

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ  
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ  
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

\_\_\_\_\_ І. А. Жуков  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2020 р.

**ДИПЛОМНА РОБОТА**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

ЗА СПЕЦІАЛЬНІСТЮ 123 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

Тема: Використання доповненої реальності для навчального процесу

Виконавець: студентка КС231М Роботницька Тетяна Петрівна  
(студент, група, прізвище, ім'я, по батькові)

Керівник: к.т.н., доцент Малярчук Василь Олександрович  
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: \_\_\_\_\_ Малярчук. В.О.  
(підпис) (ПІБ)

Засвідчую, що у дипломній роботі немає  
запозичень праць інших авторів без  
відповідних посилань

Студент \_\_\_\_\_ Роботницька Т.П.  
(підпис) (ПІБ)

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних систем та мереж

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

                                 І.А. Жуков  
(підпис)

«        »                  2020 р.

**ЗАВДАННЯ**

**на виконання дипломної роботи (проекту)**

Роботницької Тетяни Петрівни

(прізвище, ім'я, по батькові)

1. Тема роботи: Використання доповненої реальності для навчального процесу

затверджена наказом ректора від «25» вересня 2020 року №1793/ст

2. Термін виконання проекту (роботи): з 05.10.2020р. по 30.12.2020р.

3. Вихідні дані до роботи: лекційні матеріали та матеріали для проведення лабораторних робіт із дисципліни мікроелектроніка, документація сервісу AR sdk та середовищ розробки

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1. Аналіз технології доповненої реальності

2. Реалізація мобільного додатку «AR Довідник»

3. Дослідження ефективності використання технології доповненої реальності для покращення якості навчального процесу

5. Перелік обов'язкового графічного матеріалу:

Презентація *Power Point*

## 6. Календарний план-графік

№ п/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Постановка задачі для дипломної роботи	25.09.20	
2	Ознайомлення із спеціальною літературою та технічною документацією	05.10.20	
3	Аналіз технології доповненої реальності, середовищ розробки програм, сервісів для втілення доповненої реальності в програми	15.10.20	
4	Розробка мобільного додатку із використанням технології доповненої реальності	28.10.20	
5	Аналіз ефективності використання технології доповненої реальності в навчальному процесі	09.11.20	
6	Оформлення пояснювальної записки	10.12.20	
7	Підготовка графічного демонстраційного матеріалу	11.12.20	

7. Дата видачі завдання «25» вересня 2020 р.

Керівник дипломної роботи

\_\_\_\_\_ (підпис)

Малярчук В.О.

(П.І.Б.)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис студента)

Роботницька Т.П.

(П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Використання доповненої реальності для навчального процесу»: 145 ст., 61 рис., 3 табл., 58 літературних джерел.

ДОПОВНЕНА РЕАЛЬНІСТЬ, КАМЕРА, ДОДАТОК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СЕРЕДОВИЩЕ РОЗРОБКИ, *SDK*

**Об'єкт дослідження** – технологія доповненої реальності.

**Предмет дослідження** – доповнена реальність для навчального процесу.

**Мета дипломної роботи** – розробити додаток із використанням технології доповненої реальності для навчального процесу та провести дослідження ефективності використання такого інструменту для навчального процесу.

**Метод дослідження** – аналіз ефективності використання програмного забезпечення із імплементованою технологією доповненої реальності для навчального процесу.

Дипломна робота сприяє зацікавленню студентів у навчальному процесі та поглибленому вивченні та використанні технології доповненої реальності, що швидко розвивається та популяризується. Розроблене програмне забезпечення рекомендується до використання у вищих навчальних закладах у якості об'єкту для зацікавлення чи для підказки по дисципліні Мікроелектроніка.



## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.....	11
1.1. Аспекти розробки із технологією доповненої реальності .....	12
1.1.1. Камери глибини.....	13
1.1.2. Типи доповненої реальності .....	19
1.1.3. Пристрої для відтворення доповненої реальності.....	21
1.1.4. Виклики індустрії доповненої реальності .....	22
1.1.5. Сфери застосування <i>AR</i> .....	24
1.2. Аналіз процесу розробки програмного забезпечення.....	25
1.2.1. Етапи процесу розробки програмного забезпечення .....	26
1.3. Аналіз середовищ для розробки програм із доповненою реальністю	28
1.3.1. Аналіз ігрового рушія для розробки програм <i>Unity</i> .....	28
1.3.2. Аналіз ігрового рушія для розробки програм <i>Unreal Engine</i> ..	30
1.4. Аналіз наборів із засобів розробки, утиліт і документації для використання технології доповненої реальності у проектах .....	32
1.4.1. Набір засобів розробки доповненої реальності <i>ARKit</i> від <i>Apple</i>	32
1.4.2. Набір утиліт для використання доповненої реальності <i>AR Core</i> від <i>Google</i> .....	33
1.4.3. Набір засобів розробки доповненої реальності <i>Easy AR</i> .....	34
1.4.4. Набір засобів розробки доповненої реальності <i>Vuforia</i> .....	35
1.4.5. Набір засобів розробки доповненої реальності <i>MAXST AR SDK</i>	36
1.4.6. Набір засобів розробки доповненої реальності <i>Wikitude</i> .....	37
Висновки за розділом.....	38
РОЗДІЛ 2 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ «AR ДОВІДНИК» .....	40
2.1. Створення ідеї та встановлення основного посилу проекту .....	40
2.2. Створення користувацького інтерфейсу програми .....	43

2.3. Створення зображень-тригерів для відтворення контенту в доповненій реальності .....	52
2.4. Інструкція для використання сервісу <i>MAXST AR SDK</i> .....	56
2.5. <i>3D</i> контент додатку .....	62
2.6. Розробка програмної частини .....	64
Висновки за розділом.....	76
<b>РОЗДІЛ 3 ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ В НАВЧАЛЬНОМУ ПРОЦЕСІ.....</b>	<b>77</b>
3.1. Дослідження ефективності використання технології доповненої реальності для навчального процесу .....	77
3.2. Дослідження впливу використання <i>2D</i> , <i>3D</i> та <i>AR</i> технологій у навчальному процесі .....	85
Висновки за розділом.....	87
<b>СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>91</b>
<b>ДОДАТКИ.....</b>	<b>97</b>
Додаток А.....	97
Додаток Б .....	98
Додаток В.....	106
Додаток Г .....	107

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>AR</i>	–	<i>Augmented reality</i> , Доповнена реальність
<i>VR</i>	–	Віртуальна реальність
<i>3D</i>	–	<i>3 dimensions</i>
<i>API</i>	–	<i>Application Programming Interface</i> , прикладний програмний інтерфейс
<i>Unity</i>	–	Багатофункціональний інструмент для розробки програм
<i>MAXST</i>	–	сервіс, що надає у використанні один із топових <i>AR SDK</i>
<i>SDK</i>	–	<i>Software Development Kit</i> , набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи
<i>UI</i>	–	<i>User interface</i>
<i>AR камера</i>	–	камера із можливістю переглядати контент у форматі доповненої реальності
Тригер	–	<i>target image</i> , зображення, яке комп'ютер навчений розпізнавати за допомогою <i>AR SDK</i>
Скріншот	–	знімок екрана, зображення, отримане комп'ютером, що зображує дійсно те, що бачить користувач на екрані монітора

## ВСТУП

Розвиток технологій у 21 столітті є невимовно швидким. Кожен рік на світових технологічних конференціях презентується величезна кількість нових речей. Технологій настільки багато і їхній масштаб настільки великий, що людство потрібно готувати до прийому новинок. Саме для активного розвитку усього суспільства, навчальну систему необхідно постійно вдосконалювати, щоб вона встигала готувати спеціалістів гідних свого часу.

Уже встигли нашуміти технології доповненої та віртуальної реальності, вони отримали визнання в різного роду сферах від будівництва, маркетингу та до медицини. Зараз цими технологіями частіше користуються в розважальних цілях або щоб отримати саме «Вау»-ефект, хоча, на мою думку, за ними майбутнє навчання пересічного жителя. Та для такої мети потрібно заздалегідь готувати як спеціалістів, так і користувачів. Потрібен ще більший розголос технологіям.

В даній роботі поговоримо більше саме про доповнену реальність.

Доповнена реальність (*Augmented reality, AR*) – це доповнення фізичного світу за допомогою цифрових даних, яке забезпечується комп'ютерними пристроями (смартфонами, планшетами та окулярами *AR*) в режимі реального часу. Простіше кажучи, на відміну від віртуальної реальності (*VR*), яка вимагає повного занурення у віртуальне середовище, *AR* використовує середовище навколо нас та просто накладає поверх нього певну частинку віртуальної інформації, наприклад графіку, звуки та реакцію на дотики. Оскільки віртуальний та реальний світи гармонійно співіснують, користувачі з досвідом доповненої реальності мають змогу спробувати цілком новий, покращений світ, де віртуальна інформація використовується як додатковий корисний інструмент, що забезпечує допомогу в повсякденній діяльності. Додатки, що використовують *AR*, можуть бути таким же простими як, наприклад, швидкі текстові повідомлення, або ж настільки складними, як інструкція для виконання надскладної хірургічної операції. Вони можуть виділяти певні моменти, розширювати розуміння різноманітних речей, або ж надавати доступні та миттєві дані.

