловість – AR-інструкції для монтажу обладнання, ремонтних робіт, навчання персоналу;

– транспорт і логістика – оптимізація маршрутів, візуалізація складських процесів, навігація у великих приміщеннях;

– фінанси – візуалізація великих обсягів даних, інтерактивні інструменти для презентації фінансових рішень.

Попри широке використання AR у багатьох напрямах, найбільш цікавим та актуальним для даної кваліфікаційної роботи є саме застосування у дизайні інтер’єрів. Технології доповненої реальності дозволяють [4]:

- візуалізувати меблі та декор у реальному масштабі, користувач бачить, як саме виглядатимуть об’єкти у його кімнаті;

- інтерактивну взаємодію, а саме можливість переміщувати, масштабувати та змінювати параметри об’єктів (колір, матеріали);

- оцінювати масштаб та пропорції об’єктів у приміщенні, покращувати ергономіку;

- експериментувати з перепланування без фізичних витрат, користувач може випробувати кілька варіантів планування без необхідності купувати меблі чи робити ремонт.

Для реалізації таких функцій використовуються сучасні бібліотеки та фреймворки: ARKit (Apple), ARCore (Google), Vuforia, AR Foundation (Unity). Вони забезпечують відстеження площин, розпізнавання оточення, накладання 3D-об’єктів у реальному масштабі та їх інтерактивне маніпулювання.

Кожна технологія має свої особливості:

- ARKit орієнтований на пристрої iOS і забезпечує високу точність відстеження завдяки датчикам LiDAR;

- ARCore працює на Android, дозволяючи реалізувати функціонал розпізнавання площин і трекінгу без додаткових сенсорів;

- Vuforia підтримує мультиплатформеність і розпізнавання маркерів (зображень, QR-кодів);

- AR Foundation в Unity надає універсальний інтерфейс для роботи з ARKit та ARCore, що дозволяє створювати кросплатформенні AR-застосунки.

Таким чином, використання AR у сфері дизайну інтер’єру надає новий рівень інтерактивності й наочності. Вибір конкретної технології залежить від цілей проєкту, цільової платформи та ресурсів користувача.

1.3 Аналіз аналогів використання технологій доповненої реальності

Для розуміння сучасного рівня розвитку технологій AR у сфері дизайну інтер’єрів проведено аналіз існуючих аналогів. У рамках цього аналізу розглянуто найбільш відомі мобільні застосунки, що реалізують візуалізацію меблів та елементів інтер’єру у режимі доповненої реальності.

Рисунок 1.1 – IKEA Place

AR застосунок: IKEA Place.

Платформа: iOS, Android.

Фреймворк: ARKit (iOS), ARCore (Android).

IKEA Place [5] – це мобільний застосунок на основі технології доповненої реальності (AR), розроблений шведською меблевою компанією IKEA для візуалізації меблів у реальному просторі користувача (рис. 1.1). Застосунок працює на платформі Apple ARKit і дозволяє користувачам «приміряти» меблі у своїх кімнатах перед покупкою. Завдяки технологіям комп’ютерного зору та точному масштабуванню, IKEA Place допомагає краще оцінити вигляд та відповідність меблів інтер’єру.

Функціональність: IKEA Place забезпечує можливість розмістити віртуальні 3D-моделі меблів у реальному приміщенні за допомогою камери смартфона. На момент запуску бібліотека застосунку містила понад 2000 предметів: крісла, столи, дивани та інші вироби з каталогу IKEA, причому база товарів постійно оновлюється. Застосунок автоматично масштабує моделі з точністю до 98%, враховує габарити кімнати, освітлення та текстуру об’єктів, що забезпечує реалістичність відображення.

Підхід: у роботі застосунку використовується поєднання доповненої реальності та засобів тривимірного моделювання. IKEA Place орієнтований на інтерактивну взаємодію: користувач може змінювати розташування меблів, обертати їх під будь яким кутом, оцінювати відповідність стилю та кольорової гами. Такий підхід дозволяє людям швидко приймати рішення щодо покупки, знижуючи невпевненість, що часто виникає при дистанційному виборі меблів.

Ефективність: практичні результати використання IKEA Place демонструють значне підвищення задоволеності користувачів процесом вибору меблів. Завдяки точному масштабуванню та реалістичному візуальному представленню, зменшується кількість помилкових покупок та повернень. Компанія IKEA відзначає, що застосунок підвищує залученість клієнтів та сприяє більш усвідомленому прийняттю рішень.

Переваги: проста інтеграція з онлайн магазином, велика база моделей меблів, зручність у використанні, реалістичне відтворення меблів у просторі, висока точність масштабування (приблизно 98%), інтерактивність та можливість оцінити дизайн у контексті власного інтер’єру, велика база моделей, що регулярно оновлюється, зрозумілий інтерфейс.

Недоліки: обмежена бібліотека лише товарами IKEA, відсутність функцій складного перепланування чи моделювання кімнати, потреба у добре освітленому приміщенні для коректної роботи, іноді можливі неточності при визначенні площини або відтворенні тіней.

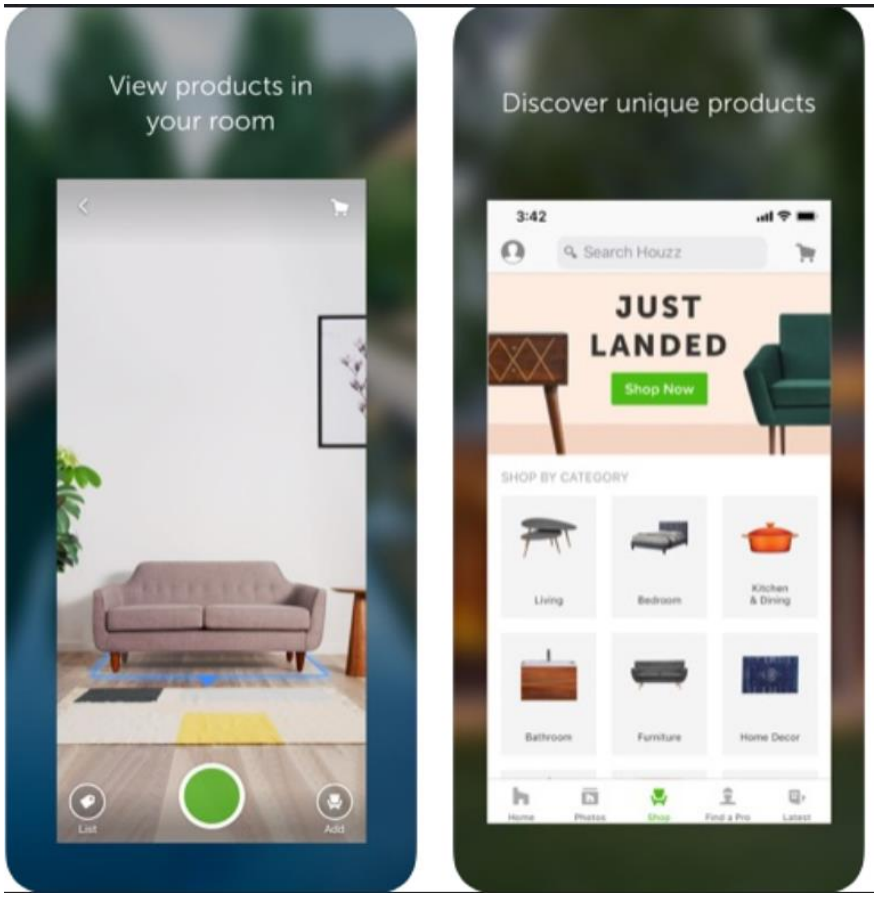

Рисунок 1.2 – Houzz

AR застосунок: Houzz.

Платформа: iOS, Android.

Фреймворк: ARCore/ARKit.

Houzz [6] – це комплексний мобільний застосунок і онлайн-платформа для планування, ремонту та дизайну житлових приміщень (рис. 1.2). Застосунок поєднує у собі каталог ідей, інструментів доповненої реальності, маркетплейс товарів для дому та сервіс для пошуку професіоналів у сфері будівництва й дизайну. Houzz орієнтований на власників осель, дизайнерів, підрядників та всіх, хто працює над створенням або оновленням житлових просторів.

Функціональність: Houzz надає доступ до великої бібліотеки з понад 25 мільйонів фотографій інтер’єрів та екстер’єрів у високій роздільності. Користувач може фільтрувати зображення за стилем, приміщенням або місцем розташування, зберігати їх у власні колекції та ділитися з професіоналами чи родиною. У застосунку реалізовано інструмент Sketch, який дозволяє залишати коментарі, робити помітки та малюнки безпосередньо на фотографіях. Houzz може влючати маркетплейс із понад 5 мільйонів товарів для дому. Технологія Visual Match дозволяє користувачу знаходити товари прямо із світлин, а View in My Room 3D дозволяє користувачу відобразити обрані предмети у своїй квартирі за допомогою AR.

Підхід: Houzz застосовує комплексну модель взаємодії з користувачем: поєднання натхнення, інструментів планування та прямої комерційної взаємодії. Інтерфейс розроблений так, щоб забезпечити глибоке занурення у проєкти: від пошуку ідей до реалізації через покупку товарів або співпрацю з професіоналом. Технології комп’ютерного зору та AR-моделювання дозволяють користувачам оцінити предмети інтер’єру у власному просторі, підвищуючи точність рішень щодо покупки.

Ефективність: Houzz визнано одним із найвпливовіших інструментів для дизайну та ремонту житла. Застосунок очолив рейтинг «найкращих програм для благоустрою дому» за версією The New York Times, отримав відгук The Washington Post як «єдине найкраще джерело натхнення», а CNN назвала його «Вікіпедією дизайну інтер’єру та екстер’єру». Така висока оцінка

пояснюється поєднанням багатого контенту, зручних інструментів візуалізації та широкої професійної бази.

Переваги: широка база товарів від різних виробників, інтеграція з магазином, інтерактивні інструменти (Sketch, Visual Match, View in My Room 3D), можливість найняти професіоналів прямо зі застосунку, наявність освітніх та інспіраційних матеріалів (статті, відео, поради спільноти), визнання та довіра у міжнародних ЗМІ.

Недоліки: складність інтерфейсу, орієнтація більше на покупки, ніж на детальне планування, частина функцій (особливо AR) працює не на всіх пристроях або потребує сучасних технічних характеристик, наявність платних товарів і послуг, що може бути недоступним для певної частини користувачів, функціональність орієнтована переважно на ринок США та ЄС, що інколи призводить до обмеженої доступності локальних товарів чи спеціалістів.

Рисунок 1.3 – Planner 5D

AR застосунок: Planner 5D.

Платформа: iOS, Android, Web.

Фреймворк: власний 3D-рушій з інтеграцією AR.

Planner 5D [7]**:** це багатофункціональний мобільний і вебзастосунок для створення планів поверхів та дизайну інтер’єрів у 2D та 3D форматах (рис. 1.3). Він орієнтований як на професійних дизайнерів, так і на користувачів без

спеціальних навичок, забезпечуючи інструменти для моделювання, реконструкції та візуалізації житлових і комерційних просторів. Planner 5D поєднує простоту використання з широкими можливостями деталізації, дозволяючи створювати реалістичні проєкти та зображення інтер’єру.

Функціональність: застосунок надає доступ до великого каталогу з понад 6700 елементів інтер’єру та екстер’єру – меблів, декору, побутових предметів, освітлення, садових об’єктів тощо. Користувач може створювати плани поверхів, змінювати розташування стін, дверей і вікон, оформлювати кімнати відповідно до свого стилю та вподобань. Planner 5D підтримує як 2D-редагування, так і повноцінну 3D-візуалізацію проєктів. Застосунок також містить інструменти AR-візуалізації, що дозволяють «вмонтовувати» створені моделі в реальний простір за допомогою камери смартфона, оцінюючи їх у натуральному масштабі.

Підхід: Planner 5D заснований на моделі інтуїтивного проєктування, що дозволяє користувачам швидко створювати складні планування без необхідності у професійних знаннях CAD-систем. В основі лежить поєднання drag-and-drop інтерфейсу, реалістичного 3D-рендерингу та AR-візуалізації, які разом забезпечують простий і водночас гнучкий процес розробки інтер’єрів. Застосунок дає можливість працювати з широкою варіативністю стилів і конфігурацій, пропонуючи інструменти для реконфігурації простору, декорування та створення авторських рішень. Його екосистема включає активну спільноту користувачів, що ділиться досвідом, ідеями та готовими проєктами, а також проводить тематичні конкурси.

Ефективність: Planner 5D є одним із найпопулярніших інструментів для домашнього дизайну, оскільки поєднує професійні можливості зі зручністю для новачків. AR-інструменти та високореалістичні знімки дозволяють користувачам точніше оцінити майбутній вигляд приміщення, що сприяє зменшенню помилок у плануванні та ремонті. Застосунок широко використовують для підготовки концепцій перед реалізацією, моделювання

кількох варіантів планування, презентацій клієнтам та візуалізації стилістичних рішень. Завдяки мультимовності й доступності на різних платформах, він має широке міжнародне поширення.

Переваги: можливість комплексного проєктування приміщення, створення кількох варіантів планування, можливість працювати як онлайн, так і офлайн, кросплатформна синхронізація через особистий акаунт, підтримка багатьох мов інтерфейсу, великий каталог об’єктів для будь-яких типів приміщень.

Недоліки: потребує певних навичок від користувача, менш зручний для швидких експериментів, частина функціоналу доступна лише у платній версії, AR-функції потребують сучасних пристроїв, високореалістичний рендеринг може бути ресурсоємним.

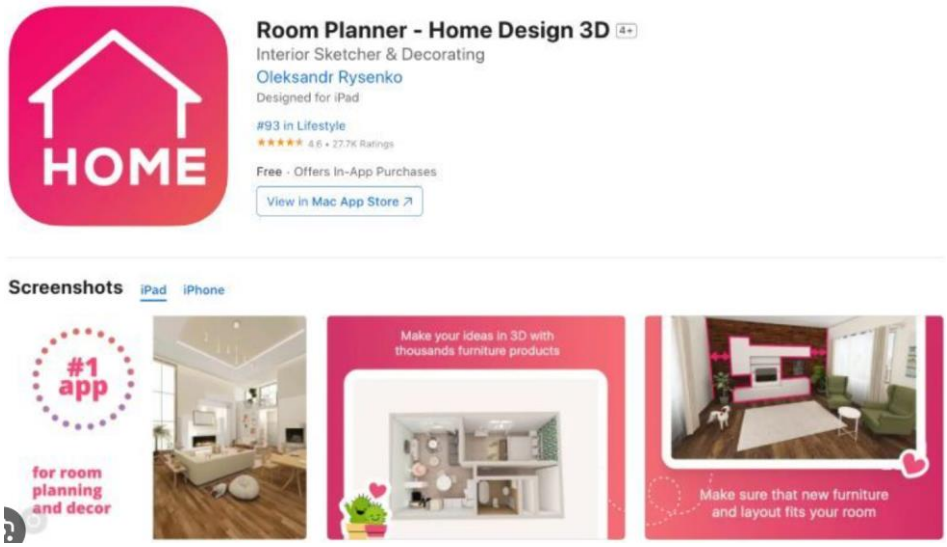

Рисунок 1.4 – Room Planner

AR застосунок: Room Planner.

Платформа: iOS, Android.

Фреймворк: ARCore/ARKit.

Room Planner [8]: це мобільний застосунок для 3D-планування приміщень та дизайну інтер’єру (рис. 1.4). Він дозволяє створювати візуалізації кімнат, планувати ремонт, підбирати меблі та декор, а також

розробляти власні дизайнерські рішення без потреби в архітектурній чи професійній підготовці. Room Planner орієнтований на власників квартир і будинків, дизайнерів-початківців та всіх, хто працює над облаштуванням житлових чи офісних просторів.

Функціональність: Room Planner пропонує широкий набір інструментів для проєктування простору. Користувач може обирати оздоблювальні матеріали, меблі, кольорові схеми та декоративні елементи з великого каталогу товарів – понад 150 000 позицій від популярних брендів та близько 5000 предметів у відкритому доступі. Застосунок дозволяє створювати кімнати будь якої конфігурації, оновлювати планування, розміщувати меблі та оцінювати результат у 3D-режимі. Підтримуються готові професійні проєкти, які можна використовувати як базу для власних рішень. Користувач також може ділитися своїми дизайнами через інтернет, що полегшує спільне планування та обговорення.

Підхід: Room Planner використовує інтуїтивну модель дизайну, яка базується на принципах drag-and-drop, реалістичної 3D-візуалізації та швидкого налаштування параметрів приміщення. Застосунок дозволяє працювати з інтер’єрами без складних технічних інструментів, тому процес створення проєкту доступний навіть новачкам. Платформа орієнтована на адаптивність – користувач може обирати готові рішення або створювати власний дизайн «з нуля». Особливий акцент робиться на поєднанні практичності (розрахунок матеріалів, моделювання перестановок) та естетики (каталог меблів, декор, стилістичні добірки).

Ефективність: Застосунок Room Planner значно спрощує процес планування ремонту та перепланування житлових приміщень. Тривимірна візуалізація дозволяє краще зрозуміти пропорції, взаємне розташування меблів та загальний стиль інтер’єру, що зменшує ймовірність помилок на етапі реалізації. Готові професійні проєкти та велика база товарів сприяють пришвидшенню процесу прийняття рішень. Завдяки можливості обміну

проєктами застосунок є зручним інструментом для спільної роботи над дизайном.

Переваги: інтуїтивний інтерфейс, підтримка роботи в AR і в класичному 3D, детальний каталог товарів і меблів (понад 150 000 позицій), широкі можливості кастомізації інтер’єрів, доступ до готових дизайнерських рішень, функція обміну проєктами через інтернет, можливість розраховувати матеріали для ремонту.

Недоліки: значна частина розширених функцій доступна тільки у PRO-підписці, робота з великими 3D-проєктами вимагає продуктивних пристроїв.

Аналіз аналогів показує, що на ринку вже існують успішні рішення, однак більшість із них або жорстко прив’язані до конкретних виробників меблів (IKEA Place, Houzz), або орієнтовані на більш професійне 3D-проєктування (Planner 5D). Це створює нішу для застосунку, який поєднуватиме простоту використання, широку бібліотеку моделей і можливості повноцінного планування простору.

1.4 Визначення цільової аудиторії

Застосунки для планування інтер’єру на основі технологій доповненої реальності орієнтовані на широку аудиторію користувачів, які прагнуть оптимізувати процес ремонту, перепланування та оформлення свого житлового простору. Такі рішення роблять процес дизайну доступним, наочним і менш ризиковим, дозволяючи користувачам бачити результат ще до реальної реалізації.

Технології AR значно спрощують візуалізацію меблів, кольорових схем і декоративних елементів у конкретному приміщенні, що робить їх особливо корисними для тих, хто не має професійної дизайнерської підготовки. Застосунок задовольняє потреби людей із різним рівнем технічної грамотності, стилістичних вподобань та фінансових можливостей. Основні сегменти цільової аудиторії можна визначити так [9]:

1) власники квартир і будинків – це найбільший сегмент користувачів, які планують ремонт, перепланування або оновлення інтер’єру. AR-застосунки дозволяють їм експериментувати з розташуванням меблів, обирати оздоблювальні матеріали, оцінювати поєднання кольорів та стилів без ризику додаткових витрат. Такі користувачі цінують простоту, наочність і можливість самостійно здійснювати дизайн-процеси;

2) молоді люди, що облаштовують житло вперше – студенти, молоді спеціалісти та пари, які переїжджають у власне або орендоване житло, прагнуть оптимізувати бюджет і уникнути помилок на етапі ремонту. Застосунок із AR допомагає їм знаходити стильові рішення, підбирати меблі, експериментувати з інтер’єром навіть без професійних навичок, що робить такі інструменти особливо привабливими;

3) дизайнери-початківці та фрилансери – користувачі, які працюють у сфері дизайну, але не мають доступу до складних або дорогих CAD-програм, можуть використовувати AR-застосунок для швидких ескізів, презентацій клієнтам, візуалізації концепцій та експериментів з простором. Для них важлива точність масштабування та можливість інтерактивної демонстрації об’єктів;

4) професійні дизайнери та архітектори – хоч вони мають доступ до професійних програм, AR-функції можуть служити додатковим інструментом для швидкого показу ідей клієнтам, демонстрації концепцій у реальному просторі та підвищення якості комунікації. Застосунок може пришвидшити первинний етап проєктування та скоротити кількість поправок;

5) люди, які роблять дрібний ремонт або планують перестановку – користувачі, які не хочуть проводити масштабний ремонт, але прагнуть освіжити або оптимізувати приміщення. AR-застосунок дозволяє швидко оцінити, як виглядатиме нове крісло, шафа чи колір стіни у кімнаті, що спрощує прийняття рішень та економить час;

6) онлайн покупці меблів та декору – користувачі, які купують меблі через інтернет, можуть використовувати AR-візуалізацію, щоб побачити, як об’єкт виглядатиме у реальній кімнаті. Це знижує ймовірність повернень і розчарувань, робить вибір більш усвідомленим та допомагає заощадити гроші.