Одна із суперсил *AR* – обмін знаннями. При порівнянні теорії гравітації з чорними дірами напрошується висновок, що теоретично ми більш обізнані про гравітацію, тому що ми можемо випробувати її на відміну від чорної діри, яку ми можемо тільки спостерігати [1].

*AR* має великий потенціал для маркетингу, оскільки передбачає, що користувач повністю занурений в новий досвід. Відомо, що повне залучення призводить до більш високого ступеня конверсії. Користувач, швидше за все, прийме рішення про покупку після того, як сам спробує продукт.

Актуальність теми дипломної роботи. Наявність у користувачів реального досвіду, замість спостереження може різко збільшити шанси на розуміння і запам'ятовування ними інформації. Це те, що робить *AR* таким привабливим середовищем для навчання і тренувань, не кажучи вже про здатність бути вільними від будь-яких фізичних обмежень і перешкод, особливо таких, як процеси під час дистанційного навчання.

Метою дипломної роботи є дослідження методів реалізації та розробка продукту з технологією доповненої реальності, що стане в нагоді під час навчального процесу та дослідження ефективності використання такого інструменту для навчального процесу.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати технологію доповненої реальності, її види, можливості, обмеження;
- проаналізувати існуючі платформи розробки програм і обрати найоптимальніший варіант;
- проаналізувати існуючі *AR SDK* і обрати найоптимальніший для роботи варіант;
- розробити архітектуру мобільного застосунку для навчального процесу із використанням технології доповненої реальності;
- розробити елементи інтерфейсу для застосунку;
- втілити в додаток технологію *3D*, з використанням *3D* моделей для використання у доповненій реальності;

– розробити додаток на основі висновків аналізів і з використанням технології доповненої реальності для організації навчального процесу;

– провести дослідження ефективності використання інструменту із технологією доповненої реальності для навчального процесу.

Мобільний застосунок із реалізованим прикладом роботи технології доповненої реальності.

Доповнена реальність безумовно є технологією за якою майбутнє. Саме тому, допомогти студентам увійти у середовище роботи із доповненою реальністю є досить вагомим у наш час. Розроблений продукт можна буде використовувати під час навчального процесу, наприклад на лабораторних роботах.

## РОЗДІЛ 1

### АНАЛІЗ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Доповнена реальність (*AR*) – це різновид віртуального середовища або віртуальної реальності, як її частіше називають. Технології *VR* повністю занурюють користувача в синтетичне середовище. Користувач не може бачити реальний світ навколо себе, поки він занурений повністю у віртуальну реальність. На відміну від *VR*, *AR* дозволяє користувачеві бачити реальний світ, тобто віртуальні об'єкти накладаються на реальний світ або інтегруються в нього. Тому *AR* доповнює реальність, а не повністю її замінює. В ідеальному випадку користувачеві здається, що віртуальні та реальні об'єкти співіснують в одному просторі. *AR* можна розглядати як «золоту середину» між віртуальним середовищем (повністю синтетичним) та телеприсутністю (повністю реальним).

Визначення *AR*, яке і сьогодні використовується як визначення технології, було описане Азумою [3] і звучить так:

*AR* як технологія:

1. поєднує реальні та віртуальні об'єкти,
2. відтворюється в реальному часі та є інтерактивною, і
3. реєструє/розміщує віртуальні об'єкти в реальному середовищі.

Згідно до проведених раніше досліджень та оцінки існуючих технологій [3,4], які показують, що системи *AR* ефективно відповідають очікуванням, коли є напрацювання на відповідних рівнях для таких компонентів, як:

- Апаратне та програмне забезпечення для відтворення графіки, що здатне створити перекриття реального контексту віртуальним.
- Адекватні методи відстеження, щоб належним чином відображати зміни від точки зору користувача до відтвореної графіки.
- Точна синхронізація калібрування та реєстрації трека.
- Синхронізація реального та віртуального подання, коли вигляд користувача виправлений.

- Дисплей повинен в достатній мірі поєднувати зображення віртуальних об'єктів з виглядом реальних компонентів.
- Комп'ютерна обробка: підтримка процесів *AR*-симуляції, що працює на апаратних пристроях вводу-виводу.
- Методи взаємодії, що описують як користувачі можуть взаємодіяти із віртуальним вмістом *AR*.

Кілька другорядних тем також є важливими, залежно від того, яка конкретна задача для *AR* розглядається. Повинні бути оцінені можливість застосування, пристосованість мобільних / портативних пристроїв, методи візуалізації, пристрої авторизації, багатомодальність входів *AR*, методи візуалізації, архітектура програмного забезпечення тощо. Питання інтеграції з обладнанням та реалізації програмного забезпечення повною мірою враховуються безпосередньо під час розробки самих *AR* додатків.

Доповнена реальність (*AR*) є однією з найбільших технологічних тенденцій зараз, і вона буде лише зростати, оскільки смартфони та інші пристрої, які підтримують технологію *AR*, стають все доступнішими по всьому світу. *AR* дозволяє бачити реальне середовище прямо перед нами - дерева, що хитаються від вітру в парку, собак, що ганяються за м'ячами, дітей, що грають у футбол, і накладене цифрове зображення поверх реального. Наприклад, птеродактиль може бути помічений при посадці на дерева, реальних собак можуть розбавити їхні колеги з мультфільмів, а дітей можна побачити, як вони б'ються повз чужорідний космічний корабель на своєму шляху, щоб забити гол у ворота суперників. З перевагами та можливостями *AR* технології вищесказане зовсім не вважатиметься абсурдом.

### **1.1. Аспекти розробки із технологією доповненої реальності**

Для *AR* можна використовувати певний діапазон даних (зображення, анімація, відео, 3D-моделі), і люди бачитимуть результат як у природному, так і в синтетичному світлі. Крім того, користувачі усвідомлюють, що перебувають у реальному світі, який вдосконалений комп'ютерним зором, на відміну від *VR*.



*AR* може відображатися на різних пристроях: екранах, окулярах, портативних пристроях, мобільних телефонах, головних дисплеях. Він включає такі технології, як *S.L.A.M.* (одночасна локалізація та картографування), відстеження глибини (коротко, дані датчика, що обчислюють відстань до об'єктів) та наступні компоненти:

– Камери та датчики. Збір даних про взаємодію користувача та надсилання їх на обробку. Камери на пристроях сканують навколишнє середовище, і за допомогою цієї інформації пристрій визначає фізичні об'єкти та створює *3D*-моделі. Це можуть бути камери спеціального режиму, як у *Microsoft HoloLens*, або звичайні камери смартфонів для фотографування / відео.

– Обробка. Зрештою *AR*-пристрої повинні діяти як маленькі комп'ютери, що сучасні смартфони вже роблять. Таким же чином їм потрібен процесор, графічний процесор, флеш-пам'ять, оперативна пам'ять, *Bluetooth / WiFi, GPS* тощо, щоб мати можливість вимірювати швидкість, кут, напрямок, орієнтацію в просторі тощо.

– Проекція. Це стосується мініатюрного проектора на гарнітурах *AR*, який бере дані з датчиків та проектує цифровий вміст (результат обробки) на поверхню для перегляду. Насправді використання проєкцій в *AR* ще не повністю винайдено для використання в комерційних продуктах чи послугах.

– Рефлексія. Деякі пристрої *AR* мають дзеркала, які допомагають людським очам переглядати віртуальні зображення. У деяких є «набір маленьких вигнутих дзеркал», а в інших - двостороннє дзеркало, яке відбиває світло на камеру та на око користувача. Метою таких шляхів відображення є правильне вирівнювання зображення.

### 1.1.1. Камери глибини

Краще зрозуміти основи камер з глибинним зондуванням і дізнатися, як і де кожен клас використовується в галузі доповненої реальності задля кращого розуміння обмежень у використанні мобільних телефонів для *AR*. Розглянемо весь спектр глибинних камер - таких, що базуються на структурованому світлі (*structured light*,

*SL*), пасивному стерео (*passive stereo, PS*), активному стерео (*active stereo, AS*) та часі польоту (*time of flight, ToF*) зображені на рис. 1.1.