1.4 Unity3D у якості рушія для розробки застосунку доповненої реальності

Unity 3D є одним із найбільш популярних і універсальних рушіїв для створення інтерактивних тривимірних застосунків, що робить його особливо придатним для розробки систем доповненої реальності. Поєднання візуального редактора, широких графічних можливостей, потужної системи компонентів та підтримки мобільних платформ перетворює Unity на гнучке середовище, у якому можна реалізувати як високорівневі концепції дизайну інтер’єру, так і складні AR-функції.

Unity працює на основі компонентної архітектури: кожен об’єкт у сцені може мати довільний набір компонентів, що відповідають за його поведінку, фізику, візуалізацію або взаємодію з оточенням. Такий підхід дає змогу легко розширювати функціональність системи, повторно використовувати елементи, структурувати об’єкти у вигляді prefab-ів і забезпечувати гнучкість у створенні тривимірних інтерфейсів. Використання сцен як окремих логічних модулів дає можливість розділяти застосунок на незалежні частини: екрани меню, простір AR-візуалізації, галереї проєктів або інші модулі.

Однією з ключових переваг Unity є підтримка різноманітних форматів 3D-моделей і матеріалів. Рушій повністю сумісний із поширеними форматами FBX, OBJ, GLTF та іншими, що дає можливість імпортувати моделі меблів, інтер’єрних елементів чи планувальних конструкцій з професійних програм, таких як Blender, 3ds Max або SketchUp. Завдяки фізично коректному рендерингу (PBR) Unity дозволяє досягти високого рівня реалізму матеріалів, що є важливою вимогою для застосунків у сфері дизайну інтер’єру. Система

шейдерів надає можливість адаптувати зовнішній вигляд об’єктів під умови освітлення, які будуть отримані з AR-платформи.

Unity має інтегровані засоби для побудови користувацького інтерфейсу, що дозволяє організувати інтуїтивну взаємодію з AR-сценою: переміщення та масштабування об’єктів, вибір елементів з каталогу, навігацію між розділами. UI-система працює незалежно від 3D-сцени, що забезпечує стабільність елементів інтерфейсу та їх коректну роботу на різних розмірах екранів мобільних пристроїв.

Важливою перевагою Unity є наявність великої екосистеми асетів та бібліотек, які можна інтегрувати у проєкт. Asset Store пропонує моделі меблів, текстури, шейдери, UI-компоненти та готові інструменти для візуалізації. Така екосистема суттєво скорочує час розробки і дозволяє зосередитися на реалізації унікальних можливостей AR-застосунку.

Unity також забезпечує підтримку AR-технологій через модуль AR Foundation, який дозволяє працювати з ARKit та ARCore з єдиною базою коду. Це означає, що розробнику не потрібно створювати окремі реалізації для Android і iOS – Unity автоматично адаптує функціональність під конкретну платформу. Такий підхід спрощує розробку, знижує витрати часу та забезпечує високу продуктивність на різних пристроях [10].

Разом із значними перевагами Unity має і певні обмеження. Для досягнення плавної роботи AR-застосунку необхідно оптимізувати 3D-моделі, скорочувати кількість полігонів, регулювати розміри текстур і контролювати кількість одночасно відображуваних об’єктів. Також важливо враховувати різні графічні можливості пристроїв на Android, що потребує додаткового тестування та адаптацій. Незважаючи на ці обмеження, рушій залишається одним із найефективніших рішень для розробки кросплатформних застосунків доповненої реальності.

Таким чином, Unity 3D є оптимальним рушієм для створення програмного забезпечення з AR-функціональністю у сфері інтер’єрного

дизайну. Він поєднує у собі кросплатформність, високий рівень реалістичності візуалізації, зручні інструменти для управління 3D-контентом і розвинуту екосистему додаткових ресурсів. Ці особливості роблять Unity не лише технічно придатним, а й стратегічно вигідним вибором для реалізації AR-застосунку.

Висновки до розділу 1

Наведений аналіз предметної області та існуючих рішень показав, що технології доповненої реальності посідають важливе місце в сучасних програмних системах і успішно застосовуються в різних галузях – від розваг, освіти й медицини до промисловості, логістики та фінансів. Окремо розглянуто специфіку використання AR у дизайні інтер’єру, де ця технологія дає змогу поєднати візуалізацію в реальному часі, просторове вимірювання, інтерактивну взаємодію з об’єктами та аналітичні можливості для оцінки ергономіки й композиції приміщення. Виділено основні групи методів (візуалізації, побудови кімнати, інтерактивності, оптимізації та комбіновані підходи), що формують теоретичну основу для проєктування системи планування інтер’єру з використанням AR.

Аналіз існуючих застосунків IKEA Place, Houzz, Planner 5D та Room Planner показав, що на ринку вже є низка зрілих рішень, які демонструють ефективність AR у задачах візуалізації меблів і планування простору. Водночас виявлено їхні обмеження: прив’язка до конкретних брендів, орієнтація переважно на маркетплейс або, навпаки, на професійне 3D-проєктування з підвищеним порогом входу. Це створює нішу для застосунку, який поєднуватиме простоту використання, достатню глибину просторового моделювання та широку бібліотеку об’єктів без жорсткої залежності від одного виробника.

Визначено основні сегменти цільової аудиторії: власники квартир і будинків, молоді люди, що вперше облаштовують житло, дизайнери-

початківці та фрилансери, професійні дизайнери й архітектори, користувачі, які планують дрібний ремонт або перестановку, а також онлайн покупці меблів і декору. Для всіх цих груп застосунок із доповненою реальністю вирішує спільну проблему – знижує ризики помилкового вибору, робить процес планування наочним і дає змогу експериментувати з інтер’єром без фінансових витрат на реальні зміни.

У якості програмного рушія для майбутнього застосунку обрано Unity 3D, що забезпечує кросплатформність, потужні засоби 3D-візуалізації, гнучку систему компонентів, підтримку фізично коректного рендерингу та інтеграцію з AR Foundation для роботи з ARKit та ARCore. Це дозволяє реалізувати єдину кодову базу для Android та iOS, ефективно працювати з 3D-контентом і скоротити час розробки завдяки використанню готових ресурсів з екосистеми Unity.

Результати аналізу доводять актуальність теми кваліфікаційної роботи та обґрунтовують вибір технологічного стеку. Визначені особливості предметної області, наявні аналоги, цільова аудиторія й обраний рушій створюють необхідне підґрунтя для подальшого формування вимог до системи, її проєктування та програмної реалізації.

2 АНАЛІЗ ТА МОДЕЛЮВАННЯ ВИМОГ

2.1 Аналіз алгоритмів комп’ютерного зору та AR-технологій для просторового моделювання

Алгоритми комп’ютерного зору є основою сучасних систем доповненої реальності, забезпечуючи коректне відображення та позиціонування віртуальних об’єктів у реальному просторі. У контексті мобільних AR-застосунків для планування інтер’єру їх основним завданням є точне визначення геометрії приміщення, відстеження руху камери, формування 3D-моделі сцени та узгодження цифрових об’єктів із фізичним середовищем. Робота таких систем базується на комплексі алгоритмів, серед яких ключове місце займають SLAM, визначення площин, трекінг руху, оцінка глибини та аналіз освітлення.

SLAM – це процес одночасного визначення положення камери в просторі та побудови карти середовища. У мобільних AR-застосунках SLAM виконується в режимі реального часу та включає такі етапи [11]:

1) виявлення особливостей (feature detection) – система ідентифікує характерні точки на зображенні, такі як кути, лінії, контури меблів;

2) відстеження точок між кадрами (feature tracking) – визначає, як переміщуються ці точки від кадру до кадру, вимірюючи рух камери;

3) оцінка позиції пристрою (pose estimation) – система обчислює просторове положення та орієнтацію камери у 3D;

4) оновлення карти простору – формується хмара точок та реконструюється геометрична структура сцени.

SLAM дозволяє AR-застосунку «орієнтуватися» в кімнаті та зберігати стабільне положення 3D-об'єктів незалежно від переміщення користувача. Це критично важливо для коректної візуалізації меблів та інтер’єрних елементів.

Для відображення віртуальних меблів у реальному просторі система повинна виявляти ключові структурні елементи кімнати (підлогу, стіни, стелі,

горизонти та вертикальні поверхні). Алгоритми виявлення площин працюють за принципом кластеризації точок хмари, отриманої SLAM. Платформи ARKit та ARCore використовують такі підходи:

1) RANSAC – для виділення рівних площин;
2) аналіз нормалей – для визначення орієнтації поверхні;
3) оцінка розміру й меж – система визначає не лише наявність площини, а й її контур.

Після цього будується спрощена 3D-модель кімнати, яка використовується для точного розміщення віртуальних об’єктів. Наприклад, AR-застосунок може визначити, де саме на підлозі можна розташувати диван, а на стіні – навісну полицю.

Реалістична інтеграція тривимірних об’єктів у сцену потребує точного визначення глибини для кожної точки зображення, оскільки саме ця інформація дозволяє системі коректно відтворювати взаємодію між віртуальними та реальними елементами. Сучасні AR-платформи застосовують низку підходів до оцінки глибини, серед яких використання глибинних карт, отриманих безпосередньо з камери, сенсорів LiDAR на новіших моделях пристроїв Apple, технологій структурованого світла на старіших версіях техніки, а також методів стереозору, що ґрунтуються на порівнянні послідовних кадрів. Інформація про глибину забезпечує низку ключових можливостей для AR-систем: правильне масштабування тривимірних моделей відповідно до фізичних розмірів простору, коректне перекривання реальних та віртуальних об’єктів, відтворення реалістичного освітлення та тіней, а також виявлення колізій між об’єктами в сцені. У контексті застосунків для планування інтер’єру ці дані є особливо важливими, оскільки вони дозволяють точно оцінювати, чи поміститься вибраний предмет меблів у конкретному місці, та гарантують відповідність між цифровою моделлю й реальними параметрами приміщення.

Без трекінгу руху камери важко уявити собі AR застосунок, тому описуючи його принцип, можна виділити його базові комбінування, а саме візуальний трекінг через камеру, інерційних датчиків (акселерометр, гіроскоп) та IMU-модулів. Використання даних із різних джерел компенсує недоліки кожного окремого методу та забезпечує плавний і стійкий трекінг. Це дозволяє стабільно утримувати 3D-моделі у просторі навіть при швидких рухах смартфона [12].

Реалістичність AR-сцени значною мірою залежить від того, наскільки точно система визначає характер освітлення у приміщенні та адаптує до нього віртуальні об’єкти. Платформи ARKit і ARCore застосовують комплексний підхід до аналізу світлового середовища, поєднуючи оцінку яскравості з RGB-кадрів, визначення інтенсивності й напрямку падаючого світла, аналіз глобального освітлення та використання HDR-фонових кубів для моделювання складних світлових умов. Отримана інформація дозволяє коректно регулювати інтенсивність і колір освітлення на тривимірних моделях, забезпечувати реалістичну симуляцію тіней та точно відтворювати фізично коректні матеріали на основі PBR-підходу. Завдяки цьому меблі й інші віртуальні об’єкти у сцені виглядають природно та гармонійно вписуються в реальний інтер’єр, максимально наближаючись до їхнього справжнього вигляду в умовах конкретного освітлення приміщення.

Незважаючи на значний прогрес, мобільні AR-технології мають низку обмежень, які впливають на точність просторового моделювання:

1) drift (похибка позиціонування) – з часом система SLAM може накопичувати помилки, через що 3D-об'єкти можуть зміщуватися;

2) нестабільне освітлення – слабке, мерехтливе або надто яскраве світло погіршує роботу алгоритмів розпізнавання;

3) однорідні поверхні – стелі, однотонні стіни, гладкі підлоги без текстури важко аналізувати – немає особливостей для SLAM-трекінгу;

4) обмеження продуктивності – мобільні пристрої мають значні обмеження на обчислювальні ресурси, що впливає на якість рендерингу складних сцен;

5) вимоги до сучасних сенсорів – старі моделі смартфонів не підтримують деякі функції, такі як LiDAR або якісні карти глибини.

2.2 Аналіз моделей та інструментів візуалізації 3D-об’єктів

Візуалізація тривимірних об’єктів є ключовим компонентом AR-застосунків, призначених для планування інтер’єру, оскільки саме якість моделей визначає реалістичність сцени, точність сприйняття меблів і коректність взаємодії користувача з цифровими елементами. У сучасних системах доповненої реальності використовуються різні формати 3D-моделей, серед яких найбільш поширеними є GLTF, FBX та OBJ. Формат GLTF вважається оптимізованим для реального часу завдяки компактності, підтримці PBR-матеріалів та швидкому завантаженню, тому часто використовується у WebAR та мобільних AR-рішеннях. FBX є гнучким і підтримує анімації та складні матеріали, проте займає більше місця та потребує додаткової конвертації. OBJ є простим форматом для геометрії, але не містить розширених властивостей матеріалів, що обмежує його застосування у високореалістичних сценах [13].

Однією з важливих характеристик 3D-моделей у мобільних застосунках є рівень деталізації (LOD). Мобільні пристрої мають обмежені ресурси графічної обробки, тому надмірно деталізовані моделі можуть призвести до зниження продуктивності. Технологія LOD дає змогу автоматично зменшувати деталізацію об’єкта залежно від його відстані до камери, зберігаючи при цьому загальну якість сцени та стабільний FPS. Оптимізація геометрії – зокрема зменшення кількості полігонів, використання нормалей замість дрібної геометрії та оптимізація UV-розгорток – є обов’язковим етапом підготовки моделей для AR.

Реалістичність відображення значною мірою залежить від матеріалів і текстур. Сучасні рушії, такі як Unity, підтримують рендеринг, заснований на фізичних властивостях (PBR), що враховує взаємодію світла з поверхнею, надаючи об’єктам природний вигляд. PBR-матеріали використовують карти альбедо, нормалей, металевості, шорсткості та інколи карту оточення. У контексті AR-інтер’єрів це дозволяє досягти реалістичної візуалізації дерева, тканини, металу, пластику чи скла. Разом із тим застосування високоякісних текстур може підвищити навантаження на графічний процесор, тому важливо дотримуватися оптимальних розмірів текстур та компресії.

Мобільна графіка має низку обмежень, пов’язаних із потужністю GPU, обсягом оперативної пам’яті та особливостями енергоспоживання. Рендеринг складних сцен або використання високополігональних моделей може призвести до падіння продуктивності, перегрівання пристрою або затримок у відображенні. Тому створення AR-застосунку потребує балансу між реалістичністю моделей та їхньою оптимальністю для виконання в реальному часі.

Вибір підходу до отримання 3D-моделей залежить від вимог системи та доступних ресурсів. Ручне створення моделей забезпечує максимальний контроль над якістю, стилістикою та оптимізацією, однак потребує значних витрат часу та досвіду 3D-художника. Імпорт моделей із бібліотек, таких як Sketchfab, CGTrader або внутрішні каталоги виробників меблів, дозволяє значно прискорити процес розробки, але вимагає додаткової обробки та оптимізації. Процедурне створення елементів інтер’єру, наприклад стін, підлоги або простих геометричних об’єктів, є ефективним у випадках, коли потрібно автоматично генерувати приміщення або адаптувати його параметри під реальну кімнату.

Зважаючи на те, що необхідно забезпечити реалістичніть разом з оптимізацією, оптимальним рішенням буде комбінований бідіх, у якому:

1) базові елементи інтер’єру (стіни, підлога, базова геометрія простору) створюються процедурно;

2) меблі та декор імпортуються з бібліотек готових моделей, після чого проходять оптимізацію;

3) усі моделі конвертуються у формат FBX як найбільш ефективний для мобільної AR-візуалізації;

4) використання PBR-матеріалів та текстур оптимізується під можливості мобільних пристроїв.

Це забезпечить найкращу оптимізацію.

2.3 Аналіз доступних фреймворків і бібліотек для розробки AR-застосунків

Технології доповненої реальності (AR) відіграють ключову роль у створенні сучасних мобільних застосунків для проєктування інтер’єру та візуалізації меблів у реальному просторі. На відміну від традиційних 3D-планувальників, AR-системи дозволяють інтегрувати цифрові моделі в середовище користувача з урахуванням масштабу, освітлення та геометрії приміщення. Найбільш поширеними платформами для впровадження AR-функціональності в мобільних застосунках є ARKit (iOS), ARCore (Android) та AR Foundation, універсальний фреймворк від Unity, що об’єднує можливості двох платформ (рис. 2.1).

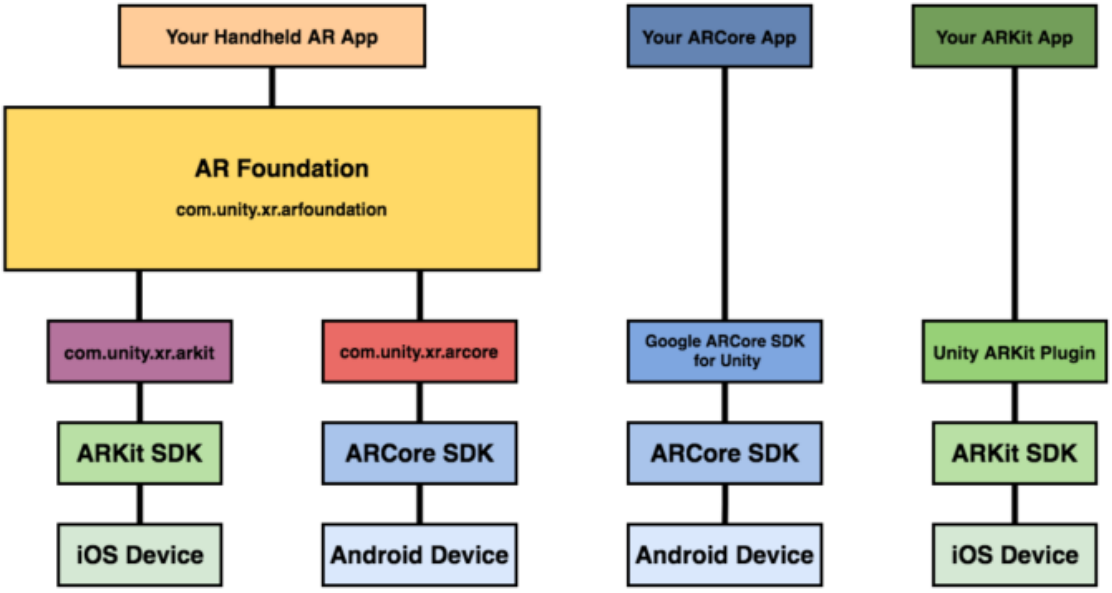

Рисунок 2.1 – Схема ARKit, ARCore та AR Foundation

ARKit – це фреймворк доповненої реальності, розроблений компанією Apple. Він забезпечує високоточне відстеження просторового положення пристрою завдяки механізмам SLAM, сенсору глибини та системам оцінки освітлення. ARKit підтримує розпізнавання горизонтальних і вертикальних площин, реконструкцію сцени, виявлення фізичних об’єктів та роботу з LiDAR-сканером на сучасних пристроях Apple. Завдяки цьому технологія дозволяє точно визначати розміри приміщення, поміщати тривимірні моделі меблів у натуральному масштабі та забезпечувати їх реалістичну інтеграцію в інтер’єр [14].

ARCore – аналогічна технологія від Google, використовується на пристроях Android. Платформа підтримує визначення площин, відстеження руху камери, оцінку освітлення та розпізнавання геометрії простору. Система працює на основі трекінгу особливостей поверхонь, що дозволяє стабільно розміщувати 3D-об’єкти у сцені навіть за умов недостатньої текстурованості. ARCore також підтримує функцію Depth API, що забезпечує коректне перекривання цифрових та реальних об’єктів, покращуючи фотореалістичність композиції [15].

Важливою перевагою сучасних AR-систем є наявність фреймворку AR Foundation, розробленого компанією Unity. Він діє як універсальний шар абстракції, що дозволяє створювати єдиний AR-застосунок одночасно для Android та iOS. AR Foundation автоматично взаємодіє з ARKit і ARCore, забезпечуючи доступ до всіх ключових функцій: визначення площин, відстеження руху, виявлення точок, роботу з глибиною, розпізнавання освітлення та розміщення цифрових об’єктів у реальному просторі. Такий підхід значно спрощує розробку AR-рішень та зменшує кількість платформозалежного коду [16].

ARKit, ARCore та AR Foundation надають інструменти для високоякісної інтеграції 3D-моделей у реальний простір, що робить їх основою сучасних додатків для ремонту та планування житла. Вони мають такі переваги як:

1) висока точність трекінгу – сучасні пристрої забезпечують стабільне визначення площин і позиціонування моделей;

2) реалістична інтеграція об’єктів – оцінка освітлення та глибини дозволяє моделювати природне взаємодоповнення між віртуальним і реальним середовищем;

3) кросплатформність – завдяки AR Foundation один застосунок може підтримувати Android та iOS без дублювання логіки;

4) простота розробки – Unity надає інструменти для зручної роботи зі сценою, 3D-об’єктами та UI, що зменшує затрати на створення AR-застосунків.

Однак, також є і недоліки:

1) високі вимоги до апаратного забезпечення – точна робота AR можлива лише на сучасних смартфонах із відповідними сенсорами;

2) чутливість до умов освітлення – слабке або занадто яскраве світло погіршує якість трекінгу;

3) обмеження продуктивності – рендеринг деталізованих 3D-моделей може впливати на FPS на малопотужних пристроях;

4) залежність від камери й сенсорів – некоректна робота камери або слабкий текстурний фон зменшують точність побудови сцени.

Таким чином, ARKit, ARCore та AR Foundation становлять технологічну основу для створення мобільного застосунку доповненої реальності, орієнтованого на планування інтер’єру. Вони забезпечують широкий спектр можливостей для точного позиціонування об’єктів, реалістичної візуалізації, кросплатформної розробки та високої якості користувацького досвіду, що робить їх оптимальним вибором для реалізації програмного продукту такого типу.

2.4 Специфікація вимог до AR-застосунку для планування інтер’єру

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення застосунку, для якої розробляється програмне забезпечення

Розроблюваний застосунок призначений для візуалізації меблів та елементів інтер’єру у режимі доповненої реальності з метою спрощення процесу планування житлового простору.

Погодження, що ухвалені в програмній документації

Погоджено, що для створення застосунку будуть використані сучасні фреймворки (Unity 3D, AR Foundation, ARCore/ARKit) та допоміжні бібліотеки для роботи з 3D-моделями.

Межі проєкту ПЗ

Проєкт орієнтований на мобільні пристрої Android та iOS. Кінцевим результатом є прототип AR-застосунку, що дозволяє розміщувати та редагувати віртуальні об’єкти у реальному просторі.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Застосунок може бути використаний для планування інтер’єру житлових приміщень, попередньої оцінки меблів перед купівлею, а також у роботі дизайнерів та рієлторів.

Характеристики користувачів

Користувачі не потребують спеціальних технічних знань. Основні вимоги – наявність мобільного пристрою з підтримкою AR.

Загальна структура і склад системи

Основні модулі застосунку:

- AR-модуль (відстеження площин, масштабування сцени, накладання об’єктів).
- Модуль взаємодії з 3D-моделями (переміщення, обертання, масштабування, зміна кольору).
- Бібліотека об’єктів (набір меблів та декору).
- Модуль збереження проєктів.

Загальні обмеження

Функціонал застосунку залежить від потужності пристрою та якості камери.

ФУНКЦІЇ СИСТЕМИ

1) Функція візуалізації меблів у приміщенні

Опис функції

Система накладає 3D-моделі меблів на зображення з камери у масштабі 1:1.

Вхідна інформація

Відеопотік із камери смартфона, дані сенсорів.

Вихідна інформація

AR-сцена з розміщеними об’єктами у реальному просторі.

Функціональні вимоги

відображення з точністю до 5 см, частота кадрів не нижче 30 FPS.

2) Функція взаємодії з об’єктами

Опис функції

Користувач може додавати, переміщати, масштабувати та обертати меблі.

Вхідна інформація

Жести користувача (tap, drag, pinch, rotate).

Вихідна інформація

Оновлене положення та параметри об’єкта у сцені.

Функціональні вимоги

Підтримка стандартних мобільних жестів, плавна зміна положення без затримок.

3) Функція зміни параметрів об’єкта

Опис функції

Зміна кольору та матеріалів об’єктів для підбору стилістики.

Вхідна інформація

Вибір кольору чи матеріалу з бібліотеки.

Вихідна інформація

Оновлений візуальний вигляд об’єкта.

Функціональні вимоги

Миттєве оновлення текстур без втрати продуктивності.

4) Функція збереження

Опис функції

Збереження сцени локально, можливість робити скріншоти та ділитися ними.

Вхідна інформація

Команда користувача «Зберегти».

Вихідна інформація

Файл проєкту у форматі JSON, скріншоти (PNG).

Функціональні вимоги

Автоматичне збереження при аварійному завершенні роботи застосунку.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

Відеопотік із камери, дії користувача, бібліотека 3D-моделей.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Стилістичні каталоги меблів, кольорові палітри.

Вимоги до способів організації, збереження та ведення інформації

Збереження даних відбувається через JSON файл. Ведення інформації відбувається під час виконання застосунку.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Пристрій із підтримкою ARCore (Android) або ARKit (iOS). Камера від 8 Мп, оперативна пам’ять від 4 ГБ.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура застосунку клієнтська (мобільний застосунок).

Системне програмне забезпечення

Android 10+, iOS 13+ із підтримкою AR.

Мережне програмне забезпечення

Використання ОС Windows 10.

Програмне забезпечення ведення інформаційної бази

Вимоги відсутні.

Мова і технологія розробки ПЗ

Unity 3D, AR Foundation, C#.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Простий і зрозумілий, з підтримкою жестів.

Апаратний інтерфейс

Камера, сенсори руху (IMU).

Програмний інтерфейс

API для роботи з ARCore/ARKit.

Комунікаційний протокол

Вимоги відсутні.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Робота без підключення до інтернету (базовий функціонал).

Супроводжуваність

Можливість розширення бібліотеки моделей.

Переносимість

Кросплатформенність (Android/iOS).

Продуктивність

Мати більше 30 FPS під час візуалізації.

Надійність

Автоматичне збереження при аварійному завершенні.

Безпека

Доступ лише до камери та пам’яті пристрою.

ІНШІ ВИМОГИ

- Підтримка мультимовності (українська, англійська).
- Можливість імпорту сторонніх 3D-моделей.

Висновки до розділу 2

У другому розділі проаналізовано технологічні засади, на яких ґрунтується реалізація мобільного AR-застосунку для планування інтер’єру, та сформовано вимоги до проєкту. Розгляд алгоритмів комп’ютерного зору показав, що коректна робота системи неможлива без поєднання SLAM, виявлення площин, оцінки глибини, трекінгу руху камери та аналізу освітлення. Саме ці компоненти забезпечують стабільне позиціонування

віртуальних об’єктів, реалістичну інтеграцію меблів у простір кімнати та коректне відтворення їхніх розмірів і візуальних властивостей. Водночас виявлено низку обмежень мобільних AR-систем (drift, залежність від освітлення, обмежена продуктивність, вимоги до сенсорів), які необхідно враховувати під час проєктування.

Аналіз форматів 3D-моделей, технік LOD, геометричної оптимізації та PBR-візуалізації засвідчив, що для досягнення балансу між реалістичністю та продуктивністю доцільно застосувати комбінований підхід: процедурно формувати базову геометрію приміщення, імпортувати меблі з готових бібліотек із подальшою оптимізацією моделей та використовувати фізично коректні матеріали з обмеженими розмірами текстур. Такий підхід мінімізує навантаження на мобільні пристрої й забезпечує достатню якість графіки.

Окремо розглянуто фреймворки ARKit, ARCore та AR Foundation, які становлять технологічну основу для реалізації AR-функцій. Їхній аналіз показав, що зв’язка Unity 3D + AR Foundation є оптимальним рішенням для кросплатформної розробки під Android та iOS, оскільки дозволяє використовувати спільний код і повноцінно задіяти можливості нативних AR-платформ. Визначені переваги (висока точність трекінгу, реалістична інтеграція об’єктів, кросплатформність) і недоліки (вимоги до технічних характеристик пристроїв, чутливість до умов зйомки) задають рамки для подальшого проєктування системи.

На основі проведеного аналізу сформульовано специфікацію вимог до розроблюваного AR-застосунку: визначено його призначення, межі проєкту, функціональні можливості (візуалізація меблів, взаємодія з об’єктами, зміна їх параметрів, збереження проєктів), а також нефункціональні вимоги до продуктивності, надійності, інтерфейсу та технічного забезпечення.

3 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ СИСТЕМИ ПЛАНУВАННЯ ІНТЕР’ЄРУ В РЕЖИМІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

3.1 Сценарії використання системи

Змоделюємо, для чого потрібна система візуального планування інтер’єру з використанням AR і які цілі її застосування. Для цього створимо короткий сценарій, поверхневий сценарій та повноцінний UseCase-сценарій.

Короткий usecase сценарій:

Мета сценарію: користувач хоче перевірити, як виглядатиме меблі у кімнаті.

Хід подій: Користувач запускає AR-застосунок. Камера смартфона зчитує простір кімнати. Користувач обирає об’єкт із бібліотеки меблів (наприклад, стіл). Система розміщує віртуальний об’єкт у реальному просторі.

Результат: користувач бачить стіл у своїй кімнаті в масштабі 1:1.

Поверхневий usecase сценарій:

Мета сценарію: користувач планує розташування меблів у кімнаті та зміну зовнішнього виду.

Хід подій:

1) Користувач відкриває AR-застосунок та наводить камеру на кімнату.
2) Система виконує аналіз площини підлоги та стін.
3) Користувач додає диван і стіл із бібліотеки об’єктів.
4) Користувач масштабує диван і переміщує його ближче до стіни.
5) Користувач змінює колір оббивки дивана для перевірки стилістики.
6) Користувач зберігає сцену.

Результат: користувач отримує попереднє планування вітальні у вигляді AR-сцени та збереженого проєкту.

Таблиця 3.1 – Повноцінний Usecase сценарій

Primary Actor	Користувач
Scope	AR-застосунок для планування інтер’єру
Level	Мета користувача
Preconditions	Користувач має смартфон із підтримкою ARCore/ARKit, камера активована
Stakeholders and interests	Користувач: отримати реалістичне уявлення про інтер’єр та перевірка кілька варіантів розташування меблів
Success guarantee	Користувач створив AR-сцену, зберіг проєкт і задоволений результатом візуалізації.
Main Success Scenario	1) Користувач запускає застосунок. 2) Система ініціалізує AR-сесію та виконує відстеження площин. 3) Користувач обирає меблі з бібліотеки (наприклад, шафу). 4) Система відображає шафу у кімнаті в масштабі 1:1. 5) Користувач переміщує шафу ближче до стіни, масштабує її. 6) Користувач додає другий об’єкт (стіл) та перевіряє, чи є достатньо простору. 7) Користувач змінює колір стільниці для перевірки стилю. Користувач зберігає сцену у проєкті.
Extensions	1) Якщо камера не може розпізнати площину – система просить користувача повільно перемістити телефон для повторного сканування.

Кінець таблиці 3.1

	2) Якщо об’єкт занадто великий – система пропонує масштабувати його.
	3) Якщо користувач випадково видалив об’єкт – передбачено функцію «Скасувати».
Special Requirements	1) Потрібне достатнє освітлення для якісного AR-трекінгу. 2) Стабільна робота на середніх за потужністю смартфонах.
Technology and Data Variations List	1) Використання AR Foundation (Unity) з підтримкою ARKit/ARCore. 2) Збереження проєктів у локальній БД (JSON). 3) Можливість експорту результатів у формат PNG.
Frequency of Occurrence	Користувач може створювати нові проєкти кожного разу при перепланування інтер’єру.

Сценарій вище демонструє роботу AR-застосунку, коли користувач взаємодіє з віртуальними об’єктами у реальному просторі. У прикладі користувач розміщує меблі у кімнаті, змінює їх розмір та колір, перевіряє зручність розташування і візуальну гармонію. Система, завдяки технологіям доповненої реальності, відслідковує площини та масштаби приміщення, забезпечуючи коректне відображення моделей у просторі.

Метою використання такого застосунку є підвищення ефективності процесу планування інтер’єру: користувач отримує змогу оцінити майбутній вигляд кімнати до здійснення реальних змін, уникнути помилок при виборі меблів чи декору та заощадити ресурси. Таким чином AR-застосунок виступає інструментом прийняття рішень, роблячи процес дизайну більш наочним, гнучким і доступним навіть для людей без спеціальної підготовки.

3.2 Діаграми станів системи

Діаграма станів допомагає показати, як працює AR-застосунок під час взаємодії користувача з інтерфейсом і 3D-об'єктами. Вона відображає, які стани може мати система, які події викликають переходи між цими станами, та як виглядає повний життєвий цикл використання застосунку [17].

Метою побудови діаграми є наочна демонстрація логіки роботи системи: від запуску AR-сесії до розміщення й редагування об'єктів та завершення роботи користувача.

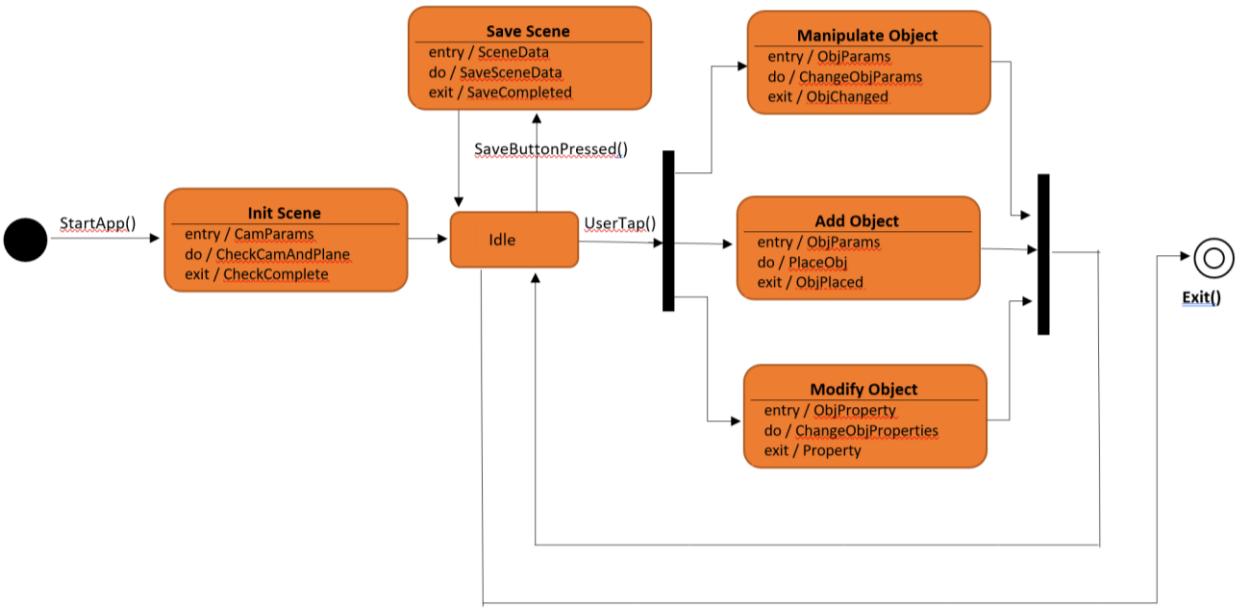

Рисунок 3.1 – Діаграма станів AR застосунку

На діаграмі станів зображено ключові стани AR-застосунку:

- Ініціалізація сцени (Init Scene) – запуск програми, перевірка камери, визначення площин;
- Очікування (Idle) – система готова до взаємодії, очікує дій користувача;
- Додавання об'єкта (Add Object) – користувач обирає меблі чи інший елемент із бібліотеки;

- Маніпуляція об’єктом (Manipulate Object) – користувач переміщує чи обертає об’єкт;
- Зміна параметрів (Modify Object) – зміна кольору, матеріалу або інших властивостей об’єкта;
- Збереження (Save Scene) – користувач зберігає проєкт або робить скріншот;
- Завершення сесії (Exit) – користувач завершує роботу або виходить із застосунку.

Діаграма станів AR-застосунку демонструє основний життєвий цикл роботи системи. Початковим станом є ініціалізація сцени (Init), де виконується запуск програми, перевірка камери та визначення площин. Після цього система переходить у стан очікування (Idle), де вона готова до дій користувача.

Далі можливі переходи до таких станів:

- Додавання об’єкта (Add Object) – вибір меблів або елементів із бібліотеки;
- Маніпуляція об’єктом (Manipulate Object) – переміщення, обертання;
- Зміна параметрів (Modify Object) – вибір кольору або матеріалу.

Після внесення змін користувач може або повернутися до очікування для нової взаємодії, або перейти до стану збереження (Save), де результати фіксуються у вигляді проєкту чи скріншоту. Завершальним станом є кінець роботи (End), коли користувач виходить із застосунку.

Таким чином, діаграма відображає повторюваний цикл «Idle → Add Object/Manipulate Object/Modify Object→ Save/Exit», що забезпечує гнучку та зручну роботу із застосунком у процесі планування інтер’єру.

3.3 Діаграма прецедентів

Діаграма прецедентів використовується для моделювання функціональної поведінки системи з точки зору користувача. Вона показує не те, як система працює всередині, а те, що саме вона повинна робити для акторів (користувачів або інших систем) [18]. Тому розглянемо діаграму прецедентів для AR застосунку планування інтер’єру (рис. 3.2).