Хоч глибинні камери (камери з можливістю захоплення тривимірних зображень) існують вже кілька десятиліть, загалом громадськість вперше потрапила під цю технологію в 2010 році, коли було представлено *Microsoft Kinect* - аксесуар для *XBOX 360*. На жаль, як і в багатьох сучасних продуктів, і, незважаючи на вражаючі можливості, широке висвітлення в пресі та значну фінансову підтримку, впровадження *Kinect* було обмеженим, що призвело до його нещодавнього припинення. Зовсім недавно ми бачили подібну технологію, інтегровану в *iPhone X* від *Apple* - передня панель датчиків, що забезпечує надійне відстеження обличчя та розблокування пристрою.

Хоча наведені вище приклади є явно новими та цікавими, їх часто вважають дещо хитрими тим, хто більше знайомий з величезними можливостями глибинних камер для мобільної доповненої реальності (*AR*). Більш доречно, такі високоякісні носимі пристрої, як *Microsoft HoloLens*, *Meta 2* та *DAQRI Smart Glasses®*, містять датчики глибини не лише для розваги, а для розуміння середовища, яке для *AR*-носіїв важко досягти за допомогою інших типів датчиків, і вони є важливими для природного введення даних користувачем за допомогою вільних рук (наприклад, жестів).

З огляду на це, важливо оцінити і зрозуміти дивовижно різноманітну глибину камер, починаючи від споживчих рішень, як у згаданому *iPhone X*, і закінчуючи великими, високоточними та дорогими промисловими рішеннями, як у *Mantis Vision F6* або *Ensenso* камери від *IDS Imaging*.

В даний час терміни структурованого світла та активного стерео не визначені на 100% чітко. Однак розглянемо структуроване світло як комбінацію однієї камери та одного проектора, коли проектор проектує відомий шаблон на відміну від активного стерео, який використовує дві камери та проектор, що проектує невідомий (випадковий) шаблон (рис. 1.1.).

Як структуроване світло, так і активне стерео, а також пасивне стерео - це всі рішення на основі стерео - це означає, що оцінки глибини базуються на виявленні

особливостей з двох різних точок зору, щоб їх можна було триангулювати. Саме це неявно дає нам глибину, відстань точки від камери вздовж напрямку огляду камери. Важливо, що оцінка глибини є стерео через дві точки зору, необхідні для оцінки глибини, а не кількість камер (типова помилкова думка). У разі структурованого світла проектор представляє одну точку зору, і, завдяки відомому проектованому візерунку, функції слід знаходити лише в одному виді камери. Активні та пасивні стерео рішення, навпаки, повинні відповідати характеристикам однієї камери іншій. Тільки "Час польоту" не покладається на стерео принципи і безпосередньо вимірює глибину (рис. 1.1.).

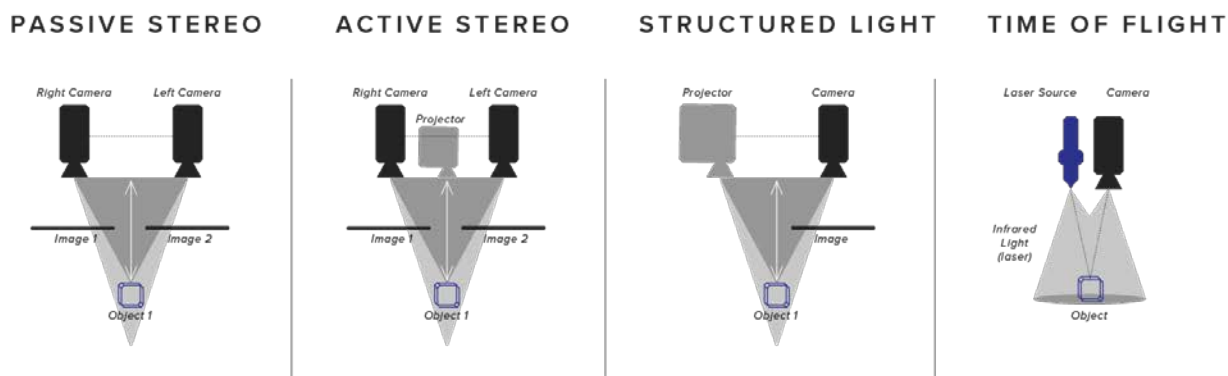


Рис. 1.1. Варіанти камер глибини

Вимог до мобільних камер глибини є чимало. По-перше, датчики повинні бути дуже маленькими, щоб інтегрувати їх у гарнітури порівняно обмеженого розміру. Для AR гарнітур необхідні невеликі за розміром, датчики класу мобільних телефонів, наприклад, модуль камери товщиною не більше 5 мм. По-друге, камера глибини повинна використовувати якомога менше енергії, в ідеалі щось менше 500 мВт, оскільки загальна здатність тепловіддачі середньої гарнітури становить лише кілька ват. По-третє, для подальшої економії енергії глибинна камера не повинна вимагати інтенсивної обробки вихідного сигналу датчика, оскільки це призведе до подальшого споживання енергії.

Сьогодні серед двох відомих випадків використання глибинних камер у мобільному AR є сканування навколишнього середовища (наприклад, реконструкція) та введення даних користувачем (наприклад, розпізнавання жестів). Ці два випадки

використання мають переважно різні вимоги. Для сканування навколишнього середовища глибинна камера повинна бачити якомога далі - на практиці це приблизно діапазон приблизно від 60 см до 5 метрів. На відміну від цього, введення даних користувачем має працювати лише на відстані витягнутої руки, отже, діапазон становить від 20 до 100 см. Таким чином, загалом діапазон складає від 20 см до 5 метрів, співвідношення 25х. На жаль, такий широкий діапазон сьогодні проблематичний для більшості камер глибини. Подібним чином, рішення на основі стерео, що забезпечує значущу точність на 5 метрів, вимагає великої базової лінії, що одночасно робить надзвичайно складним пошук відповідностей зблизька. На щастя, ці два діапазони зазвичай не повинні підтримуватися одночасно; хоча для глибинної камери може бути вигідно підтримувати псевдоодносні режими, що чергуються між режимами короткого та далекого діапазону на основі кожного кадру.

Також є питання калібрування. Як і будь-який інший датчик, камери глибини повинні бути відкалібровані для належної роботи. Мобільні пристрої зазнають великих фізичних навантажень протягом декількох років використання, що може призвести до втрати заводських калібрувань.

Як зазначалося вище, структуроване світло проектує відомий шаблон за допомогою проектора (див. рис. 1.1). Проектор і камера утворюють стереосистему, отже, друга камера не потрібна. Крім того, системи структурованого світла зазвичай працюють в інфрачервоному спектрі, так що картина не видна людському оку. З міркувань безпеки проектор повинен мати обмежену потужність проєкції, що зазвичай призводить до відстані проєкції до 4–5 метрів. Зрозуміло, що між стійкістю та точністю існує компроміс. Наприклад, навіть незважаючи на те, що *Kinect v1* постачав карти глибини *VGA*, він проектував шаблон лише з 30000 крапок. Отже, він також міг обчислити лише близько 30000 значень глибини, але використовував розумну фільтрацію для заповнення інших 90% записів карти глибини. Плюсом є те, що, проектуючи лише невелику кількість точок, проектор може використовувати відносно велику кількість енергії на точку. Як результат, цей датчик був досить міцним до складних матеріалів та умов освітлення.

Як і у всіх рішень на основі стереосистеми, структуроване світло може постраждати від структурної цілісності: якщо базову лінію змінити - наприклад, через тиск на пристрій - якість глибини може швидко погіршитися до неприйняттого рівня. Для досягнення високоякісних вимірювань до 5 м потрібна вихідна лінія ~ 8 см. Це добре підходить для планшетів, але може бути складним для мобільних телефонів, а також для гарнітур AR. Структуровані світлові рішення, оптимізовані для сканування середовища, зазвичай мають мінімальний діапазон зондування 50–70 см, що запобігає використанню для відстеження рук.

Підводячи підсумок, структуроване світло не вимірює глибину безпосередньо, а покладається на стереозвук, що є дорогою операцією, що вимагає спеціальних апаратних блоків, для обробки в режимі реального часу з низьким енергоспоживанням.

Часто можна почути аргументи, що пасивне стерео добре працює на відкритому повітрі, де нетекстуровані поверхні рідкісні. Хоч це вірно для випадків використання, таких як самокеровані автомобілі, невелика вихідна лінія менше 10 см обмежує відстань, до якої можна точно виконати триангуляцію відповідних функцій, що не є практичним у більшості випадків використання AR. Хоча, звичайно, можна використовувати більше двох камер, але такі підходи не підходять для мобільних AR через збільшення вартості, розміру та складності обчислень.

Великою перевагою всіх рішень на основі стерео (включаючи структуроване світло) є те, що вони, як правило, мають досить хорошу роздільну здатність. На великій відстані обидві камери бачать переважно однаково, але зблизька дві камери можуть бачити дуже різні частини об'єктів. Оскільки карти глибини зазвичай обчислюються з точки зору однієї камери, це виглядає так, ніби крупний об'єкт кидає тінь на себе там, де відсутні дані про глибину.