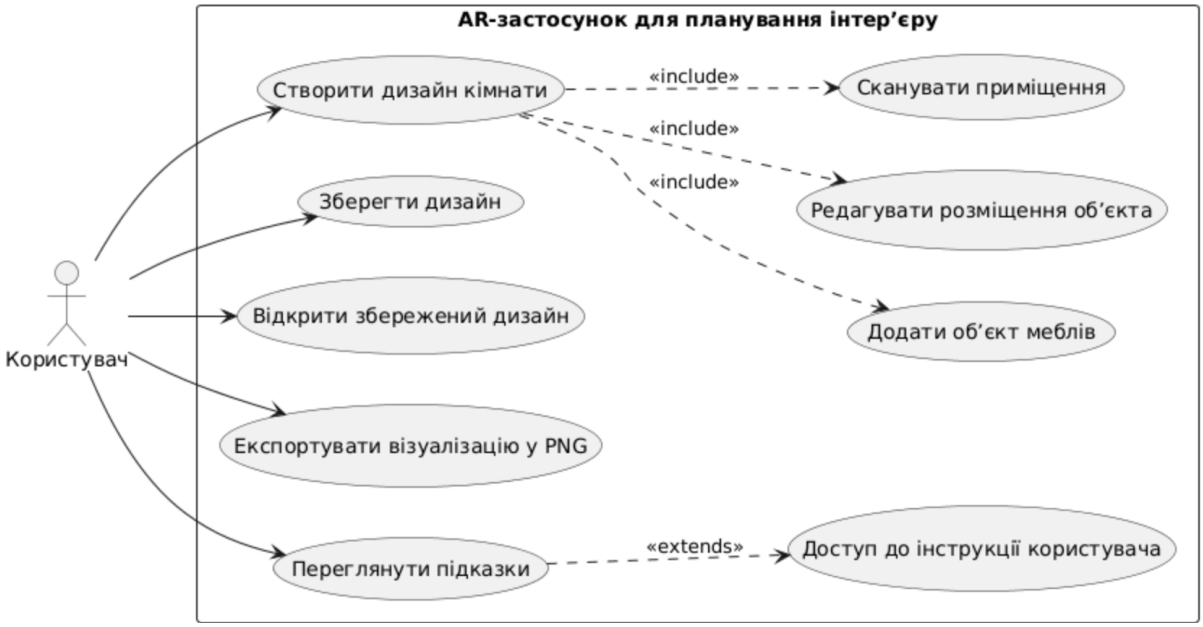

Рисунок 3.2 – Діаграма прецедентів

На цій діаграмі можна побачити основні сценарії взаємодії користувача із застосунком доповненої реальності для планування інтер’єру. Користувач виступає єдиним актором системи та має доступ до всіх ключових функцій застосунку, серед яких створення нового дизайну кімнати, робота з існуючими проєктами та отримання довідкової інформації.

Центральним прецедентом є «Створити дизайн кімнати», який визначає основний робочий процес користувача. Цей прецедент включає в себе такі підзадачі, як «Сканувати приміщення», «Додати об’єкт меблів» та «Редагувати розміщення об’єкта». Включення цих прецедентів («include») відображає те, що вони є невід’ємними етапами створення інтер’єру та виконуються в межах головного сценарію автоматично або за необхідності.

Окрім основних сценаріїв створення інтер’єру та роботи з AR-сценою, діаграма прецедентів включає три важливі функціональні можливості, що забезпечують повноцінне використання застосунку в реальних умовах.

«Зберегти дизайн» – це прецедент, який дозволяє користувачеві зафіксувати поточний стан створеної AR-сцени у вигляді проєкту. Збереження включає інформацію про розташування об’єктів меблів, їх параметри, вибрані моделі та структуру кімнати. Такий механізм забезпечує можливість повернення до роботи над дизайном у будь-який момент, що є важливою складовою зручності застосунку.

«Відкрити збережений дизайн» відповідає за завантаження раніше створених проєктів. Цей сценарій дає змогу користувачу продовжити роботу над дизайном, переглянути збережені варіанти або виконати подальше редагування. Прецедент є логічним продовженням функції збереження та забезпечує безперервність робочого процесу.

«Експортувати візуалізацію» дозволяє користувачеві створити кінцеве представлення проєкту у вигляді зображення або скриншота AR-сцени. Експортований результат може бути використаний для презентації, консультацій із фахівцями, обговорення з родиною чи друзями або для подальшого планування покупки меблів. Цей прецедент завершує цикл роботи над дизайном, даючи можливість вивести проєкт за межі застосунку.

Окремо передбачено прецедент «Переглянути підказки», який допомагає швидко отримати інструкції щодо роботи з інтерфейсом. Цей прецедент розширює функціональність «Доступ до інструкції користувача» («extends»), показуючи, що короткі підказки можуть бути частиною більш розгорнутого користувацького посібника.

Таким чином, діаграма прецедентів демонструє логічну структуру застосунку, відображає залежності між його функціональними можливостями та показує, яким чином користувач взаємодіє з ключовими елементами системи при створенні та редагуванні інтер’єрних проєктів.

3.4 Діаграма послідовності

Діаграма послідовності використовується для моделювання динамічної поведінки системи, тобто того, як різні компоненти взаємодіють між собою протягом виконання певного сценарію. Вона показує, у якій послідовності передаються повідомлення між об’єктами, які операції викликаються та в якому порядку відбуваються події [19].

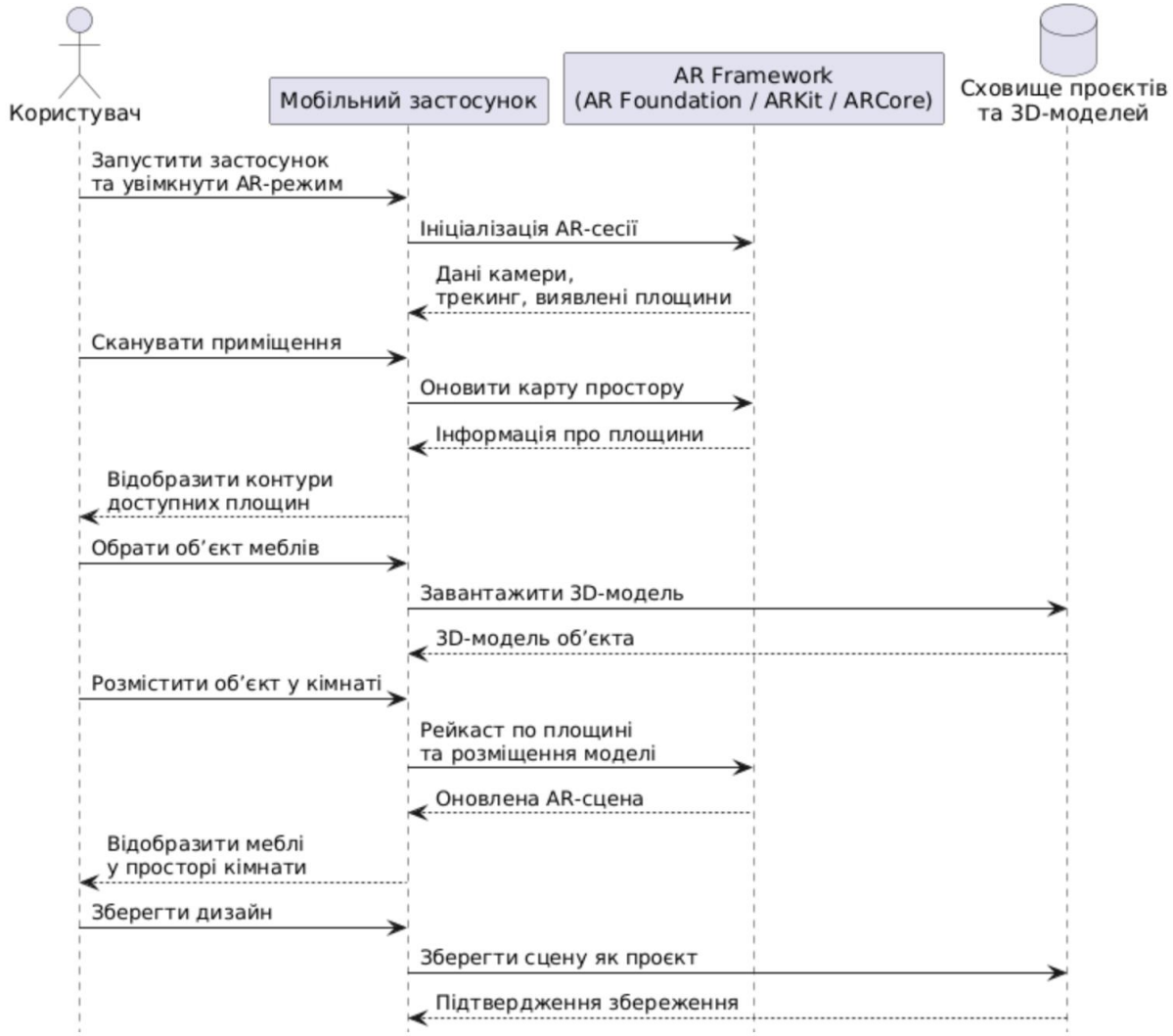

Рисунок 3.3 – Діаграма послідовності

На створеній діаграмі послідовності відображено порядок взаємодії між користувачем, мобільним застосунком, AR-фреймворком та сховищем даних під час створення дизайну кімнати в режимі доповненої реальності. Діаграма демонструє динаміку роботи системи та послідовність обміну повідомленнями

у межах основного сценарію – від запуску AR-режиму до збереження готового проєкту.

Процес починається з того, що користувач запускає застосунок і активує AR-режим. У відповідь мобільний застосунок ініціалізує AR-сесію, звертаючись до AR Foundation або нативних платформ ARKit/ARCore. Фреймворк повертає дані камери, координати пристрою та інформацію про виявлені площини, що дає змогу застосунку відобразити користувачеві структуру навколишнього простору.

Далі користувач виконує сканування приміщення. Застосунок передає AR-фреймворку запит на оновлення карти простору, після чого отримує уточнені дані про площини та відображає їх контури на екрані. Це дозволяє визначити місця, придатні для розміщення віртуальних об’єктів меблів.

Після цього користувач обирає потрібний предмет меблів. Застосунок звертається до внутрішнього сховища 3D-моделей або до локальної бази збережених активів, отримує модель обраного об’єкта та готує її до рендерингу. На наступному етапі користувач розміщує модель у кімнаті. Для цього застосунок виконує AR-рейкастинг – аналізує торкання екрана та визначає точку перетину променя з реальною площиною. Фреймворк повертає координати цієї точки, що дозволяє розмістити обраний об’єкт у відповідному місці та відобразити його у контексті реального середовища.

Після завершення роботи користувач може зберегти створений дизайн. Мобільний застосунок формує структуру проєкту (розташування меблів, їх параметри, стан сцени) і передає її до сховища. Після успішного збереження система повертає підтвердження, яке відображається користувачеві у вигляді повідомлення.

Таким чином, діаграма послідовності демонструє покрокову взаємодію всіх компонентів AR-застосунку, підкреслюючи логіку виконання ключових операцій: ініціалізацію AR-середовища, сканування приміщення, завантаження моделей, розміщення меблів у просторі та збереження

результатів роботи. Вона наочно показує, як система реагує на дії користувача та як відбувається обмін даними між різними частинами програмного забезпечення.

3.5 Діаграма класів

Розглянемо діаграму класів, яка демонструє структуру AR-застосунку для планування інтер’єру. Діаграма відображає основні сутності, їхні атрибути та взаємозв’язки. У ній показано, які дані обробляються системою: кімнати, меблі, параметри об’єктів, каталог моделей та механізми збереження.

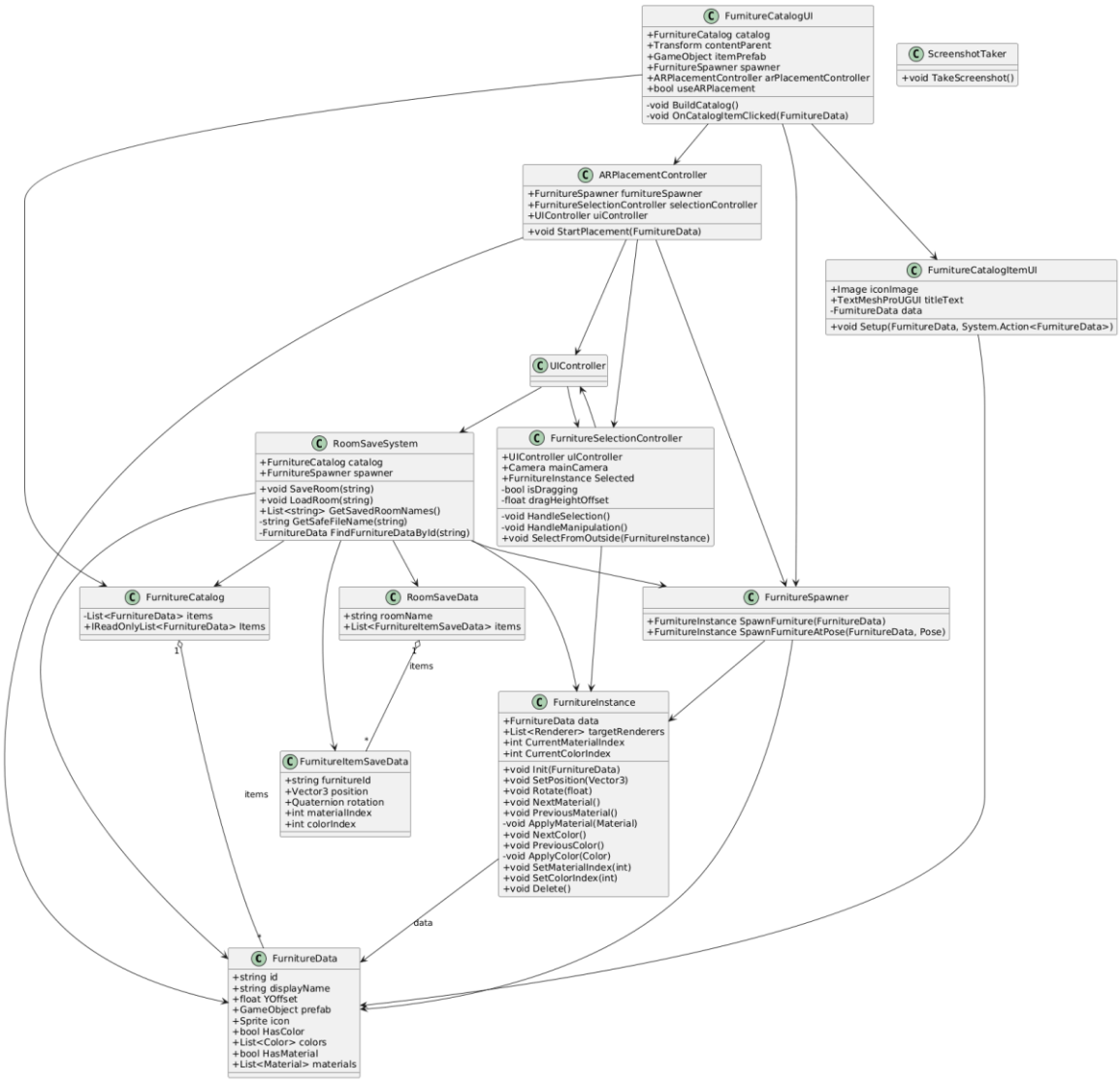

Рисунок 3.4 – Діаграма класів застосунку

Діаграма класів має декілька рівнів:

– рівень даних. FurnitureData описує модель меблів (id, назва, префаб, іконка, доступні матеріали й кольори). Колекція таких моделей зберігається в FurnitureCatalog, який виступає єдиним каталогом доступних об’єктів;

– рівень сцени. FurnitureInstance представляє окремий екземпляр меблів у кімнаті: містить посилання на FurnitureData, поточний матеріал і колір, а також методи для переміщення, обертання, зміни зовнішнього вигляду і видалення. Створення екземплярів винесено в FurnitureSpawner, який уміє спавнити меблі в стандартну точку сцени або в заданий AR-Pose;

– збереження кімнат. Структури FurnitureItemSaveData і RoomSaveData описують один об’єкт і всю кімнату (позиції, повороти, індекси матеріалів/кольорів). Клас RoomSaveSystem відповідає за запис і читання цих даних у JSON-файли, а також за відновлення сцени через FurnitureSpawner і FurnitureCatalog;

– ui та керування. UIController керує панелями, слайдером висоти, полем введення назви кімнати й списком збережених кімнат; через нього викликаються методи RoomSaveSystem і налаштовується інтерфейс під вибраний об’єкт. FurnitureSelectionController обробляє вибір меблів їх переміщення по поверхні, поворот, зміну матеріалів/кольорів і видалення;

– каталог в інтерфейсі. FurnitureCatalogUI будує список елементів каталогу на основі FurnitureCatalog і при натисканні на елемент викликає спавн або AR-розміщення. Окремий клас FurnitureCatalogItemUI описує один елемент списку з картинкою й назвою;

– AR та допоміжні можливості. ARPlacementController реалізує розміщення меблів у просторі доповненої реальності (використовуючи дані AR-рейкастів і FurnitureSpawner) та одразу передає створений об’єкт у FurnitureSelectionController для подальшого редагування. ScreenshotTaker дає змогу робити скріншоти сцени у вигляді зображень.

3.6 Діаграма розгортання

Діаграма розгортання - це тип діаграми, який визначає фізичне обладнання, на якому працюватиме програмна система. Він також визначає, як програмне забезпечення розгортається на базовому обладнанні. Він прив’язує частини програмного забезпечення системи до пристрою, який збирається її виконувати [20].

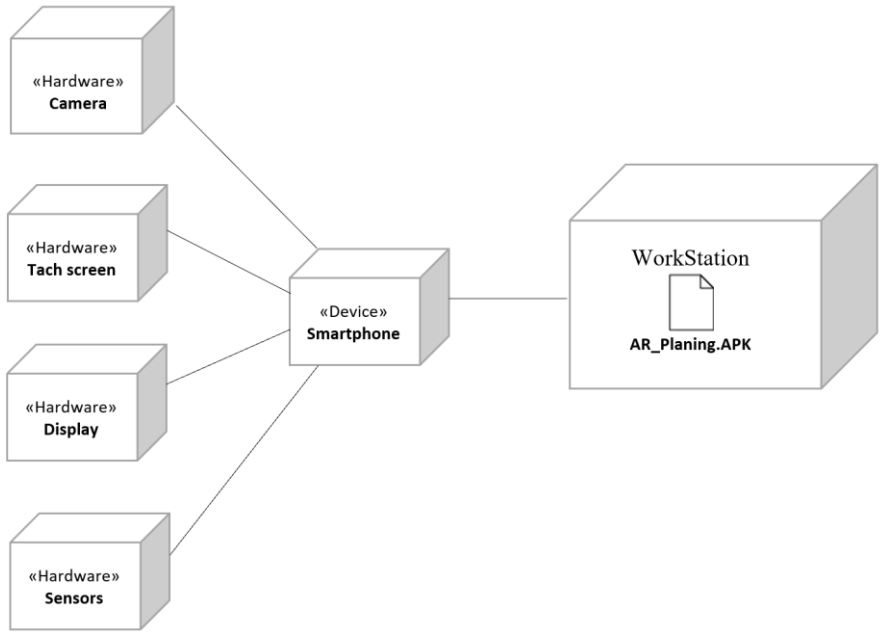

Рисунок 3.5 – Діаграма розгортання застосунку

На діаграмі зображено, що AR-застосунок для планування інтер’єру працює на мобільному пристрої (смартфоні), який включає в себе такі апаратні компоненти:

- Камера мобільного пристрою – використовується для зчитування простору та визначення площин;
- Дисплей – для відображення AR-сцени та візуалізації меблів у реальному масштабі;
- Сенсори руху (акселерометр, гіроскоп, LiDAR) – забезпечують точність позиціонування й визначення глибини;

- Тачскрін – для реалізації інтерактивної взаємодії з об’єктами (масштабування, переміщення, обертання за допомогою жестів).

Таким чином, застосунок працює виключно у межах локального середовища мобільного пристрою, не потребуючи підключення до зовнішніх серверів для основного функціоналу. Це підвищує стабільність та забезпечує роботу навіть без доступу до мережі інтернет. Водночас для завантаження або оновлення бібліотеки моделей меблів може використовуватися мережева синхронізація, однак основні функції AR-візуалізації залишаються незалежними від онлайн-з’єднання.

Висновки до розділу 3

У цьому розділі виконано логічне та структуроване проєктування AR-застосунку для планування інтер’єру на основі попередньо сформованих вимог. На основі сценаріїв використання системи були описані короткий, поверхневий та повноцінний use case-сценарії, що дозволило чітко визначити основну мету роботи застосунку: надання користувачу засобів для візуального планування інтер’єру, експериментів із розташуванням меблів та перевірки різних варіантів оформлення приміщення до здійснення реальних змін. Це заклало основу для подальшої побудови UML-діаграм.