У той час як рішення на основі стерео оцінюють глибину за допомогою триангуляції відповідних функцій, камери *Time of Flight (ToF)* вимірюють відстань безпосередньо, тобто пристрій вимірює часовий зсув між сигналом, що надходить випромінювачем, поки він не надійде назад до датчика. Як і рішення на основі стерео, більшість камер *ToF* сьогодні випромінюють інфрачервоне світло на довжині хвилі

850 нм, роблячи його невидимим для людських очей і менше страждають від перешкод більшості штучних джерел світла, що знаходяться в приміщенні.

На основі вимірюючого зміщення часу та використання відомої швидкості світла можна розрахувати відстань. Однак, завдяки величезній швидкості світла, для отримання точних результатів потрібна точність синхронізації в наносекундах. Через технічні проблеми лише декілька пристроїв насправді використовують цей принцип.

Більшість камер *ToF*, які сьогодні використовуються в *AR* та *VR*, вимірюють не час сигналу, а натомість його фазовий зсув, з якого також можна оцінити відстань. Для вимірювання фази випромінюваний сигнал потрібно модулювати. Однак фазу неможливо виміряти безпосередньо, а натомість її слід обчислити принаймні з чотирьох вимірювань модульованого інфрачервоного сигналу. В результаті камери *ToF* записують принаймні чотири так звані підкадри. Потім ці чотири вимірювання на кожному пікселі об'єднують для обчислення фази сигналу на цьому пікселі, від якої в свою чергу можна оцінити відстань.

У цього методу є проблема – для досягнення точних вимірювань фази фаза сигналу повинна бути короткою, але короткі фази можуть обернутися, що означає, що певний зсув фази може бути результатом об'єкта на відстані  $N$ , але також і на відстанях  $2N$ ,  $3N$  тощо.

Однозначною перевагою камер *ToF* є те, що вони можуть бути дуже малими: принцип вимірювання не покладається на триангуляцію, тому не потрібно базової лінії, а випромінювач і датчик можуть бути зібрані настільки тісно, наскільки це дозволяє інтеграція. Крім того, обчислення фази і, отже, відстань від чотирьох вимірювань вимагає набагато менше обробки, ніж відповідність характеристик, як у рішеннях на основі стереосистеми.

Проте камери *ToF* не позбавлені недоліків. Найбільш очевидно, що камери *ToF* сьогодні мають відносно низьку роздільну здатність 100 тис. пікселів і часто навіть нижчу. Крім того, камери *ToF* страждають від ефектів багатопроменевості, що відбувається, якщо сигнали не тільки відображаються безпосередньо назад до датчика, але й здійснюють кілька обертів. Розсіяне світло теж може бути проблематичним. Усі ці проблеми можуть бути вирішені до певної міри, але вони

вимагають особливої обережності при розташуванні датчика або додаткової обробки, тим самим зменшуючи обчислювальні переваги камер *ToF*.

Іншим важливим фактором є вартість. Завдяки величезному успіху кольорових камер у мобільних телефонах, високоякісні модулі камер сьогодні коштують лише ~ 10 доларів. Донедавна не було ринку, який би підтримував масштабне та недороге виробництво датчиків глибини, але, схоже, це змінюється, і очікувано, що ціни на датчики глибини також знизяться [5].

### 1.1.2. Типи доповненої реальності

*AR* на основі маркера – зображення-тригера (маркерний *AR*). Також цей тип називають розпізнаванням зображень, оскільки потрібен спеціальний візуальний об'єкт і камера для його сканування. Це може бути що завгодно, від роздрукованого *QR*-коду до спеціальних знаків, зображень. Пристрій *AR* також обчислює положення та орієнтацію маркера для позиціонування вмісту, в деяких випадках. Таким чином, маркер ініціює цифрову анімацію для перегляду користувачами, і тому зображення в журналі можуть перетворюватися на *3D*-моделі. На рисунку 1.2 можна побачити приклад роботи маркерного *AR*, при наведенні камери девайсу на зображення-тригер на екрані можна спостерігати доповнення реальності за допомогою *3D* об'єкту із його анімаціями.

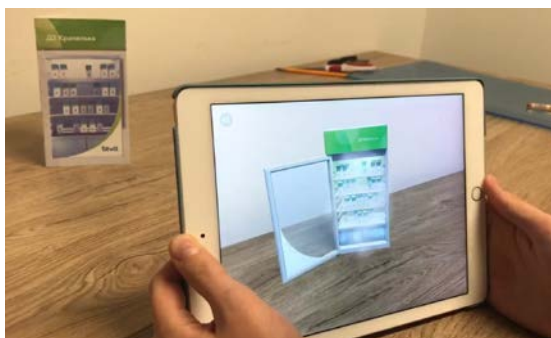


Рис. 1.2. Приклад маркерного *AR*

Безмаркерний *AR* (*S.L.A.M.*) - доповнена реальність, що базується на основі місцезнаходження або позиції, яка використовує *GPS*, компас, гіроскоп та акселерометр для надання даних на основі місцезнаходження користувача. Потім ці

дані визначають, який вміст *AR* буде знайдено чи отримано в певній області. За наявності смартфонів цей тип *AR* зазвичай видає карти та вказівки, інформацію про компанії поблизу. Програми включають події та інформацію, спливаючі вікна бізнес-оголошень, навігаційну підтримку. На рисунку 1.3 можна побачити приклад безмаркерного *AR*, де накладаються вивіски з позначенням будівель, їхній рейтинг та відстань до них від місцезнаходження користувача.



Рис. 1.3. Приклад безмаркерного *AR*

*AR* на основі проєкції. Проектування синтетичного світла на фізичні поверхні, а в деяких випадках з можливістю взаємодії з ним. Це голограми, які широко використовувались/використовуються в науково-фантастичних фільмах, наприклад, Зоряні війни. Він виявляє взаємодію користувача з проєкцією за її змінами. На рисунку 1.4 зображено приклад *AR* із проектуванням синтетичного світла на кросівок – фізичний об'єкт.



Рис. 1.4. Приклад *AR* на основі проєкції

*AR* на базі накладання. У цьому випадку оригінальний вигляд замінюється на доповнений, повністю або частково. Розпізнавання об'єктів відіграє ключову роль, без нього вся концепція просто неможлива. Широко відомими прикладами



накладеної доповненої реальності є додаток *IKEA Catalog*, який дозволяє користувачам розміщувати віртуальні предмети свого каталогу меблів у своїх кімнатах, приклад роботи додатку можна побачити на рис. 1.5.

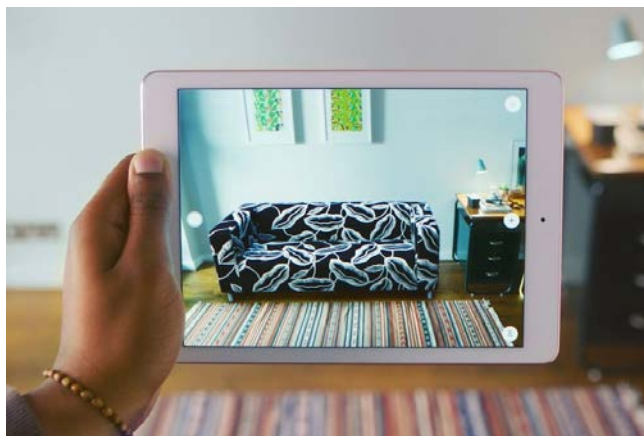


Рис. 1.5. Приклад *AR* на базі накладання

### 1.1.3. Пристрої для відтворення доповненої реальності

Багато сучасних пристроїв вже підтримують доповнену реальність. Починаючи від смартфонів і планшетів, закінчуючи такими гаджетами, як *Google Glass* або портативні пристрої, ці технології продовжують розвиватися. Для обробки та проєкції *AR* пристрої та апаратні засоби, насамперед, мають такі вимоги, як датчики, камери, акселерометр, гіроскоп, цифровий компас, *GPS*, процесор, дисплеї.

Пристрої, для відтворення доповненої реальності, можна віднести до наступних категорій:

– Мобільні пристрої (смартфони та планшети) - найбільш доступні та найкращі для мобільних додатків із *AR*, починаючи від просто ігор та розваг, закінчуючи бізнес-аналітикою, спортом та соціальними мережами.

– Спеціальні *AR*-пристрої, розроблені головним чином і виключно для досвіду доповненої реальності. Одним із прикладів є хед-ап дисплеї (*HUD*), які надсилають дані на прозорий дисплей безпосередньо в поле зору користувача. Спочатку були представлені для підготовки пілотів військових винищувачів, тепер такі пристрої застосовуються в авіації, автомобільній промисловості, виробництві, спорті тощо.