За допомогою діаграми станів змодельовано життєвий цикл роботи системи: від ініціалізації AR-сцени та очікування дій користувача до додавання, маніпуляції та модифікації об’єктів, збереження результатів і завершення сесії. Така модель дозволяє зрозуміти, як змінюються стани застосунку під час типового використання та які події призводять до переходів між ними.

Діаграма класів описала статичну структуру програмної системи: виділено основні сутності (кімната, сцена, об’єкти меблів і декору, бібліотека моделей, менеджер збереження), їх атрибути та зв’язки. Це дало змогу формалізувати дані, з якими працює застосунок, та визначити розподіл

відповідальностей між класами, що є важливим кроком перед безпосередньою реалізацією.

Діаграма розгортання показала, що AR-застосунок функціонує на мобільному пристрої та використовує його апаратні ресурси – камеру, дисплей, сенсори руху та сенсорний екран. Така модель підкреслює автономність роботи застосунку та незалежність основного функціоналу від мережевого підключення, що підвищує стійкість і зручність використання.

Діаграма прецедентів і діаграма послідовності разом деталізували функціональну поведінку системи: перша – з точки зору користувача, друга – з погляду послідовності викликів між компонентами. На їх основі продемонстровано, як користувач створює та редагує дизайн кімнати, а система поетапно виконує ініціалізацію AR-сесії, сканування простору, завантаження моделей, розміщення об’єктів та збереження проєкту

У сукупності побудовані UML-діаграми сформували цілісну модель системи, що описує її функціональність, стани, структуру, інфраструктуру розгортання та динаміку роботи. Результати третього розділу створюють необхідне архітектурне та логічне підґрунтя для переходу до наступного етапу – безпосередньої програмної реалізації AR-застосунку, описаної у четвертому розділі.

4 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ІЗ ПЛАНУВАННЯ ІНТЕР'ЄРУ В РЕЖИМІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

4.1 Ознайомлення із базовим функціоналом Unity 3D

Графічний інтерфейс Unity в міру простий. Для початку роботи із користувацьким інтерфейсом треба знати наступні його частини та кнопки, це дозволить без сторонньої допомоги користуватись Unity 3d. Для прикладу беремо проєкт, що розробляється, та завдяки ньому опишемо основні компоненти користувацького інтерфейсу (рис. 4.1).

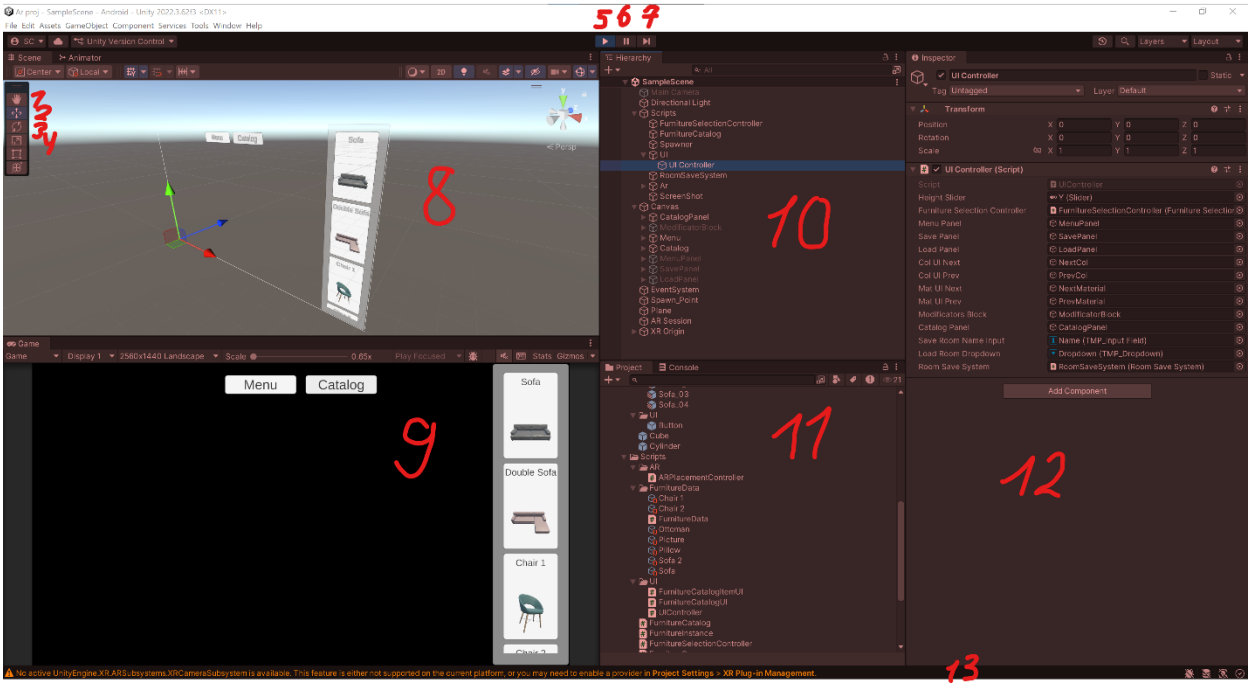

Рисунок 4.1 – Користувацький інтерфейс Unity 3D

Опишемо основні елементи вікна застосунку [21]:

1) Кнопка, що дозволяє обертатись у тривимірному простору на ігровій сцені;

2) Кнопка, що дозволяє рухатись у тривимірному простору на ігровій сцені;

3) Кнопка, що дозволяє редагувати кути об'єктів у тривимірному просторі;

4) Кнопка, що дозволяє редагувати розміри об’єктів у тривимірному просторі;

5) Кнопка «Play» – дозволяє скомпілювати та запустити гру;

6) Кнопка «Stop» – дозволяє зупинити гру;

7) Кнопка «Next frame» – промотувати гру по кадрово;

8) Scene – Ігрова сцена для розробника. Сюди переносяться всі об'єкти, моделі, систему часток, та інше;

9) Вікно «Game» – показує нам те, як саме буде бачити гравець саму гру;

10) Hierarchy – ієрархія об'єктів у грі. Тут відображаються всі об'єкти, що є на ігровій сцені;

11) Project – ієрархія проєкту. У ньому демонструється все, що містить у собі проєкт (звуки, текстури, матеріали, ассети, скрипти та інше);

12) Inspector – сюди переносяться компоненти для об’єктів на сцені;

13) Console – відображає результати логів.

Тепер, коли є представлення про користувацький інтерфейс Unity 3d, можна перейти до основних компонентів Unity 3d, завдяки яким можна створювати проєкти:

1) Scene – це місце, де ви працюєте з контентом в Unity. Це ресурси, що містять всю гру чи програму або їх частину. Наприклад, ви можете створити просту гру з однієї сцени, тоді як для складнішої гри ви можете використовувати одну сцену на рівень, кожна з власним середовищем, персонажами, перешкодами, декораціями та інтерфейсом користувача. Ви можете створити будь яку кількість сцен у проєкті. Коли створюється новий проект і відкривають його вперше, Unity відкриває зразок сцени, який містить лише камеру та світло [22];

2) Script – скрипти дозволяють налаштовувати та розширювати можливості проєкту за допомогою коду C#. Скрипти, що походять від вбудованого класу MonoBehaviour Unity, дозволяють створювати власні

компоненти для керування поведінкою GameObjects. За допомогою скриптів, що походять від ScriptableObject, можна ефективно зберігати великі обсяги даних програмі. Крім того, можна почати з порожнього скрипта C# для розробки власних класів, відмінних від Unity [23];

3) Sprite – це тип 2D-ресурсів, які можна використовувати у своєму проєкті Unity;

4) Prefab – система префабів Unity дозволяє створювати, налаштовувати та зберігати GameObject разом з усіма його компонентами, значеннями властивостей та дочірніми GameObjects як ресурс повторного використання. Префаб діє як шаблон, з якого можна створювати нові префаб-екземпляри в сцені [24];

5) Model – це файли, що містять дані про форму та зовнішній вигляд 3D-об'єктів, таких як персонажі, ландшафт або об'єкти оточення. Файли моделей можуть містити різноманітні дані, включаючи сітки, матеріали та текстури. Вони також можуть містити дані анімації для анімованих персонажів [25].

Всі об’єкти мають хоча б один компонент. За замовчення Unity 3D завжди додає об’єкт компонент Transform що містить у собі інформацію про положення, кути та розміри цього об’єкту (рис. 4.2).

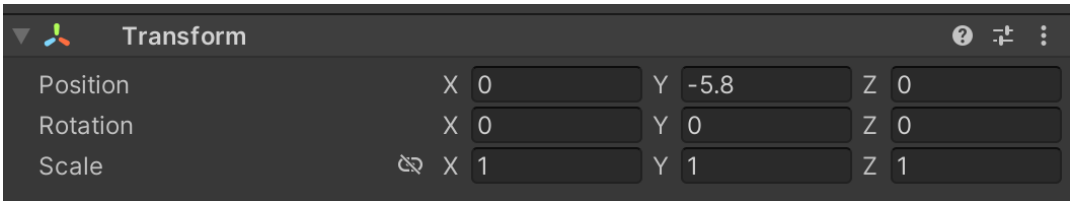

Рисунок 4.2 – Базовий компонент Transform

Для додавання будь якого об’єкту, потрібно його перетягнути на ігрову сцену – це дуже зручно та швидко. Для прототипування можна використати стандартні примітиви Unity 3D (для 3Д це куб, циліндр, сферу та інше, а для 2D – коло, квадрат та інше). Щоб їх використати потрібно у вікні Hierarchy правою кнопкою миші натиснути на пусте місце та обрати у списку потрібний

примітив. Це зручно, коли немає готових моделей, та потрібно випробувати механіки – це пришвидшує та спрощує розробку гри.

До будь якого об’єкта можна додати інші компоненти. Можна це зробити натисканням на потрібний об’єкт, потім натиснути кнопку Add Component у вікні Inspector, та знайти потрібний компонент у представленому списку (рис. 4.3).

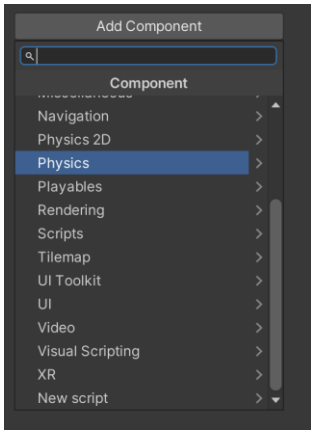

Рисунок 4.3 – Список класифікацій компонентів Unity 3D

Одними із найпоширенішими та часто використовуваними компонентами є:

1) Mesh Filter – форма 3D-об’єкту;
2) Mesh Renderer – відображення 3D-об’єкту;
3) Collider – дозволяє об’єктам фізично взаємодіяти один з одним;
4) Rigidbody – імітація гравітації фізичного об’єкту;
5) Camera –компонент призначений для відображення ігрової сцени, те як її бачить користувач;
6) Light – джерело світла;
7) Audio Source – призначений для відтворення звуків.

4.2 Створення логіки застосунку

Для початку треба створити контейнер де можна зберігати дані про об’єкти які будуть розташовані на сцені. Для цього краще за все підійде клас

який є наслідником від ScriptableObject. Це дозволить зручно оперувати даними. Створимо скрипт FurnitureData та зробимо його наслідником від ScriptableObject. На сріншоті (рис. 4.4) можна побачити поля які має кожен об’єкт та їх опис. Також можна побачити приклад заповненя цих полів (рис. 4.5).

```
[CreateAssetMenu(menuName = "Furniture/Furniture Data")]
Скрипт Unity | Ссылок: 16
public class FurnitureData : ScriptableObject
{
    public string id; // id об'єкту для структурованості
    public string displayName; // назва об'єкту

    public float YOffset; // можливість змінити позицію Y через специфіку 3д об'єктів

    public GameObject prefab; // сам об'єкт
    public Sprite icon; // його іконка

    public bool HasColor; // чи є в об'єкта зміна кольорів
    public List<Color> colors; // список доступних кольорів

    public bool HasMaterial; // чи є в об'єкта зміна матеріалів
    public List<Material> materials; // список доступних матеріалів

}
```

Рисунок 4.4 – Поля контейнеру для зберігання інформації

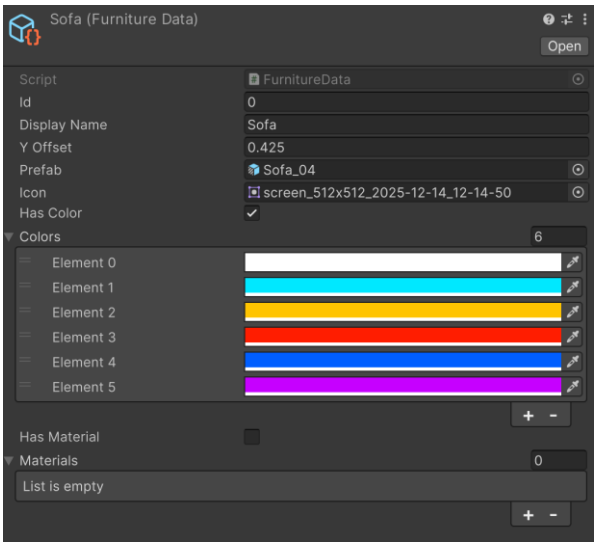

Рисунок 4.5 – Заповнені поля на прикладі об’єкту Sofa

Наступним кроком є створити клас FurnitureCatalog (рис. 4.6) із списком цих контейнерів, щоб за необхідністю кожен клас міг отримати доступ до них.

```
Скрипт Unity (1 ссылка на ресурсы) | Ссылок: 2
public class FurnitureCatalog : MonoBehaviour
{
    [SerializeField]
    private List<FurnitureData> items = new List<FurnitureData>();

    Ссылок: 2
    public IReadOnlyList<FurnitureData> Items => items;
}
```

Рисунок 4.6 – Клас FurnitureCatalog

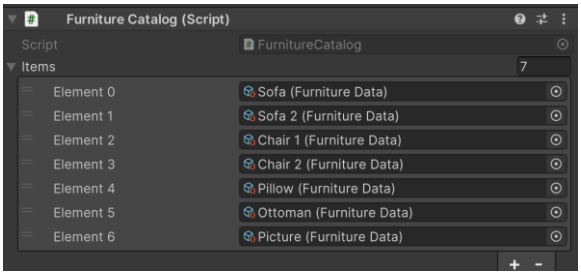

Рисунок 4.7 – Заповнений список класу FurnitureCatalog

Тепер, коли ми маємо об’єкти з якими можна працювати, створимо клас в якому буде ініціалізація майбутніх об’єктів, можливість зміни кольору, матеріалу, поворот, розміщення, видалення. Скорочений код можна подивитися на скріні (рис. 4.8), а з повним кодом можна ознайомитись у додатку А.

```
public class FurnitureInstance : MonoBehaviour
{
    public FurnitureData data;
    public List<Renderer> targetRenderers = new List<Renderer>();
    private int currentMaterialIndex = 0;
    private int currentColorIndex = 0;
    ссылка: 1
    public int CurrentMaterialIndex => currentMaterialIndex;
    ссылка: 1
    public int CurrentColorIndex => currentColorIndex;
    Сообщение Unity | Ссылок: 0
    private void Awake()...
    Ссылок: 2
    public void Init(FurnitureData furnitureData)...
    ссылка: 1
    public void SetPosition(Vector3 position)...
    ссылка: 1
    public void Rotate(float degrees)...
    ссылка: 1
    public void NextMaterial()...
    ссылка: 1
    public void PreviousMaterial()...
    Ссылок: 5
    private void ApplyMaterial(Material mat)...
    ссылка: 1
    public void NextColor()...
    ссылка: 1
    public void PreviousColor()...
    Ссылок: 5
    private void ApplyColor(Color color)...
    ссылка: 1
    public void Delete()...
    ссылка: 1
    public void SetMaterialIndex(int index)...
    ссылка: 1
    public void SetColorIndex(int index)...
}
```

Рисунок 4.8 – Поля та методи для ініціалізації об’єктів

Тепер перейдемо до класу FurnitureSpawner (рис. 4.9) де можна створити об'єкт, додати до нього потрібні компоненти, задати йому позицію.

```
public class FurnitureSpawner : MonoBehaviour
{
    ссылка: 1
    public FurnitureInstance SpawnFurnitureAtPose(FurnitureData data, Pose pose)
    {
        //перевірка наявності компонентів та об'єктів
        if (data == null || data.prefab == null)
            return null;

        //позиція після створення
        Vector3 spawnPos = pose.position + new Vector3(0, data.YOffset, 0);
        //обертання
        Quaternion spawnRot = pose.rotation;

        //створення об'єкту
        GameObject instanceGO = Instantiate(data.prefab, spawnPos, spawnRot);

        //створення компоненту FurnitureInstance
        FurnitureInstance instance = instanceGO.AddComponent<FurnitureInstance>();
        //ініціалізація об'єкту
        instance.Init(data);
        return instance;
    }
}
```

Рисунок 4.9 – Клас FurnitureSpawner

Після того, як створились різні об'єкти, їх якось треба обрати, щоб можна було з ними взаємодіяти. Тому потрібен клас, де буде відслідковуватись натискання на об'єкт (рис. 4.10).

```
public class FurnitureSelectionController : MonoBehaviour
{
    public UIController uIController; //має інформацію про користувацький інтерфейс
    public Camera mainCamera; //головна камера
    public LayerMask furnitureLayer; //ідентифікація об'єктів розміщення
    public LayerMask floorLayer; //ідентифікація полу
    public FurnitureInstance Selected; //обраний об'єкт
    private bool isDragging = false; //чи ще переміщується об'єкт
    private float dragHeightOffset = 0.0f; //ініціалізація зміщення об'єкту
    Сообщение Unity | Ссылок: 0
    private void Update()
    {
        HandleSelection();
        HandleManipulation();
    }
    ссылка: 1
    private void HandleSelection()...
    ссылка: 1
    private void HandleManipulation()
    {
        if (Selected == null) return;

        if (Input.GetMouseButtonDown(0))
        {
            isDragging = true;
            dragHeightOffset = Selected.transform.position.y;
        }
        if (Input.GetMouseButtonUp(0))
            isDragging = false;

        if (isDragging)
        {
            Ray ray = mainCamera.ScreenPointToRay(Input.mousePosition);
            if (Physics.Raycast(ray, out RaycastHit hit, 1000f, floorLayer))
            {
                Vector3 pos = hit.point;
                pos.y = dragHeightOffset;
                Selected.SetPosition(pos);
            }
        }
    }
```

Рисунок 4.10 – Клас FurnitureSelectionController

Справедливо можна задати питання, а що робити, коли розставив меблі, а тепер хочеться зберегти результати своєї роботи та зробити новий набір меблів? Для цього потрібно зробити систему збереження та відтворення кімнати. Створимо клас RoomSaveSystem. В ньому будуть допоміжні класи та основні методи для збереження та відтворення сцени. На скріні (рис. 4.11) можа побачити частковий код, з повним кодом можна ознайомитись у додатку А.

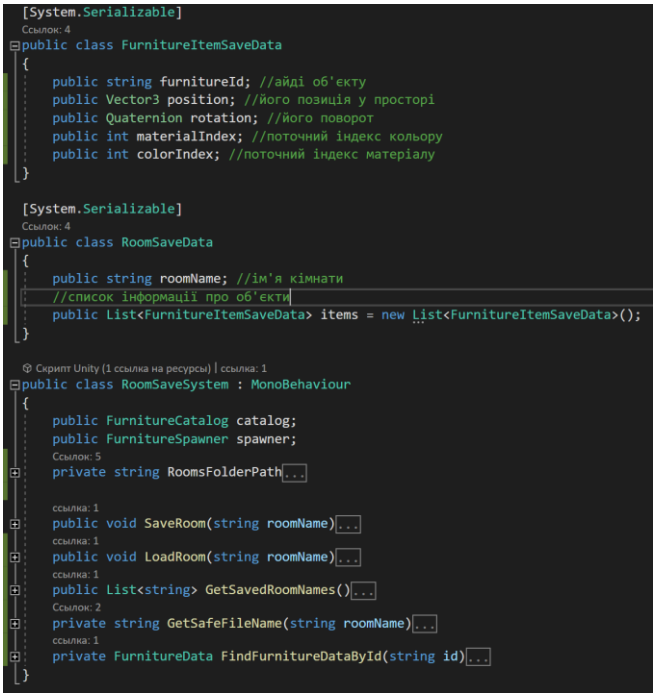

Рисунок 4.11 – Клас RoomSaveSystem

Останньою механікою застосунку є створення скріншоту кімнати яку змоделювали. Для цього створимо клас ScreenshotTaker. На скріні (рис. 4.12) можна побачити що самі він робить.