– AR-окуляри (або розумні окуляри) - *Google Glasses, Meta 2 Glasses, Laster See-Thru, Laforge AR* окуляри та ін. Ці пристрої здатні відображати сповіщення зі смартфона, допомагати працівникам конвеєра тощо.

– AR-лінзи (або розумні лінзи) просовують доповнену реальність ще на крок. Такі виробники, як *Samsung* і *Sony*, оголосили про розробку AR-лінз. Відповідно, *Samsung* працює над об'єктивами як аксесуаром до смартфонів, тоді як *Sony* розробляє об'єктиви як окремі AR-пристрої (з такими функціями, як фотографування чи зберігання даних).

– Віртуальні дисплеї сітківки (*VRD*), що створюють зображення, проектуючи лазерне світло в людське око. Орієнтуючись на яскраві зображення з високою контрастністю та високою роздільною здатністю, такі системи ще не створені для практичного використання.

#### 1.1.4. Виклики індустрії доповненої реальності

Відсутність перевірених бізнес-моделей. Одне з найдивніших речей технології доповненої реальності полягає в тому, що, незважаючи на широке прийняття та масове прийняття громадськістю, вона ще не робить все так добре в бізнесі. Однак є промінь світла у вигляді промислових компаній. Вони мають достатньо фінансових ресурсів для стимулювання розвитку, і існує безліч сфер, де технологія AR може стати в нагоді. Такі додатки можуть створити технологію доповненої реальності не як іграшку, а як життєздатний допоміжний інструмент, що покращує загальну якість кінцевого продукту.

Відсутність стандартів дизайну та розробки додатків з доповненою реальністю. На даний момент це те, що будується для доповненої реальності.

Проблеми безпеки та конфіденційності з доповненою реальністю. Конфіденційність та безпека також ставлять значні виклики, що ставляться до галузі AR. Через невідповідність програмування доповненої реальності, нагляд та недбалість є законний шанс потрапити в халепу, не маючи сенсу це робити. Найбільша проблема полягає в тому, що жодне фактичне регулювання не визначає, що дозволено, а що не

дозволено в середовищі доповненої реальності. А це означає, що технологію можна використовувати зі зловмисними намірами так само, як і для розваг.

Можливість фізичної шкоди. Існує значна можливість завдати шкоди собі та оточенню через характер програми та відсутність уваги. Справа в тому, що доповнена реальність працює в реальному світі і додає в неї трохи цифрового. Ці елементи відволікають увагу від реальності, що може спричинити потенційно небезпечну ситуацію. Доповнена реальність, хоч і є вартим доповненням до реальності, але також і може слугувати значним фактором, що відволікає. Ось чому виробники автомобілів настільки неохоче впроваджують *AR*-дисплеї – це більше відповідальність, ніж фактична перевага для керування процесом. Таким чином, існує потреба у розробці певного виду *AR*-інтерфейсу, який буде корисним, але при цьому не відволікатиме від процесів навколо.

Низька якість контенту та прикладів використання. Поки технологія *AR* швидко розвивається і поступово розширює сферу застосування, проблема з випадками її використання та вмістом залишається. Частково причиною цієї проблеми є відсутність досвіду у цій галузі. Не вистачає розробників, які чітко розуміються на технологіях і можуть забезпечити доступний та корисний досвід. Розширена реальність є додатковим елементом до інших видів діяльності, за задумом – вона повинна суттєво сприяти процесу, надаючи додаткову інформацію зручніше або надаючи допомогу у виконанні конкретних дій.

Інша частина проблеми полягає в тому, що громадськість здебільшого не усвідомлює переваг доповненої реальності в різних сферах. Значна частина потенційних користувачів все ще сприймає його як хитру наукову фантастику.

Ще однією великою проблемою впровадження *AR*-рішень є технологічний розрив між *AR*-пристроями. Беручи до уваги, що більшість цільової аудиторії, швидше за все, не придбає *AR*-обладнання через його непрактично та високі ціни смартфони залишаються переважною функцією, і оскільки вони мають певні обмеження в дизайні додатків з доповненою реальністю – це значно обмежує і всю суть впровадження *AR*-рішення [6].

Хоча перелік вищезазначених проблем, пов'язаних з *AR*, може здатися залякуючим і відволікаючим – для технології доповненої реальності немає про що турбуватися. *AR* довів свої можливості розширення користувацького досвіду та покращення зручності. Вирішення цих викликів – це лише питання часу.

#### 1.1.5. Сфери застосування *AR*

Доповнена реальність покращує сприйняття користувачем взаємодії з реальним світом. Віртуальні об'єкти відображають інформацію, яку користувач не може безпосередньо виявити власними почуттями. Інформація, передана віртуальними об'єктами допомагає користувачеві виконувати реальні завдання. *AR* - конкретний приклад того, що Фред Брукс називає Інтелектуальне посилення (*Intelligence Amplification*): використання комп'ютера як інструменту для полегшення виконання завдання людиною [2].

У повсякденному житті існує безліч застосувань технології *AR*, наприклад:

- Освіта: інтерактивні моделі для навчальних цілей, від математики до хімії.
- Медицина / охорона здоров'я: допомагає діагностувати, контролювати, навчати, локалізувати тощо. Нейрохірурги іноді використовують *AR*-проекцію тривимірного мозку, щоб допомогти їм в операціях.
- Військові: для вдосконаленої навігації, розмітки об'єктів у режимі реального часу. Пілоти військових винищувачів бачать *AR*-проекцію своєї висоти, швидкості та інших даних на своєму козирку шолома, а це означає, що їм не потрібно витрачати фокус, поглядаючи вниз, щоб їх побачити.
- Мистецтво / інсталяції / візуальне мистецтво / музика. На таких історичних місцях, як Помпеї в Італії, *AR* може проектувати вигляд давніх цивілізацій на сучасні руїни, оживляючи минуле.
- Туризм: дані про пункти призначення, визначні пам'ятки, навігацію та напрямки. Покращені навігаційні системи використовують доповнену реальність для накладання маршруту прямо на вигляд дороги.

– Трансляція: покращення прямих подій та трансляції подій шляхом накладання вмісту. Під час футбольних ігор коментатори використовують *AR*, щоб намалювати лінії на полі для ілюстрації та аналізу гри.

– Промисловий дизайн: для візуалізації, обчислення або моделювання.

– Реклама. Гігант меблів та предметів домашнього вжитку *IKEA* пропонує програму *AR*, яка дозволяє побачити, як якийсь предмет меблів буде виглядати та вміщуватися у вашому просторі. Також уже існує безліч програм від різних світових брендів для примірки одягу, окулярів, машини.

## **1.2. Аналіз процесу розробки програмного забезпечення**

Програмне забезпечення – це інструкції, що вказують комп'ютеру, що потрібно робити. Програмне забезпечення включає в себе весь набір програм, та процедур, пов'язаних з роботою комп'ютерної системи. Цей термін був введений для того, щоб відрізнити ці інструкції від апаратних засобів, що в свою чергу є фізичними компонентами комп'ютерної системи. Набір команд, що спрямовують апаратне забезпечення комп'ютера на виконання завдання називаються програмою або програмним забезпеченням.

Існує три основних типи:

– Системне програмне забезпечення для забезпечення основних функцій, таких як операційні системи, управління дисками, утиліти, управління обладнанням та інші операційні потреби.

– Програмне забезпечення для надання програмістам таких інструментів, як текстові редактори, компілятори, лінкери, налагоджувачі та інші інструменти для створення коду.

– Прикладне програмне забезпечення (програми або програми), що допомагає користувачам виконувати завдання. Наприклад, пакети офісної продуктивності, програмне забезпечення для управління даними, медіаплеєри та програми безпеки. Поняття додатки також стосується веб- і мобільних додатків, таких як ті, що

використовуються для покупок на *rozetka.com*, спілкування з *Facebook* або розміщення фотографій в *Instagram* [7].

Можливим четвертим типом є вбудоване програмне забезпечення. Вбудоване програмне забезпечення системи використовується для управління машинами та пристроями, які зазвичай не вважаються комп'ютерами - телекомунікаційними мережами, автомобілями, промисловими роботами тощо. Ці пристрої та їх програмне забезпечення можна підключити як частину Інтернету речей (*IoT*) [8].

Зазвичай програмне забезпечення зберігається на зовнішньому пристрої довготривалої пам'яті, наприклад, на жорсткому диску. Коли програма використовується, комп'ютер зчитує її з накопичувача і тимчасово поміщає інструкції в оперативну пам'ять (ОЗУ). Процес зберігання, а потім виконання інструкцій називається «запуском» або «виконанням» програми. Програмні програми та процедури, які постійно зберігаються в пам'яті комп'ютера за допомогою технології лише для читання (*ROM*), називаються мікропрограмою або «жорстким програмним забезпеченням».