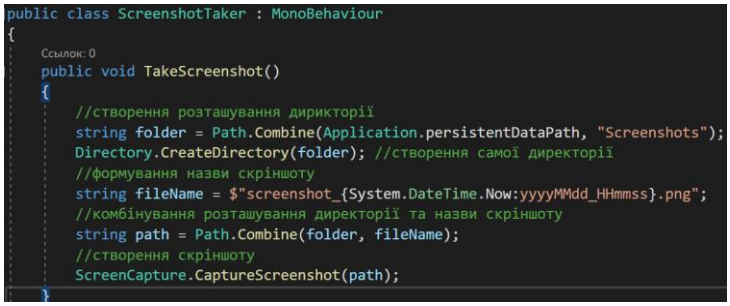

Рисунок 4.12 – Клас ScreenshotTaker

4.3 Користувацький інтерфейс

Тепер, коли в нас є кодова база основних функцій застосунку, треба усі ці методи поєднати із користувацьким інтерфейсом. Для початку, зробимо потрібні панелі та кнопки. Почнемо із головного меню (рис. 4.12).

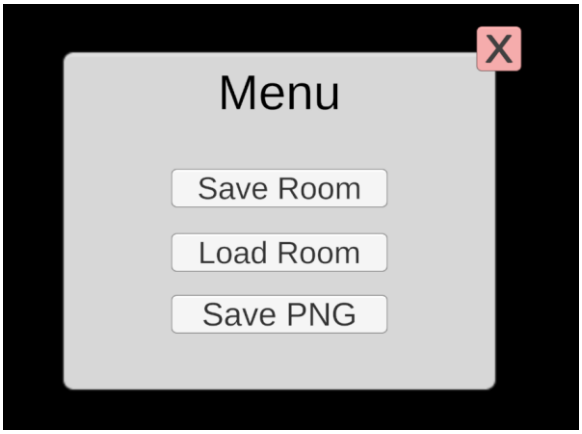

Рисунок 4.12 – Головне меню застосунку

На ньому можна побачити кнопки Save Room, Load Room, Save PNG та кнопку покинути меню. Щоб кнопка працювала, до неї потрібно додати компонент Button, а в ньому у подію OnClick перемістити об’єкт із потрібним скріптом на якому знаходиться потрібний метод. У нашому випадку продемонстровано на прикладі кнопки Save Room із додаванням методу OpenSavePanel() (рис. 4.13).

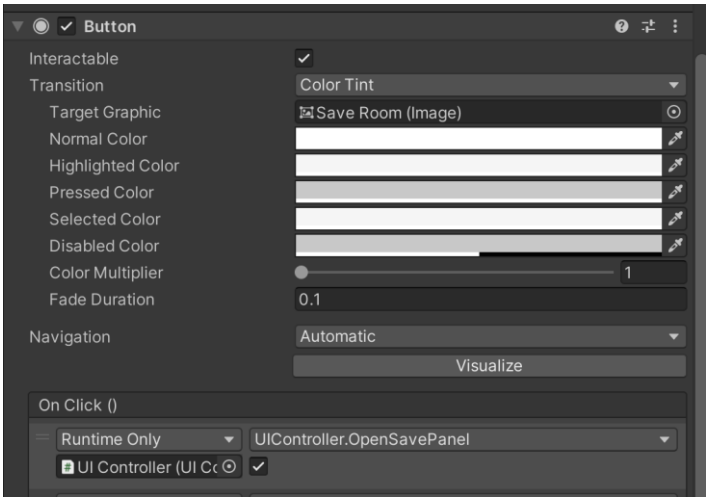

Рисунок 4.13 – Приклад способу відкриття панелі на кнопці

Якщо натиснути на кнопку Save Room, відкриється панель де можна вказати назву кімнати та зберегти кімнату (рис. 4.14). Якщо натиснути Load Room, то відкриється панель де можна завантажити попередньо збережені кімнати (рис. 4.14). І якщо натиснути Save PNG, буде зроблений скріншот.

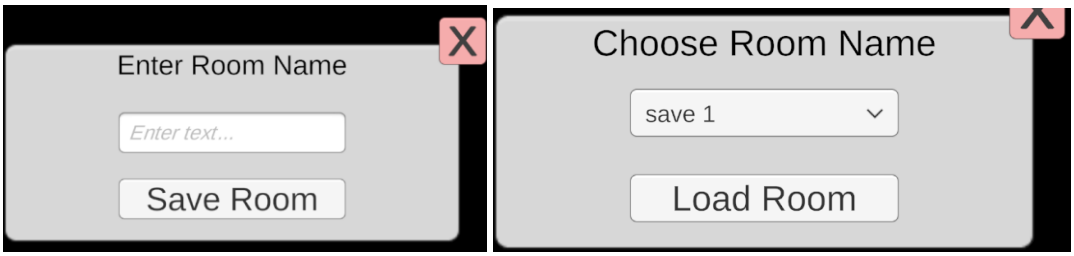

Рисунок 4.14 – Панелі зберігання та завантаження кімнат

Тепер перейдемо до основної панелі застосунку. Там можна побачити дві кнопки Menu та Catalog (рис. 4.15).

Рисунок 4.15 – Кнопки Menu та Catalog

Якщо із Menu все зрозуміло, тоді докладніше розглянемо що робить кнопка Catalog. Вона відкриває список із каталогу предметів, що можна виставити на сцену застосунку у правій частині екрану (рис. 4.16). Цей список можно промотувати, щоб подивитись весь асортимент.

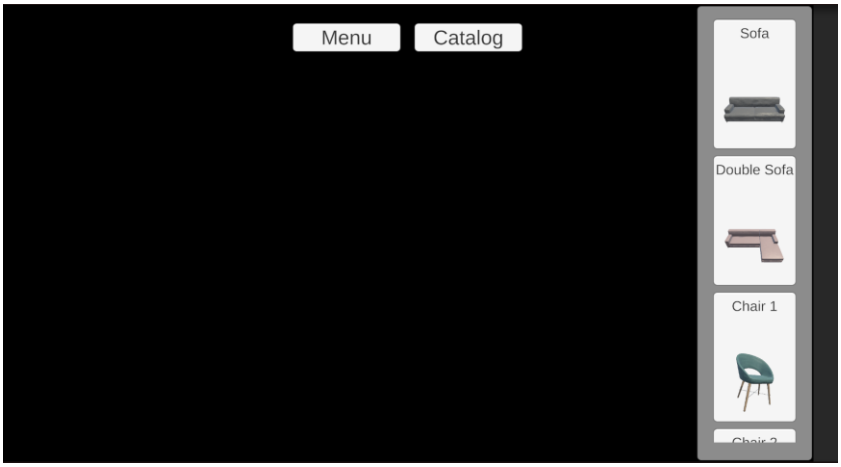

Рисунок 4.16 – Демонстрація каталогу меблів

Такий список своїми руками робити не продуктивно. Тому, використовуючи ScriptableObject контейнер (FurnitureData) який був описаний у попередньому підрозділі «Створення логіки застосунку», створений Prefab (шаблон, який можна використовувати багато разів), де можна побачити місце для іконки об’єкту (який заповнений випадковою картинкою для наглядності) та його назву (заповнено Name, яке потім буде замінено на справжню назву задопомогою скрипта) (рис. 4.17).

Рисунок 4.17 – Prefab для елементу каталогу

Але мало зробити просто префаб. Потрібно через код створити його клони та заповнити його відносно контейнеру FurnitureData. Для цього створимо FurnitureCatalogItemUI (рис. 4.18) де розпишемо поля які потрібні для заповненя префабу та логіку їх заповненя.

```
public class FurnitureCatalogItemUI : MonoBehaviour
{
    public Image iconImage;
    public TextMeshProUGUI titleText;
    public Button button;

    private FurnitureData data;
    private System.Action<FurnitureData> onClick;

    ссылка: 1
    public void Setup(FurnitureData furnitureData, System.Action<FurnitureData> onClickCallback)
    {
        data = furnitureData;
        onClick = onClickCallback;

        if (iconImage != null && data.icon != null)
            iconImage.sprite = data.icon;

        if (titleText != null)
            titleText.text = data.displayName;

        if (button != null)
        {
            button.onClick.RemoveAllListeners();
            button.onClick.AddListener(() =>
            {
                onClick?.Invoke(data);
            });
```

Рисунок 4.18 – Код FurnitureCatalogItemUI

Після цього, навішаємо скрипт на префаб, та заповнимо потрібні поля (рис. 4.19).

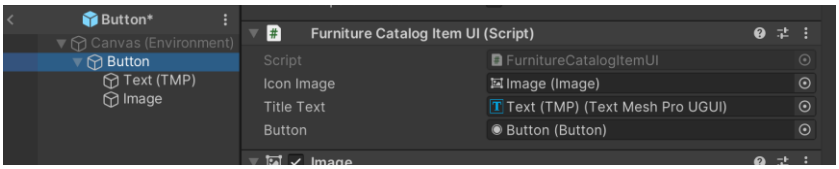

Рисунок 4.19 – Заповнення компонентів префабу

Тепер можна написати скрипт FurnitureCatalogUI (рис. 4.20), який зробить клони цих префабів та заповнить їх значеннями які були взяті у списку FurnitureData. Після цбого ці префаби будуть додані до панелі в якій потім можна обирати фурнітуру.

```
public class FurnitureCatalogUI : MonoBehaviour
{
    public FurnitureCatalog catalog;
    public Transform contentParent;
    public GameObject itemPrefab;
    public FurnitureSpawner spawner;
    public ARPlacementController arPlacementController;
    public bool useARPlacement = true;

    Сообщение Unity | Ссылок: 0
    private void Start()
    {
        BuildCatalog();
    }

    ссылка: 1
    private void BuildCatalog()
    {
        foreach (var item in catalog.Items)
        {
            var go = Instantiate(itemPrefab, contentParent);
            var ui = go.GetComponent<FurnitureCatalogItemUI>();
            if (ui != null)
            {
                ui.Setup(item, OnCatalogItemClicked);
            }
        }
    }

    ссылка: 1
    private void OnCatalogItemClicked(FurnitureData data)
    {
        if (useARPlacement && arPlacementController != null)
        {
            arPlacementController.StartPlacement(data);
        }
        else
        {
            spawner.SpawnFurniture(data);
        }
    }
}
```

Рисунок 4.20 – Код FurnitureCatalogUI

Коли є каталог, можна обирати фурнітуру. Після того, як вона з'явилась на сцені, з'являється панель із кнопками у лівій частині екрану (рис. 4.21). Там представлений слайдер переміщення по осі Y, обертання на 10 градусів в ліву

або в праву сторони, можливість змінити колір та матеріал об'єкту (якщо це застосовно для цього об'єкту), також можна видалити цей об'єкт.

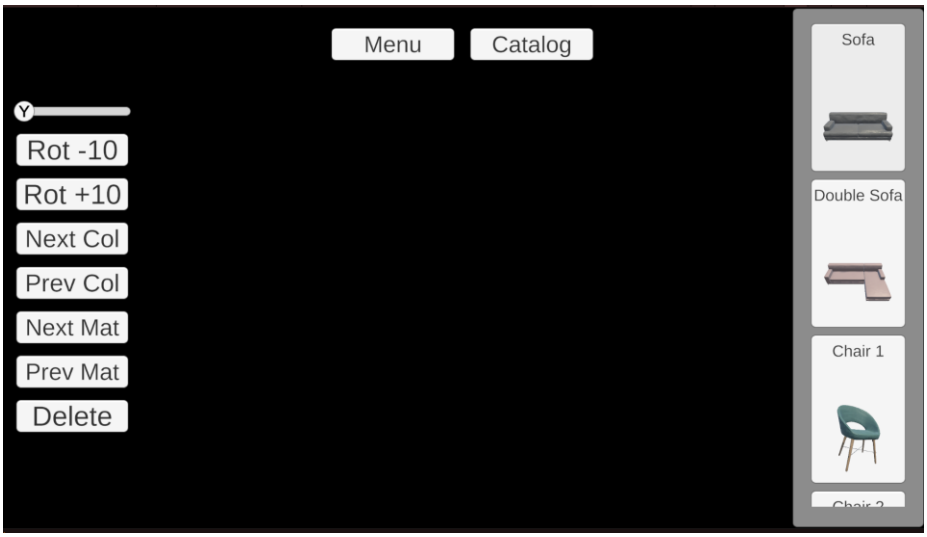

Рисунок 4.21 – Демонстрація панелі модифікації фурнітури

Також у проєкті присутній головний клас UIController (рис. 4.22) який відповідає за увесь користувацький інтерфейс та взаємодію із ним. Він має багато простих методів для взаємодії користувача із користувацьким інтерфейсом, тому немає необхідності на ньому зупинятись докладно.

```
public class UIController : MonoBehaviour
{
    public Slider heightSlider;
    [SerializeField] private FurnitureSelectionController _furnitureSelectionController;
    public GameObject menuPanel;
    public GameObject savePanel;
    public GameObject loadPanel;
    public GameObject colUINext;
    public GameObject colUIPrev;
    public GameObject matUINext;
    public GameObject matUIPrev;
    public GameObject modificatorsBlock;
    public GameObject catalogPanel;
    bool isCatalogOpen = false;
    public TMP_InputField saveRoomNameInput;
    public TMP_Dropdown loadRoomDropdown;
    public RoomSaveSystem roomSaveSystem;
    Сообщение Unity | Ссылок: 0
    private void Start()
    {
        RefreshRoomList();
    }
    Ссылок: 2
    public void ResetUI(bool hasCol, bool hasMat)...
    ссылка: 1
    public void OnHeightSliderChanged(float value)...
    Ссылок: 0
    public void OpenCatalog()...
    Ссылок: 2
    public void OpenModificatorBlock(bool open)...
    Ссылок: 0
    public void OpenMenuPanel(bool active)...
    Ссылок: 0
    public void OpenSavePanel(bool active)...
    Ссылок: 0
    public void OpenLoadPanel(bool active)...
    Ссылок: 0
    public void OnClickSaveRoom()...
    Ссылок: 0
    public void OnClickLoadRoom()...
    Ссылок: 3
    public void RefreshRoomList()...
```

Рисунок 4.22 – Код UIController

4.4 Інтеграція технології доповненої реальності у проєкт

Для початку роботи із доповненою реальністю у Unity 3D, треба завантажити плагін AR Foundation (рис. 4.23).

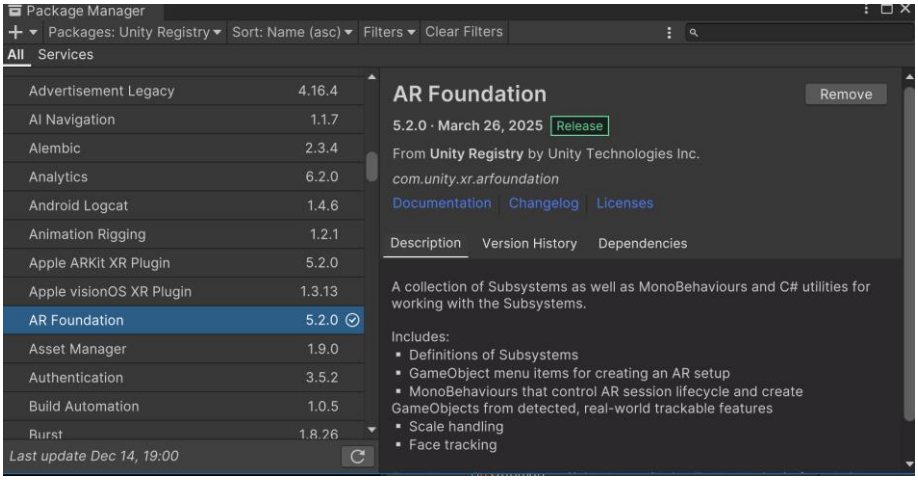

Рисунок 4.23 – Панель завантаження плагіну AR Foundation

Тепер, треба додати необхідні AR компоненти, спеціальну AR камеру щоб все працювало. Першим ділом додаємо AR Session та AR Input Manager (рис. 4.24).

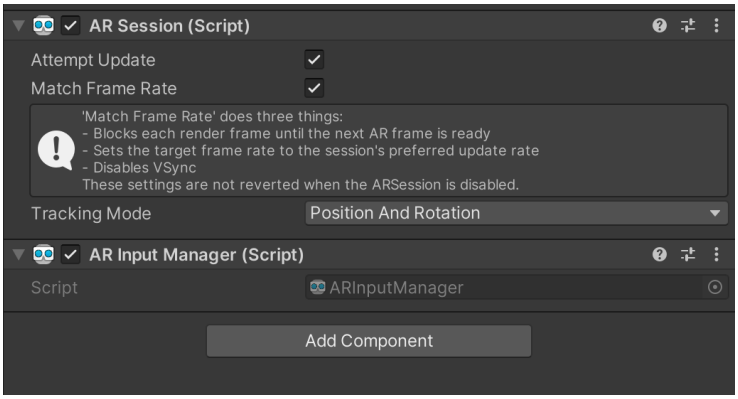

Рисунок 4.24 – Додавання компонентів AR Session та AR Input Manager

AR Session – контролює життєвий цикл доповненої реальності, вмикаючи або вимикаючи доповнену реальність на цільовій платформі. Сеанс посилається на екземпляр AR. Хоча інші функції, такі як виявлення площини, можна незалежно вмикати або вимикати, сеанс контролює життєвий цикл усіх функцій AR [26].

AR Input Manager – дає інформацію про позу пристрою.

Наступними компонентами є XR Origin, Ar Plane Manager, Ar Raycast Manager (рис. 4.25).

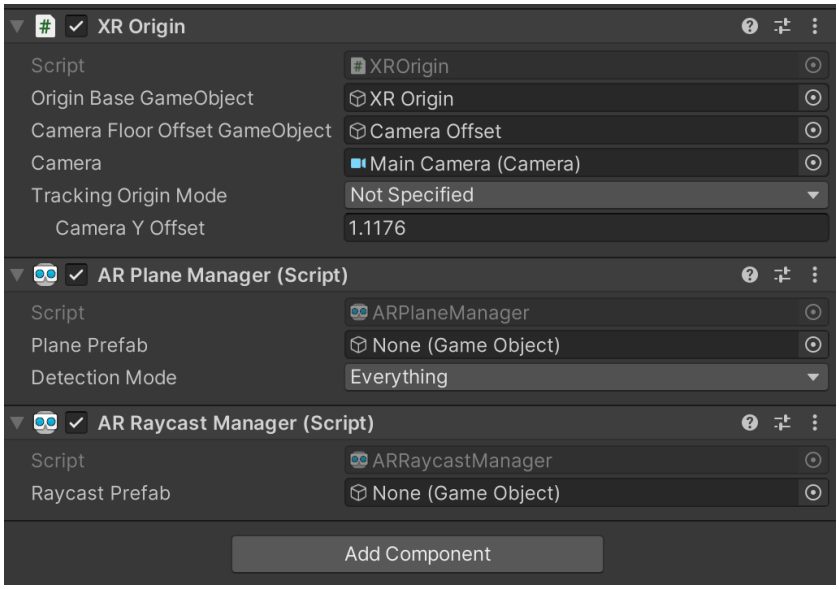

Рисунок 4.25 – Додавання XR Origin, Ar Plane Manager, Ar Raycast Manager

XR Origin – служить центром простору відстеження в XR-сцені. Це набір ігрових об'єктів та компонентів, які працюють разом для перетворення даних відстеження XR у простір світу сцени [27].

Ar Plane Manager – Менеджер площин створює ігрові об'єкти (GameObjects) для кожної виявленої площини в середовищі. Площина — це плоска поверхня, представлена позою, розмірами та граничними точками. Граничні точки є опуклими [28].

Ar Raycast Manager – Також відоме як тестування на потрапляння, кастинг променів дозволяє визначити, де промінь (визначений початком координат та напрямком) перетинається з відстежуваним об'єктом. Поточний інтерфейс кастингу променів тестує лише площини та точки в хмарі точок. Інтерфейс кастингу променів подібний до інтерфейсу в модулі Unity Physics, але оскільки відстежувані об'єкти AR не обов'язково присутні у світі фізики, AR Foundation надає окремий інтерфейс [29].

Останніми важливими копонентами є Ar Camera Manager, Ar Camera Background, Tracked Pose Drive (рис. 4.26).

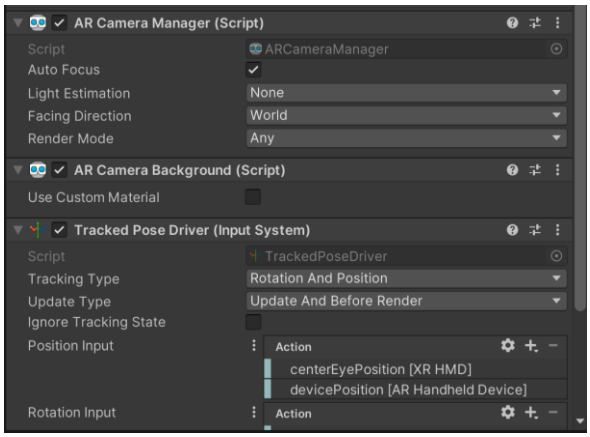

Рисунок 4.26 – Додавання Ar Camera Manager, Ar Camera Background, Tracked Pose Drive

Ar Camera Manager – керує часом життя XRCameraSubsystem. Треба додати один із них до камери у сцені, щоб інформація про текстуру камери та оцінку освітлення була доступною.

Ar Camera Background – додається цей компонент до камери, щоб скопіювати текстуру кольорової камери на фон.

Tracked Pose Drive – компонент TrackedPoseDriver застосовує поточне значення Pose відстежуваного пристрою до перетворення GameObject.

Недостатньо просто додати ці компоненти. Потрібно також написати скрипт, який дозволить користуватись AR функціями. Для цього був створений ARPlacementController (рис. 4.27), щоб можна було виставляти об'єкти на площину та все працювало як потрібно.

```
public class ARPlacementController : MonoBehaviour
{
    public ARRaycastManager arRaycastManager;
    public FurnitureSpawner furnitureSpawner;
    public FurnitureSelectionController selectionController;
    public UIController uiController;
    private static List<ARRaycastHit> hits = new List<ARRaycastHit>();
    private FurnitureData furnitureToPlace;
    private bool isPlacementMode = false;
    ссылка: 1
    public void StartPlacement(FurnitureData data)
    {
        furnitureToPlace = data;
        isPlacementMode = furnitureToPlace != null;
    }

    Сообщение Unity | Ссылок: 0
    private void Update()
    {
        if (!isPlacementMode || furnitureToPlace == null)
            return;
        if (Input.touchCount == 0)
            return;
        Touch touch = Input.GetTouch(0);
        if (touch.phase != TouchPhase.Began)
            return;
        if (arRaycastManager == null)
            return;
        if (arRaycastManager.Raycast(touch.position, hits, TrackableType.PlaneWithinPolygon))
        {
            Pose pose = hits[0].pose;

            FurnitureInstance instance = furnitureSpawner.SpawnFurnitureAtPose(furnitureToPlace, pose);
            isPlacementMode = false;

            if (instance != null && selectionController != null)
            {
                selectionController.SelectFromOutside(instance);
            }
        }
    }
}
```

Рисунок 4.27 – Клас ARPlacementController

4.5 Результат та тестування проєкту

Для початку треба обрати фурнітуру з каталогу. Додаю об’єкт під назвою Sofa. Це сірий диван. Як можна побачити (рис. 4.28) диван створився повернутий.

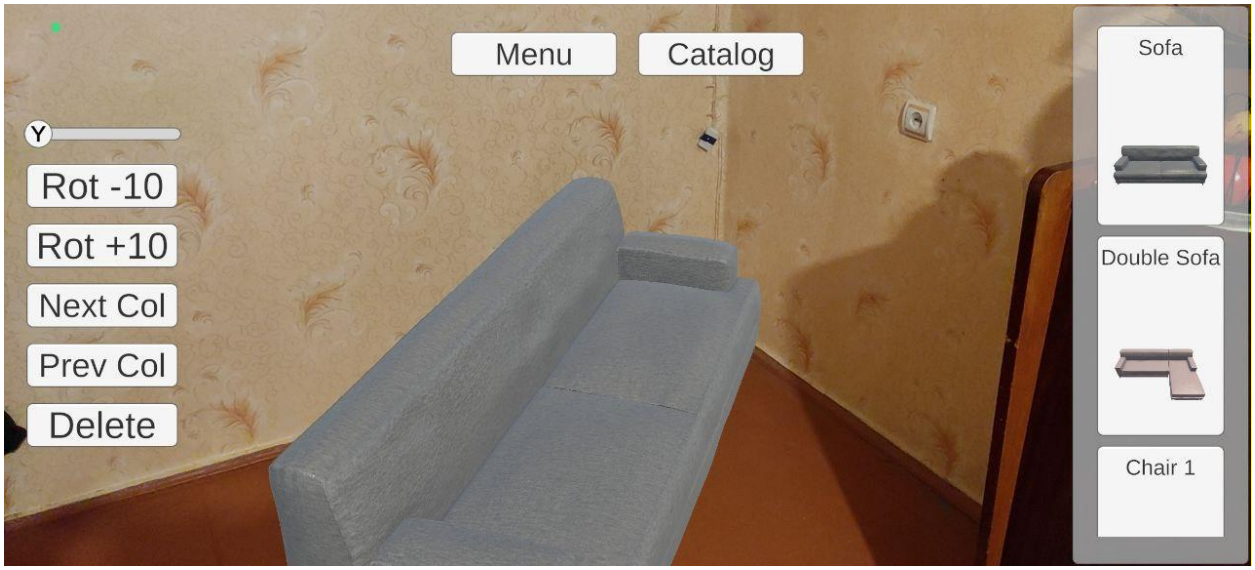

Рисунок 4.28 – Додавання об’єкту Sofa

Повертаємо його декілька разів за допомогою «Rot +10» поки не повернеться потрібною стороною (рис. 4.29).

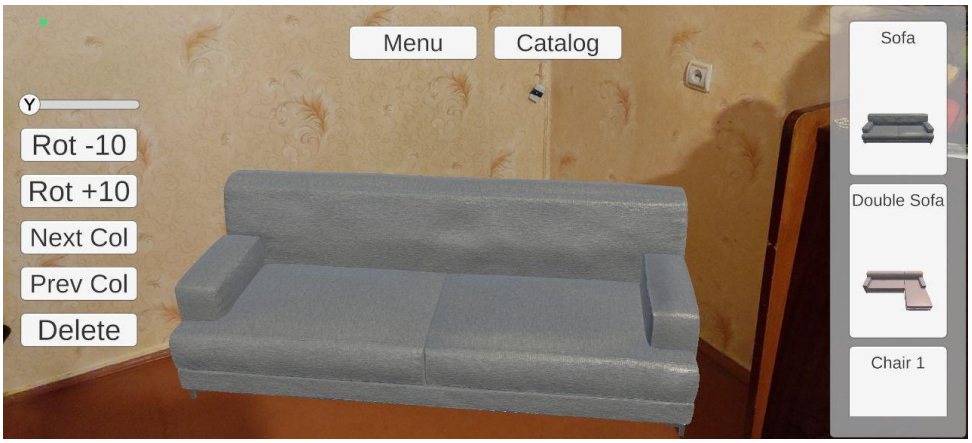

Рисунок 4.29 – Результат обертання дивану

Тепер, коли положення дивану нас влаштовує, знайдемо білу подушку та покладемо її на диван (рис. 4.30).

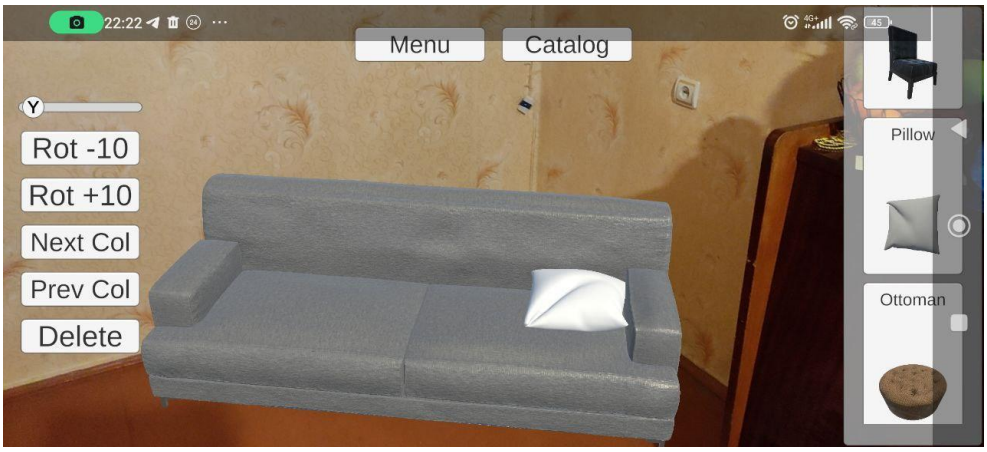

Рисунок 4.30 – Додавання білої подушки

Колір подушки білий – занадто просто. Натискаємо кнопку «Next Col» поки не отримаємо колір, який сподобається (рис. 4.31).

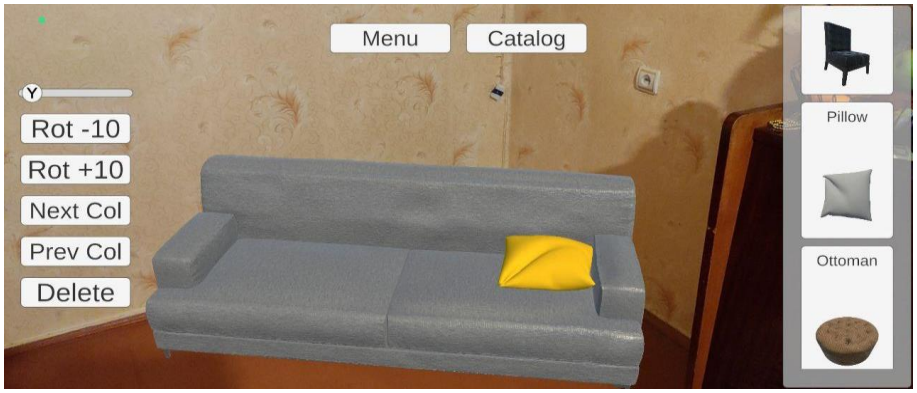

Рисунок 4.31 – Зміна кольору подушки

Після того, як пофарбували подушку у жовтий колір, шукаємо картину, та переносимо її на сцену (рис. 4.32).

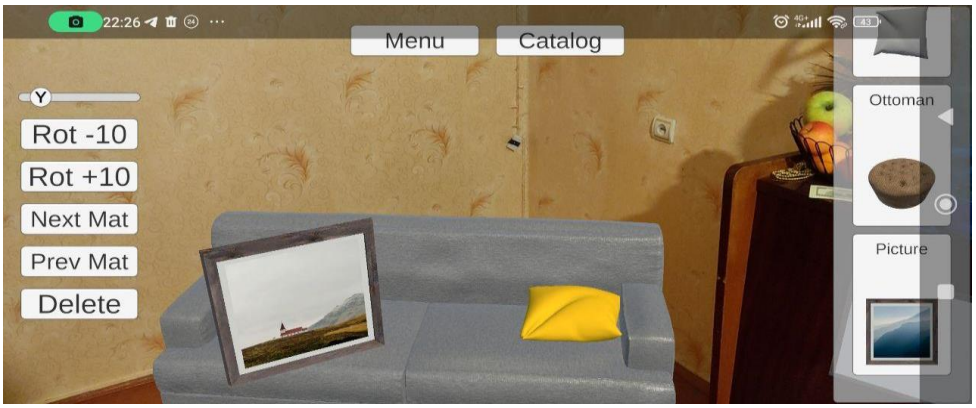

Рисунок 4.32 – Додавання картини

Картина з’явилась занадто низько, можна використати слайдер Y, щоб підняти її на потрібну висоту (рис. 4.33).

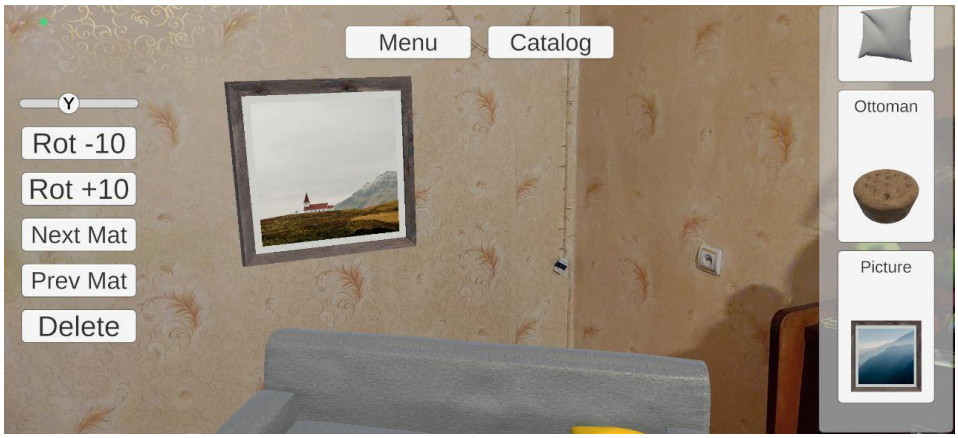

Рисунок 4.33 – Зміна положення картини по Y

Коли картина на потрібній висоті, тепер можна змінити її матеріал, щоб отримати новий вид картин, для цього натискаємо «Next Mat» поки не знайдемо матеріал, який сподобається (рис. 4.34).

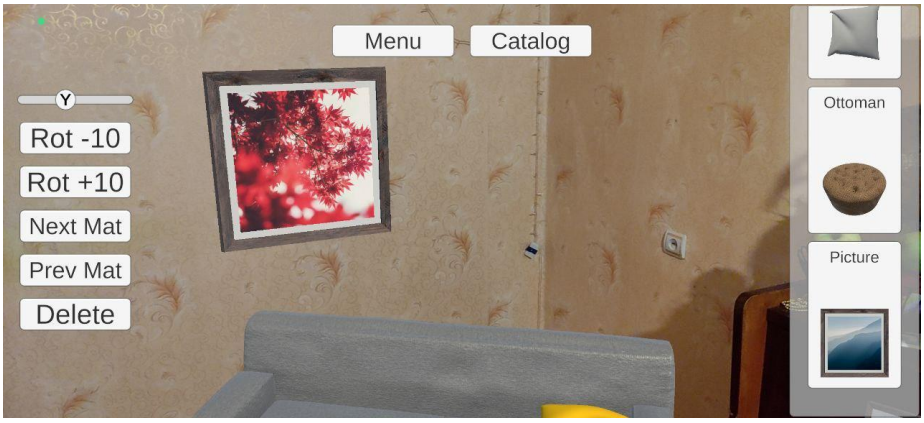

Рисунок 4.34 – Зміна матеріалу картини

Коли сцена нам до вподоби, її можна зберегти. Для цього перейдемо у «Menu» та натиснемо на кнопку «Save Room», у вікні у назві вказуємо «КРМ 1» (рис. 4.35), та натискаємо кнопку «Save Room».

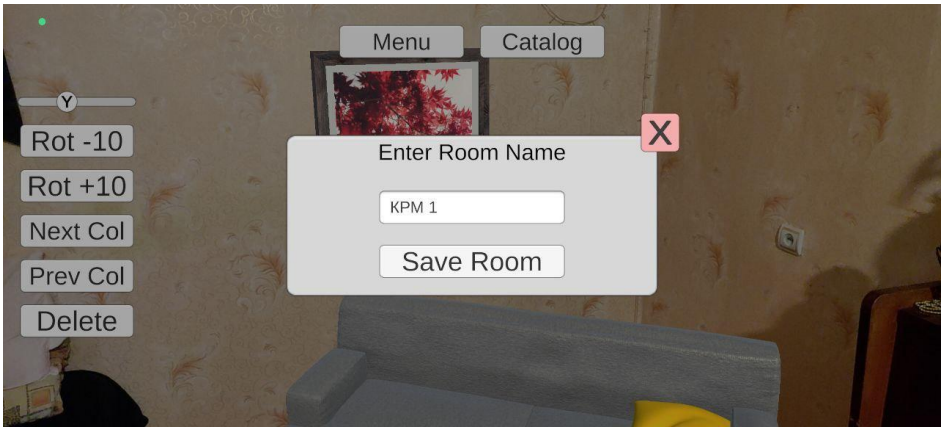

Рисунок 4.35 – Процес збереження кімнати

Тепер можна перезайти у застосунок, або видалити усі об’єкти на сцені чи додати нові об’єкти і потім спробувати завантажити збережену сцену. Можна побачити, що у панелі «Load Room» вже відображається збережена кімната (рис. 4.36).

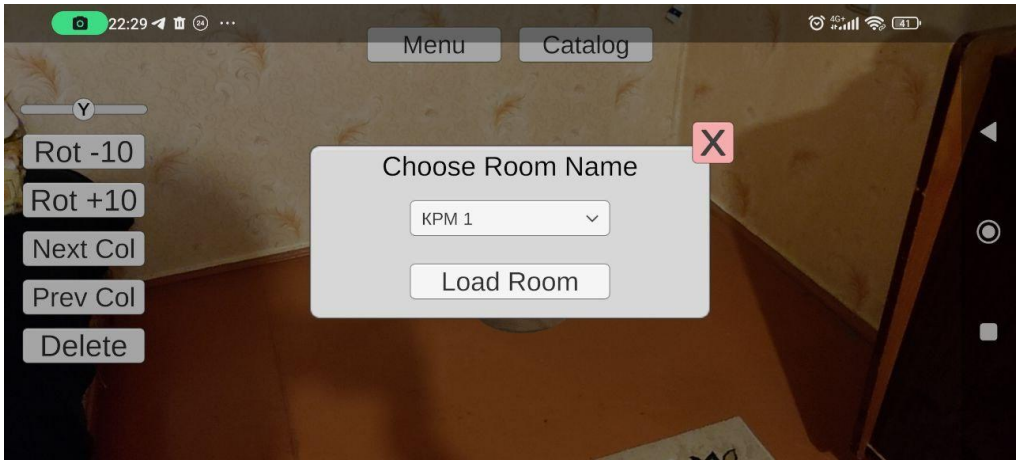

Рисунок 4.36 – Процес завантаження кімнати

Натискаємо кнопку Load Room та дивимось на результат (рис. 4.37).