Розробкою програмного забезпечення в основному займаються програмісти, інженери та розробники програмного забезпечення. Ці ролі взаємодіють і перекриваються. Робота з розробки програмного забезпечення не обмежується програмістами або командами розробників. Професіонали, такі як науковці, виробники пристроїв та виробники обладнання, також створюють програмний код, хоча вони і не є в основному розробниками програмного забезпечення. Також це не обмежується традиційними галузями інформаційних технологій, такими як програмне забезпечення або напівпровідниковий бізнес. Насправді, за даними Інституту Брукінгса, на цей бізнес «припадає менше половини компаній, що займаються розробкою програмного забезпечення» [9].

### 1.2.1. Етапи процесу розробки програмного забезпечення

Розробка програмного забезпечення, як правило, передбачає такі кроки:

– Вибір методології для встановлення основи, в якій застосовуються етапи розробки програмного забезпечення. Вона описує загальний робочий процес або дорожню карту для проекту. До методологій можна віднести *Agile development*, *DevOps*, *Rapid Application Development (RAD)*, *Scaled Agile Framework (SAFe)*, *Waterfall* та інші.

– Збір вимог для розуміння та документування того, що вимагають користувачі та інші зацікавлені сторони.

– Вибір або побудова архітектури як основної структури, в якій буде працювати програмне забезпечення.

– Розробка дизайну навколо рішень проблем, що пред'являються вимогами, часто залучаючи моделі процесів та розкадрування.

– Побудова коду відповідною мовою програмування. Залучає експертну оцінку та експертну групу для попереднього усунення проблем та швидшого створення якісного програмного забезпечення.

– Тестування за заздалегідь запланованими сценаріями в рамках проектування та кодування програмного забезпечення, і проведення тестування продуктивності для імітації тестування навантаження на додаток.

– Управління конфігурацією та дефектами, щоб зрозуміти всі артефакти програмного забезпечення (вимоги, дизайн, код, тест) та побудувати різні версії програмного забезпечення. Встановлення пріоритетів забезпечення якості та критеріїв випуску для усунення та відстеження дефектів.

– Розгортання програмного забезпечення для використання, реагування та вирішення проблем користувачів.

– Перенесення даних на нове або оновлене програмне забезпечення із існуючих програм або джерел даних, якщо це необхідно.

– Управління та вимірювання проекту для підтримки якості та доставки протягом життєвого циклу програми, а також для оцінки процесу розробки.

Етапи процесу розробки програмного забезпечення вписуються в життєвий цикл розробки програм, що містить в собі наступні кроки:

– аналіз та специфікація вимог;

- проектування та розробка;
- тестування;
- розгортання;
- технічне обслуговування та підтримка.

Етапи процесу розробки програмного забезпечення можна згрупувати за фазами життєвого циклу, але важливість життєвого циклу полягає в тому, що воно переробляється для постійного вдосконалення. Наприклад, проблеми користувачів, які з'являються на етапі технічного обслуговування та підтримки, можуть стати вимогами на початку наступного циклу.

Розробка програмного забезпечення є важливою, оскільки допомагає бізнесу диференціюватися та бути більш конкурентоспроможним. Це може покращити взаємодію з клієнтами, швидше вивести на ринок більше інноваційних, багатофункціональних продуктів та зробити операції більш ефективними, безпечними та продуктивними [9].

### **1.3. Аналіз середовищ для розробки програм із доповненою реальністю**

#### **1.3.1. Аналіз ігрового рушія для розробки програм Unity**

Платформа 3D-розробки в реальному часі *Unity* дозволяє художникам, дизайнерам та розробникам спільно створювати дивовижні захоплюючі та інтерактивні враження. (Доступно для *Windows*, *Mac* та *Linux*.) Гнучка платформа розробки *Unity* у режимі реального часу пропонує неймовірні можливості для всіх галузей та додатків.

*Unity* – це провідна платформа у світі для створення та експлуатації інтерактивного 3D-вмісту в режимі реального часу, що забезпечує інструменти для створення дивовижних ігор та їх публікації на широкому діапазоні пристроїв. Основна платформа *Unity* дозволяє цілим творчим колективам бути більш продуктивними разом.



Основною перевагою *Unity* є кросплатформність, що дозволяє розроблювати додаток одночасно на кілька платформ (*Android* і *iOS*) та надає можливість охопити величезну кількість гравців/користувачів, де б вони не були, побудувавши свою гру на всіх основних мобільних, консольних, ПК, *AR*, *VR* та веб-платформах.

Документація для даного ігрового рушія є вичерпною, що допомагає швидко створити свою першу програму. Ком'юніті користувачів *Unity* величезна і є багато різних формумів, уроків, туторіалів, де можна почерпнути знання. За даними *LinkedIn*, *Unity Developer* посідає 7 місце серед 10 найбільш швидкозростаючих робочих місць у США. Завдяки централізованому пулу талантів на *Unity Connect* та програмі сертифікації для перевірки навичок відповідно до рівня здібностей, легко знайти кваліфікованих талантів *Unity* для побудови команди.

Невимовним плюсом даного ПЗ є те, що його можна використовувати безкоштовно нелімітовану кількість часу. Також пропонується варіація різних планів, якщо займатись розробкою ПЗ на цьому рушії в комерційних цілях.

Ще є хороше доповнення до програми *Unity* – *Unity Hub*, де можна зберігати всі свої проекти, навіть кільком учасникам команди, що дозволяє легку передачу проектів між колегами.

Платформа *Unity* є надзвичайно розширюваною і готова для адаптації до сучасних та нових потреб завдяки потужній системі сценаріїв *C #*, всебічному *API* та великій документації. Доступ до вихідного коду можна придбати для розробки *C ++*.

Завдяки інтуїтивно зрозумілому інтерфейсу та інструментам *Unity* дозволяє легко запуснути робочий прототип і обійти витрати часу на програмування на низькому рівні, інтерфейс проекту під час розробки можна побачити на рисунку 1.7. Зміни прості та миттєві завдяки магії реального часу, що полегшує їх створення та повторення.

*Unity* включає надійний і добре задокументований *API* з доступом до повного спектру систем *Unity*, включаючи фізику, візуалізацію та комунікації, що забезпечує розширену модель взаємодії та інтеграцію з іншими системами.

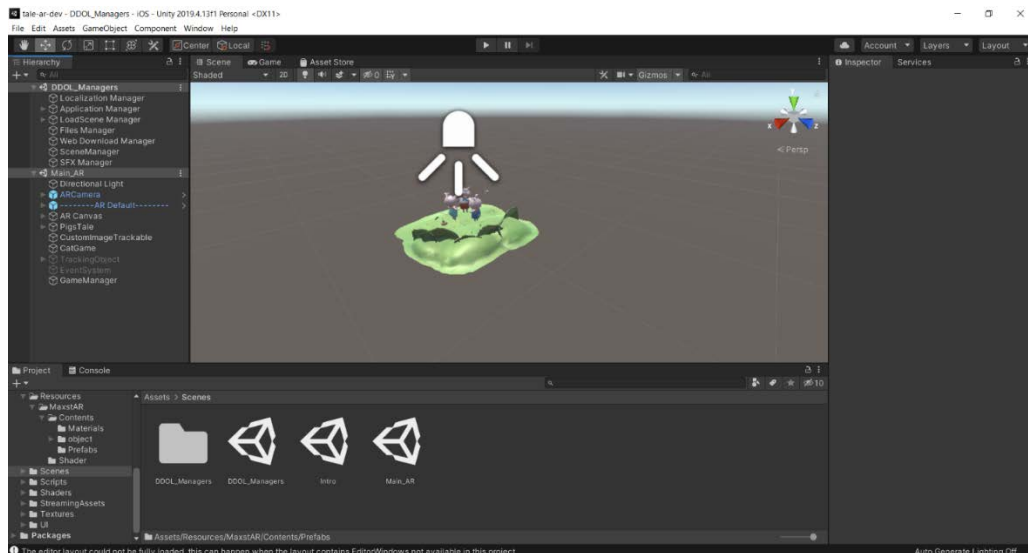


Рис. 1.6. Інтерфейс програми *Unity*

*Unity's Scriptable Render Pipeline* забезпечує повну гнучкість для оптимізації графіки. Високоякісний конвеєрний рендер пропонує візуальну якість світового класу на високопродуктивному обладнанні, тоді як *Universal Render Pipeline* підтримує чутливі характеристики при масштабуванні для мобільних пристроїв.

*Unity* має величезний магазин асетів, що надає доступ до десятків тисяч готових рішень та інструментів для підвищення продуктивності для швидкого запуску проекту. Ці будівельні блоки скорочують час на розробку, щоб швидше розпочати роботу та пройти через фінішну пряму.

*Unity* пропонує перше в історії рішення про монетизацію ігор, яке оптимізоване для значення життєвого циклу. Можна заробляти найбільший дохід, забезпечуючи чудові враження користувачам [12].

### 1.3.2. Аналіз ігрового рушія для розробки програм Unreal Engine

*Unreal Engine* – це відкритий інструмент створення 3D у реальному часі. Безперервно розвиваючись, слугуючи не лише своєму початковому призначенню як ультрасучасного ігрового двигуна, сьогодні він надає творцям різних галузей свободу та контроль, щоб забезпечити найсучасніший вміст, інтерактивний досвід та захоплюючий віртуальний світ.

*Unreal Engine* – це повний набір засобів розробки для тих, хто працює з технологіями реального часу. Від візуалізації дизайну та кінематографічного досвіду до високоякісних ігор на ПК, консолі, мобільному пристрої, *VR* та *AR* - *Unreal Engine* надає все необхідне для запуску, доставки, зростання та виділення серед натовпу.

Підключення до конвеєрів для виробництва медіа, підтримує такі галузеві стандарти, як *FBX*, *USD* та *Alembic*. Першокласна підтримка *USD* дозволяє користувачам краще співпрацювати з членами команди та працювати паралельно. *Unreal Engine* може читати файл *USD* з будь-якого місця на диску без трудомісткого повного імпорту та записувати зміни до нього як заміни. В *Unreal Engine* можна легко автоматизувати робочі процеси підготовки даних, навіть якщо ви не програміст, за допомогою спрощеного візуального інструменту, що дозволяє створити фільтри та оператори, які можна зберігати та використовувати повторно для інших сцен чи проектів.

*Unreal Engine* включає в себе *Unreal Editor* – інтегроване середовище розробки, доступне для *Linux*, *MacOS* та *Windows* для створення вмісту. Завдяки підтримці багатокористувацького редагування, художники, дизайнери та розробники можуть одночасно вносити зміни в один і той же проект *Unreal Engine* безпечним та надійним способом, тоді як можливість запуску повного *Unreal Editor* у режимі *VR* означає, що ви можете створити *WYSIWYG* (*what you see is what you get* ) середовище.

*Unreal Engine Marketplace* має тисячі високоякісних об'єктів та плагінів для прискорення виробництва та привнесення нових функцій у роботу. Можна легко отримати доступ до нових середовищ, персонажів, анімації, текстур, реквізиту, звукових та візуальних ефектів, музичних композицій, креслень, плагінів для інтеграції проміжного програмного забезпечення, додаткових інструментів та повних початкових комплектів. Вміст на мільйони доларів знаходиться у вільному доступі на *Marketplace*, ще більше вмісту можна придбати. Також в *Unreal Engine* існує кілька зразків проектів, що теж допоможуть швидше розробити свій перший проект та навчитися працювати в даному середовищі.

Можна отримати безкоштовний доступ до повного вихідного коду C ++, і завдяки цьому можна вивчити, налаштувати, розширити та налагодити весь *Unreal Engine* і завершити свій проект без перешкод.

Безпосередньо в *Unreal Engine* можна об'єднати та використовувати величезні набори даних, захоплені з реального світу, з можливістю імпортувати, візуалізувати, редагувати та взаємодіяти з хмарами точок, отриманими від пристроїв лазерного сканування. Хмари точок можна використовувати для візуалізації місцеположень та для надання точного контексту нещодавно розробленим елементам.

*Unreal Engine* пропонує такі інструменти для підготовки та оптимізації складної моделі для кращої роботи в режимі реального часу, як автоматичне створення рівня деталізації; обшивка і поразка, що усуває приховані поверхні та непотрібні деталі; та інструмент проксі-геометрії, який поєднує кілька сіток та їх матеріалів в єдину сітку та матеріал [13].

#### **1.4. Аналіз наборів із засобів розробки, утиліт і документації для використання технології доповненої реальності у проектах**

*SDK* (від англ. *Software Development Kit*) – набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи. Роль *AR SDK* полягає у виконанні нетривіального завдання злиття цифрового вмісту та інформації з реальним світом. Можливості *SDK* в кінцевому підсумку підкріплять функції та функціональність програми *AR*, тому важливо вибрати правильну платформу, виходячи з вимог проекту.

##### **1.4.1. Набір засобів розробки доповненої реальності *ARKit* від *Apple***

У 2017 році *Apple* випустила *iOS 11*, а подальший запуск *ARKit* став очевидно найвибуховішою подією в історії технологій доповненої реальності. *ARKit* – це унікальний фреймворк, який дозволяє брендам та розробникам створювати

неперевершені враження для сумісних пристроїв *iPhone* та *iPad*. *ARKit SDK* функціонує так само, як і більшість функцій *AR SDK*, дозволяючи поєднувати цифрову інформацію та тривимірні об'єкти з реальним світом, але пропонує в значній мірі неперевершену доступність з точки зору кількості існуючих пристроїв, які він підтримує.

*ARKit* можна запускати на будь-якому пристрої, оснащеному процесором *Apple A9, A10* або *A11* і використовує *VIO* (візуальна інерційна одометрія) для відстеження навколишнього середовища з безперервним рівнем точності. *VIO* дозволяє *ARKit* поєднувати дані *Core Motion* із датчиками камери та надає можливість розробляти програми, які можуть виявляти горизонтальні площини та вертикальні площини. Це дозволяє *ARKit* точно розуміти динаміку та склад певної сцени, а також надає можливість розміщувати тривимірні об'єкти та накладати цифрову інформацію в контекстно відповідний спосіб (наприклад, оскільки *ARKit* розуміє різницю між підлогою та столом, він знає, що поставити пляшку вина на стіл, а не на підлогу).

Розробники та компанії можуть створювати додатки за допомогою *ARKit* та пов'язаних з ними оптимізацій за допомогою сторонніх 3D-механізмів, таких як *Unity*, *Unreal Engine* та *SceneKit*.

*ARKit* надає такі функціональні можливості:

- відстеження *SLAM* (одночасна локалізація та відображення) та злиття датчиків;
- оцінка навколишнього освітлення;
- оцінки шкал;
- оцінка вертикальної та горизонтальної площини з основними межами;
- стабільне та швидке відстеження руху [14, 20].

1.4.2. Набір утиліт для використання доповненої реальності *AR Core* від *Google*

*ARCore* – це власний *SDK* від *Google* для доповненої реальності. Подібно до *ARKit*, дозволяє брендам та розробникам запускати та випускати додатки *AR* на

сумісних смартфонах та планшетах *Google*. Однією з найбільш помітних особливостей *ARCore* є те, що він також підтримує пристрої з підтримкою *iOS* і надає розробникам безпрецедентний доступ до користувачів на обох платформах. *ARCore* має три суттєві особливості, які дозволяють розробникам об'єднати реальний світ з віртуальним:

- оцінка освітленості: оцінює реальні умови освітлення;
- розуміння довкілля: виявляє розмір та розташування вертикальних, горизонтальних та кутових поверхонь;
- відстеження руху: розуміє положення телефону щодо оточення.

Вся пропозиція *ARCore* в основному побудована навколо двох ключових елементів: відстеження в реальному часі та обчислення місцезнаходження пристрою в поєднанні з інтеграцією віртуальних об'єктів із реальним середовищем. Це дозволяє компаніям та брендам розробляти багатий та захоплюючий досвід *AR* для мобільних пристроїв, дозволяючи *3D*-об'єктам, тексту та цифровій інформації безпосередньо потрапляти в навколишнє реальне середовище. *ARCore* безкоштовне для розробників та підтримує цілий ряд смартфонів та планшетів із підтримкою *Android* (і *iOS*), включаючи *Samsung Galaxy* та *Google Pixel*, а також багато інших [15, 20].

#### 1.4.3. Набір засобів розробки доповненої реальності *Easy AR*

*EasyAR SDK* доступний для бізнесу та розробників у дворівневих цінових пакетах: *EasyAR SDK Basic* та *EasyAR SDK Pro*. Базовий пакет обіцяє розробникам розширені *API*, зручний робочий процес та підвищену сумісність. Пакет *Pro* є абсолютно новим і оснащений ексклюзивними функціями, які відсутні в базовій упаковці. Базовий пакет безкоштовний для розробників, які хочуть розробляти додатки з доповненою реальністю, і підтримує *API Java* для *Android*, *API Swift* для *iOS*, а також підтримує ОС *Windows*. Деякі додаткові функції, що підтримуються базовим пакетом, включають відтворення відео, відтворення відео з прозорістю, сканування *QR*-коду та комплексну інтеграцію *Unity*.

Пакет *EasyAR Pro* доступний з усіма функціями, пов'язаними з безкоштовним пакетом базової версії платформи, а також підтримкою *SLAM*, *3D*-відстеженням об'єктів, записом екрану та одночасним виявленням і відстеженням для різних типів цілей.