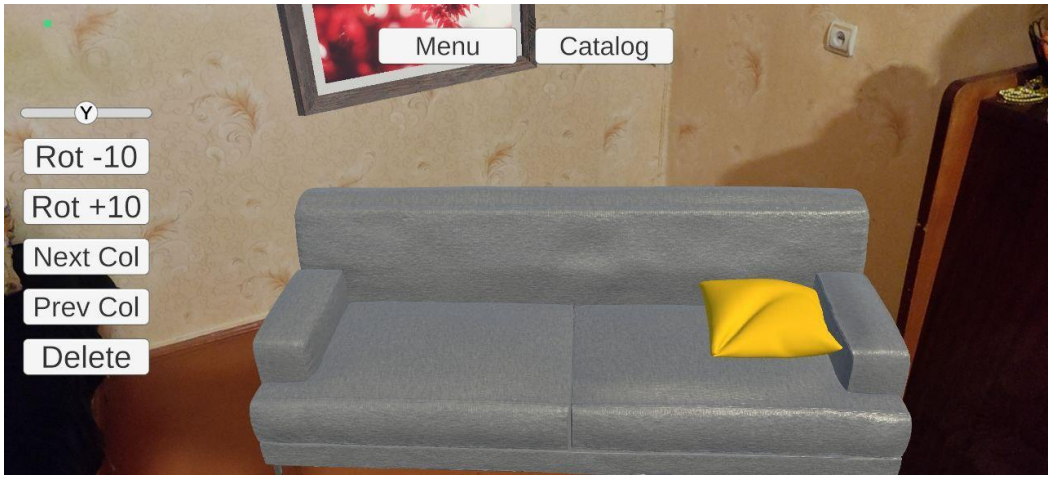

Рисунок 4.37 – Результат завантаження кімнати

У результаті можна побачити, що при завантаженні кімнати позиціювання майже співпадає з тим, як було раніше, хоча усі об’єкти трохи висунулися вперед. Таке буває через складність технології доповненої реальності. В залежності від яскравості світла, сканування навколишнього середовища може кульгати точність відтворення. В цілому це не є критичним.

Тепер, можна детальніше підійти до дивану із подушкою, щоб краще їх роздивитись (рис. 4.38) та можна побачити, що диван із подушка не віддаляються від камери.

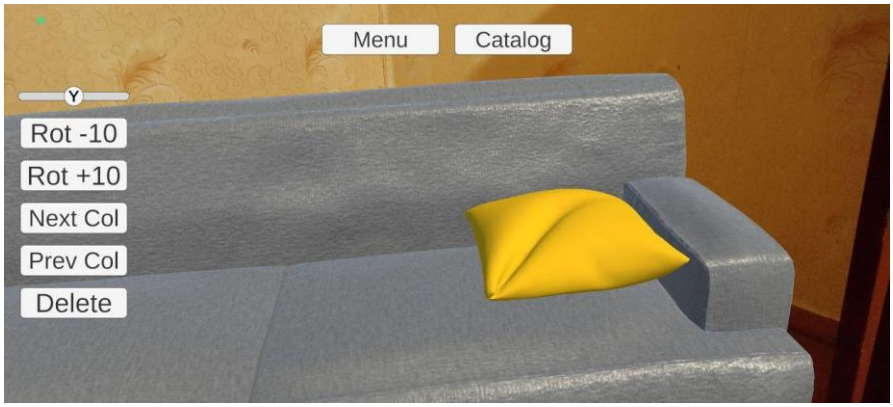

Рисунок 4.38 – Перевірка чи віддаляється диван та подушка

Остання функція яка ще не була продемонстрована – це видалення об’єкту. Обираємо диван, та натискаємо «Delete» (рис. 4.39).

Рисунок 4.39 – Результат видалення дивану

Результатом є те, що подушка тепер «літає», а диван зник разом із панеллю налаштування.

Висновки до розділу 4

У цьому розділі було реалізовано програмний прототип мобільного AR-застосунку для планування інтер’єру квартири, а також продемонстровано основні етапи його створення, інтеграції та перевірки її функціоналу. Спочатку виконано ознайомлення з базовими можливостями Unity 3D та його ключовими елементами, що сформувало практичну основу для подальшої реалізації функціоналу застосунку.

Далі було побудовано логіку роботи системи, зокрема створено структури для зберігання даних про меблі, реалізовано каталог об’єктів, механізми створення та ініціалізації об’єктів у сцені, взаємодію з ними через вибір користувача, а також функції зміни параметрів, переміщення, обертання та видалення об’єктів. Реалізовано систему збереження і відтворення сцен у форматі JSON та механізм створення скріншотів у форматі PNG.

Приділений час користувацькому інтерфейсу, який поєднує всі функції застосунку та робить їх доступними для користувача. Реалізовано меню керування кімнатами, каталог меблів із динамічним формуванням елементів каталогу, а також панель редагування вибраного об’єкта. Координацію взаємодії UI із функціоналом забезпечує окремий контролер інтерфейсу.

Для реалізації доповненої реальності продемонстровано інтеграцію з AR Foundation та налаштовано необхідні компоненти AR-сцени. Додатково розроблено контролер розміщення об’єктів у AR-середовищі.

Під час тестування продемонстровано повний робочий сценарій: додавання меблів і декору з каталогу, обертання, зміна кольору та матеріалу, зміна положення по осі Y, збереження кімнати, її повторне завантаження, створення скріншоту та видалення об’єктів. Результати перевірки показали працездатність реалізованих функцій і відповідність сформованим вимогам. Виявлені незначні відхилення позиціювання після завантаження сцени пояснюються типовими обмеженнями мобільних AR-систем, що не критично для роботи прототипу та може бути враховано при подальшому вдосконаленні.

ВИСНОВКИ

У ході виконання кваліфікаційної магістерської роботи сформовано комплексне теоретичне та аналітичне підґрунтя для розробки мобільного застосунку доповненої реальності, призначеного для планування інтер’єру та візуалізації меблів у реальному просторі. Проведений аналіз підтвердив актуальність використання AR-технологій у сфері дизайну приміщень, оскільки вони забезпечують новий рівень наочності, дозволяють уникнути помилок при виборі меблів та сприяють прийняттю більш обґрунтованих рішень користувачами.

У першому розділі досліджено предметну область, сучасні напрямки використання технологій доповненої реальності, проведено аналіз існуючих аналогів і визначено цільову аудиторію майбутнього застосунку. Розгляд популярних рішень (IKEA Place, Houzz, Planner 5D, Room Planner) показав, що сучасні сервіси або жорстко прив’язані до конкретних виробників, або орієнтовані на професійне 3D-проєктування, що створює нішу для більш універсального і водночас доступного інструменту. Аналіз аудиторії показав високу потребу у простому, інтуїтивному AR-застосунку з можливістю швидкої і реалістичної візуалізації меблів. Дослідження рушія Unity 3D підтвердило його придатність для створення кросплатформенного прототипу з розширеною підтримкою AR-технологій.

У другому розділі детально проаналізовано технологічні основи системи: алгоритми комп’ютерного зору, побудову простору за допомогою SLAM, визначення площин, обчислення глибини, трекінг руху та розпізнавання освітлення. Розглянуто підходи до підготовки та візуалізації 3D-моделей, зокрема використання форматів GLTF, FBX і OBJ, оптимізацію геометрії та застосування PBR-матеріалів для досягнення фотореалістичності. Проведений огляд ARKit, ARCore та AR Foundation показав, що ці технології забезпечують точне визначення положення об’єктів у реальному просторі,

стабільний трекінг і кросплатформний розвиток, що є ключовими вимогами для мобільного AR-застосунку. Також сформовано специфікацію вимог, яка визначає чітку структуру функціоналу, обмеження та технічні потреби системи.

У третьому розділі виконано моделювання програмної системи за допомогою UML-діаграм: сценаріїв використання, діаграми станів, діаграми класів, діаграми розгортання та діаграми послідовності. Це дозволило детально описати структуру системи, її логіку роботи, можливі стани, взаємодію між компонентами та послідовність виконання ключових операцій у межах AR-сесії. Моделі підтвердили узгодженість вимог і забезпечили чітке уявлення про архітектуру майбутнього застосунку.

У четвертому розділі спочатку описано базові інструменти які є у ігровому рушії Unity 3D. Далі були розписані логіка конструювання застосунку, класи, поля та методи які були сформовані у проєкті. Також розглянуто користувацький інтерфейс, взаємодію коду із елементами користувацького інтерфейсу. Докладно розглянуто інтеграцію доповненої реальності у проєкт, її взаємодію із основною логікою проєкту. У кінці було протестовано та продемонстровано роботу застосунку на реальній кімнаті. Поставлена мета була досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is augmented reality?: вебсайт. URL: https://www.ibm.com/think/topics/augmented-reality (дата звернення: 08.09.2025).

2. Kán P., Kaufmann H. Automatic Interior Design in Augmented Reality Based on Hierarchical Tree of Procedural Rules. *Electronics*. 2021. Vol. 10(3). P. 245. URL: https://www.mdpi.com/2079-9292/10/3/245 (дата звернення: 09.09.2025).

3. Augmented Reality (AR) for Attractions: вебсайт. URL: https://rockpaperreality.com/insights/ar-use-cases/augmented-reality-for-attractions/ (дата звернення: 11.09.2025).

4. Nikitha G.S., Amrutha M.C., Gagana G.T., Sathveeka S. Interior Design Using Augmented Reality. International Research Journal of Modernization in Engineering Technology and Science. 2023. Vol. 5, Issue 5. URL: https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2023/37917/final/fin_irjmets1683191907.pdf (дата звернення: 13.09.2025).

5. Відео: IKEA створила додаток з доповненою реальністю: вебсайт. URL: https://ua.news/ua/technologies/video-ikea-stvorila-dodatok-z-dopovnenoyu-realnistyu (дата звернення: 15.09.2025).

6. Houzz – Home Design & Remodel: вебсайт. URL: https://play.google.com/store/apps/details?id=com.houzz.app&hl=uk (дата звернення: 23.09.2025).

7. Planner 5D: AI Home Design: вебсайт. URL: https://play.google.com/store/apps/details?id=com.planner5d.planner5d&hl=uk. (дата звернення: 27.09.2025)

8. Home Planner: House Design AI: вебсайт. URL: https://play.google.com/store/apps/details?id=com.icandesignapp.all. (дата звернення: 28.09.2025)

9. Chang Y. S., Hu K. J., Chiang C. W., Lugmayr A. Applying Mobile Augmented Reality (AR) to Teach Interior Design Students in Layout Plans: Evaluation of Learning Effectiveness Based on the ARCS Model of Learning Motivation Theory. *Sensors*. 2020. Vol. 20(1). P. 105. DOI: https://doi.org/10.3390/s20010105 (дата звернення: 06.10.2025).

10. Augmented Reality (AR) App & Game Development Solution: вебсайт. URL: https://unity.com/solutions/xr/ar (дата звернення: 15.12.2025).

11. SLAM (Simultaneous Localization and Mapping): вебсайт. URL: https://unity.com/glossary/slam (дата звернення: 09.10.2025).

12. Billinghurst M., Clark A., Lee G. Augmented Reality: An Emerging Technologies Guide to AR. Pearson, 2021. 360 p. (дата звернення: 11.10.2025)

13. GLTF vs OBJ vs FBX: Choosing the Right 3D File Format: вебсайт. URL: https://www.alpha3d.io/kb/3d-modelling/gltf-vs-obj/ (дата звернення: 15.10.2025).

14. ARKit: ARKit Definition: вебсайт. URL: https://unity.com/glossary/arkit (дата звернення: 21.10.2025).

15. ARCore: ARCore Definition: вебсайт. URL: https://unity.com/glossary/arcore (дата звернення: 23.10.2025).

16. AR Foundation (документація): вебсайт. URL: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.0/manual/index.html (дата звернення: 25.10.2025).

17. Діаграма станів: вебсайт. URL: https://emtd.ztu.edu.ua/view/word-3021/0 (дата звернення: 29.10.2025).

18. Презентація "Діаграми UML. Діаграми прецедентів": вебсайт. URL: http://naurok.com.ua/prezentaciya-diagrami-uml-diagrami-precedentiv-238715.html (дата звернення: 07.11.2025).

19. Діаграма послідовності (Sequence Diagrams): вебсайт. URL: https://www.maxzosim.com/sequence-diagrams/ (дата звернення: 15.11.2025).

20. Діаграма розгортання: Підручник з UML із ПРИКЛАДОМ: вебсайт. URL: https://www.guru99.com/uk/deployment-diagram-uml-example.html (дата звернення: 17.11.2025).

21. The Editor interface (Unity Manual): вебсайт. URL: https://docs.unity3d.com/Manual/unity-editor.html (дата звернення: 21.11.2025).

22. Creating Scenes / Introduction to scenes (Unity Manual): вебсайт. URL: https://docs.unity3d.com/Manual/CreatingScenes.html (дата звернення: 28.11.2025).

23. Creating scripts (Unity Manual): вебсайт. URL: https://docs.unity3d.com/Manual/creating-scripts.html (дата звернення: 01.12.2025).

24. Prefabs (Unity Manual): вебсайт. URL: https://docs.unity3d.com/Manual/Prefabs.html (дата звернення: 02.12.2025).

25. Models (Unity Manual): вебсайт. URL: https://docs.unity3d.com/Manual/models.html (дата звернення: 03.12.2025).

26. AR Session component | AR Foundation: вебсайт. URL: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/features/session.html (дата звернення: 05.12.2025).

27. XR Origin (Unity Manual): вебсайт. URL: https://docs.unity3d.com/6000.2/Documentation/Manual/xr-origin.html (дата звернення: 07.12.2025).

28. AR plane manager | AR Foundation: вебсайт. URL: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/plane-manager.html (дата звернення: 08.12.2025).

29. AR Raycast Manager | AR Foundation: вебсайт. URL: https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/raycast-manager.html (дата звернення: 08.12.2025).

ВИСНОВКИ

Програмний код

```
// Клас FurnitureInstance
private void Awake()
    {
        if (targetRenderers == null || targetRenderers.Count == 0)
            targetRenderers.AddRange(GetComponentsInChildren<Renderer>());

        if (data != null)
        {
            if (data.materials != null && data.materials.Count > 0)
                ApplyMaterial(data.materials[currentMaterialIndex]);

            if (data.colors != null && data.colors.Count > 0)
                ApplyColor(data.colors[currentColorIndex]);
        }
    }
    public void Init(FurnitureData furnitureData)
    {
        data = furnitureData;
        currentMaterialIndex = 0;
        currentColorIndex = 0;

        if (data != null)
        {
            if (data.materials != null && data.materials.Count > 0)
                ApplyMaterial(data.materials[currentMaterialIndex]);

            if (data.colors != null && data.colors.Count > 0)
                ApplyColor(data.colors[currentColorIndex]);
        }
    }
    public void SetPosition(Vector3 position)
    {
        transform.position = position;
    }
    public void Rotate(float degrees)
    {
        transform.Rotate(Vector3.up, degrees, Space.World);
    }
    public void NextMaterial()
    {
        if (data == null || data.materials == null || data.materials.Count == 0) return;

        currentMaterialIndex = (currentMaterialIndex + 1) % data.materials.Count;
        ApplyMaterial(data.materials[currentMaterialIndex]);
    }
    public void PreviousMaterial()
    {
        if (data == null || data.materials == null || data.materials.Count == 0) return;

        currentMaterialIndex--;
        if (currentMaterialIndex < 0)
            currentMaterialIndex = data.materials.Count - 1;

        ApplyMaterial(data.materials[currentMaterialIndex]);
    }
    private void ApplyMaterial(Material mat)
```

```
    {
        if (mat == null) return;

        foreach (var r in targetRenderers)
            r.material = mat;
    }
    public void NextColor()
    {
        if (data == null || data.colors == null || data.colors.Count == 0) return;

        currentColorIndex = (currentColorIndex + 1) % data.colors.Count;
        ApplyColor(data.colors[currentColorIndex]);
    }
    public void PreviousColor()
    {
        if (data == null || data.colors == null || data.colors.Count == 0) return;

        currentColorIndex--;
        if (currentColorIndex < 0)
            currentColorIndex = data.colors.Count - 1;

        ApplyColor(data.colors[currentColorIndex]);
    }
    private void ApplyColor(Color color)
    {
        foreach (var r in targetRenderers)
        {
            var matInstance = r.material;
            matInstance.color = color;
        }
    }
    public void Delete()
    {
        Destroy(gameObject);
    }
    public void SetMaterialIndex(int index)
    {
        if (data == null || data.materials == null || data.materials.Count == 0) return;
        index = Mathf.Clamp(index, 0, data.materials.Count - 1);
        currentMaterialIndex = index;
        ApplyMaterial(data.materials[currentMaterialIndex]);
    }
    public void SetColorIndex(int index)
    {
        if (data == null || data.colors == null || data.colors.Count == 0) return;
        index = Mathf.Clamp(index, 0, data.colors.Count - 1);
        currentColorIndex = index;
        ApplyColor(data.colors[currentColorIndex]);
    }


// Клас RoomSaveSystem

public void SaveRoom(string roomName)
    {
        if (string.IsNullOrEmpty(roomName))
        {
            Debug.LogWarning("RoomSaveSystem: roomName is empty, not saving.");
            return;
        }

        FurnitureInstance[] instances = FindObjectsOfType<FurnitureInstance>();
        RoomSaveData roomData = new RoomSaveData
```

```
        {
            roomName = roomName
        };

        foreach (var inst in instances)
        {
            if (inst == null || inst.data == null) continue;

            FurnitureItemSaveData itemData = new FurnitureItemSaveData
            {
                furnitureId = inst.data.id,
                position = inst.transform.position,
                rotation = inst.transform.rotation,
                materialIndex = inst.CurrentMaterialIndex,
                colorIndex = inst.CurrentColorIndex
            };

            roomData.items.Add(itemData);
        }

        Directory.CreateDirectory(RoomsFolderPath);
        string safeName = GetSafeFileName(roomName);
        string fullPath = Path.Combine(RoomsFolderPath, safeName + ".json");

        string json = JsonUtility.ToJson(roomData, true);
        File.WriteAllText(fullPath, json);

        Debug.Log($"RoomSaveSystem: room '{roomName}' saved to {fullPath}");
    }
    public void LoadRoom(string roomName)
    {
        if (string.IsNullOrEmpty(roomName))
        {
            Debug.LogWarning("RoomSaveSystem: roomName is empty, cannot load.");
            return;
        }

        string safeName = GetSafeFileName(roomName);
        string fullPath = Path.Combine(RoomsFolderPath, safeName + ".json");

        if (!File.Exists(fullPath))
        {
            Debug.LogWarning($"RoomSaveSystem: file for room '{roomName}' not found at
path {fullPath}");
            return;
        }

        string json = File.ReadAllText(fullPath);
        RoomSaveData roomData = JsonUtility.FromJson<RoomSaveData>(json);

        FurnitureInstance[] existingInstances = FindObjectsOfType<FurnitureInstance>();
        foreach (var inst in existingInstances)
        {
            if (inst != null)
                Destroy(inst.gameObject);
        }

        if (roomData == null || roomData.items == null) return;

        foreach (var item in roomData.items)
        {
            FurnitureData furnitureData = FindFurnitureDataById(item.furnitureId);
            if (furnitureData == null)
```

```
            {
                Debug.LogWarning($"RoomSaveSystem: FurnitureData with id
'{item.furnitureId}' not found in catalog.");
                continue;
            }

            FurnitureInstance instance = spawner.SpawnFurniture(furnitureData);
            if (instance == null) continue;

            instance.transform.position = item.position;
            instance.transform.rotation = item.rotation;

            if (item.materialIndex >= 0)
                instance.SetMaterialIndex(item.materialIndex);

            if (item.colorIndex >= 0)
                instance.SetColorIndex(item.colorIndex);
        }

        Debug.Log($"RoomSaveSystem: room '{roomName}' loaded.");
    }
    public List<string> GetSavedRoomNames()
    {
        List<string> result = new List<string>();

        if (!Directory.Exists(RoomsFolderPath)) return result;

        string[] files = Directory.GetFiles(RoomsFolderPath, "*.json");
        foreach (string file in files)
        {
            string name = Path.GetFileNameWithoutExtension(file);
            result.Add(name);
        }

        return result;
    }
    private string GetSafeFileName(string roomName)
    {
        foreach (char c in Path.GetInvalidFileNameChars())
            roomName = roomName.Replace(c, '_');
        return roomName;
    }
    private FurnitureData FindFurnitureDataById(string id)
    {
        if (catalog == null) return null;

        foreach (var item in catalog.Items)
        {
            if (item != null && item.id == id)
                return item;
        }

        return null;
    }
```