Основна функція пакету *EasyAR Pro* зосереджена на наступному: *SLAM* (включаючи відстеження поз камери шести ступенів свободи (*DOF*) в режимі реального часу *Monocular* та повну мобільну сумісність), відстеження *3D*-об'єктів (оснащене можливістю розпізнавання та відстеження загального *3D*-об'єкта в повному обсязі) із текстурами в реальному часі), запис на екран (забезпечує простий та ефективний спосіб запису вмісту *AR*), планове відстеження зображень (можливість відстеження та ідентифікації площинних зображень у реальному часі), стислий *API*, який інтегрується з усіма основними мобільними *AR* платформами, а також підтримка взаємодії з метою відображення найбільш переконливого *AR*-вмісту з додатковою функціональністю. Веб-сайт *EasyAR* містить безліч корисної інформації, щоб забезпечити розробку додатка *AR* у найкоротші терміни, з вичерпною інформацією, підтримкою, відповідями на запитання та спільнотою [16, 20].

#### 1.4.4. Набір засобів розробки доповненої реальності *Vuforia*

*Vuforia* – це пакет *AR SDK*, який дозволяє компаніям та розробникам додатків швидко розподіляти високоякісні, орієнтовані на мобільні пристрої *AR*-можливості. *Vuforia SDK* використовує технологію комп'ютерного зору для виявлення та відстеження цілей зображення та *3D*-об'єктів у режимі реального часу. Ця функціональність дозволяє компаніям та агенціям з розробки *AR* орієнтувати та розміщувати віртуальні об'єкти, включаючи *3D*-моделі та інший вміст, стосовно реального середовища. Потім *3D*-моделі та цифрова інформація можуть бути накладені на реальну сцену та переглянуті щодо навколишнього середовища за допомогою смартфона або планшета з підтримкою *AR*.

*SDK* доповненої реальності від *Vuforia* здатний підтримувати широкий спектр *3D* і *2D* цілей, включаючи *3D* мультитаргеторі налаштування, безмаркерні

налаштування та фідуціальні маркери, що називаються «*VuMark*». Деякі з додаткових функцій даного *SDK* включають в себе локалізоване виявлення оклюзії за допомогою віртуальних кнопок, здатність розробляти та калібрувати цільові набори під час виконання та вибір цільового зображення під час виконання.

*Vuforia* надає *API* на *Java*, *C ++* та *Objective C ++* та *.NET* через розширення ігрового механізму *Unity*. З огляду на це, *Vuforia SDK* здатний підтримувати як власну розробку для *iOS* та *Android*, так і розробку *AR*-додатків та прототипів у *Unity*, які можна легко перенести на обидві платформи. Це чудовий варіант для підприємств та брендів, які прагнуть розробляти програми, що охоплюють як *iOS*, так і *Android*, мінімізуючи комерційні та технічні ризики. Це означає, що додатки *AR* можна розробити безперешкодно для якнайширшої кількості цільових мобільних пристроїв в найкоротші терміни [17, 20].

#### 1.4.5. Набір засобів розробки доповненої реальності *MAXST AR SDK*

*MAXST AR SDK* – це крос-платформний механізм *AR*, який надає всі функції та середовища, необхідні для розробки програми доповненої реальності. *SDK* для доповненої реальності *MAXST* забезпечує всеосяжний крос-платформний механізм *AR*, оснащений усіма функціями, необхідними брендам і розробникам для створення досвіду *AR* та додатків. Платформа *MAXST* обіцяє конкурентні ціни в поєднанні зі швидкістю та простотою розробки додатків *AR*.

*SDK MAXST AR* забезпечує наступну функціональність:

- миттєве відстеження (забезпечує можливість ідентифікації горизонтальних / вертикальних площин для накладання відповідного вмісту),
- візуальне *SLAM* (використовує камеру смартфона для створення віртуальної карти навколишньої території),
- об'єкт відстеження (можливість імпорту файлів карт, створених за допомогою візуального *SLAM*),
- відстеження зображень (накладання *3D*-вмісту, відео та зображень),



– відстеження маркерів (накладання вмісту поверх маркерів із наданими 8192 маркерами),

– функція сканування *QR* / штрих-коду.

*SDK MaxST AR* також пропонує цілий ряд корисних функцій, таких як можливості крос-платформної розробки, що працюють на всіх основних платформах, включаючи *Mac OS, iOS, Android, Windows* та *Unity 3D*. Платформа також сумісна з широким асортиментом продуктів *HUD* та розумних окулярів [18, 20].

#### 1.4.6. Набір засобів розробки доповненої реальності *Wikitude*

*Wikitude* – це *SDK*, спеціально розроблений для розробки мобільних додатків *AR* та прототипів. Компанія була заснована ще в 2008 році в Зальцбурзі, Австрія. Коли *Wikitude SDK* був первинно запущений, платформа була розроблена з основною метою: дозволити розробникам *AR* створювати орієнтований на розташування досвід доповненої реальності за допомогою програми *Wikitude World Browser*. Швидко перейти до 2012 року, і *Wikitude* змінив свою основну технологічну пропозицію, запустивши *Wikitude SDK* з функціями геолокації, відстеження та розпізнавання зображень, запечених безпосередньо на базовій платформі.

Зараз *Wikitude SDK* – це основний продукт компанії, який обіцяє розробникам можливість створювати захоплюючі *AR*-можливості в найкоротші терміни. Тепер *Wikitude SDK* також включає такі функції, як візуалізація *3D*-моделі, *AR* на основі розташування та накладання відео. Нарешті компанія впровадила технологію *SLAM* (одночасна локалізація та картографування), яка полегшує безперешкодне відстеження та розпізнавання об'єктів поряд із моментальним відстеженням без маркерів.

*Wikitude SDK* працює на багатьох платформах і в даний час доступний для підтримки ОС *Windows, iOS, Android* та ряду *HUD* (головних дисплеїв) [19-20].

## Висновки за розділом

Технологія доповненої реальності досить нова, проте вона дуже швидко розвивається та популяризується, виглядає дуже цікаво, красиво, часом неочікувано і обов'язково сприяє запам'ятовуванню такого пережитого досвіду. Все більше бізнесу прагне використовувати дану технологію в своїх цілях, адже в даний час, технологія справляє вау-ефект на своїх користувачів, що дуже цінується в маркетингу, і сприяє відкладенню в пам'яті отриманих вражень та інформації, що передавалась за допомогою інструменту з використанням *AR*. Можливих шляхів застосування *AR* є дуже багато, але потрібно завжди спочатку подумати наскільки вона є доречною в кожному окремому випадку. Також дуже важливо звертати увагу на умови, в яких будуть використовуватись продукти з використанням технології доповненої реальності. На даний момент уже пророблено чимало роботи по створенню всіх необхідних пристроїв для ефективного використання цієї технології, проте прогрес не стоїть на місці і всі технології стають нано, тому скоро можна буде носити лінзи на очах, з втіленою технологією доповненої реальності. Взагалі ця технологія призначена для доповнення реальності, тобто вона повинна допомагати у повсякденному житті.

Розробка програмного забезпечення досить трудомісткий процес, який вимагає залучення багатьох спеціалістів та чимало часу на процес. В даний час існує вже дуже багато середовищ розробки програм для різних потреб. Обирати їх потрібно відповідно до кінцевої цілі проекту та навичок розробника. Обидва, розглянуті в розділі 2 рушії є потужним інструментом для створення програмного забезпечення різного роду. Але, на мою думку, *Unreal Engine* створений для більш складних проектів. Для розробки додатку, що створюється під час виконання даної дипломної роботи було обрано середовище розробки *Unity* через його зручність і тому, що я з ним уже знайома.

Я вважаю, що ми живемо в чудовий час, коли вже багато чого створено, залишається пожинати плоди і творити із набору даних, що уже відомі. Саме для цього готуються розробниками цільової технології набори із засобів розробки, утиліт

і документації. Для використання у своїх проектах технології *AR* зараз уже існує багато сервісів, що пропонують свої послуги за оплату своєї праці. Чудовою новиною є те, що такими сервісами можна також користуватись і безкоштовно з різними своїми правилами. Після детального аналізу найвідоміших та найпопулярніших *AR SDK* зроблено висновок, що для даної дипломної роботи буде найкраще використовувати безкоштовний план *SDK* від *MAXST*. Як і з вибором середовища розробки вибір *sdk* було зроблено в сторону відомого мені раніше сервісу, з яким я постійно співпрацюю в реальному житті. Сервіс зручний, легкий у імпорті в проект, із зрозумілою повною документацією для роботи з ним.

## РОЗДІЛ 2

### РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ «AR ДОВІДНИК»

#### 2.1. Створення ідеї та встановлення основного посилу проекту

Першим етапом розробки додатку для досягнення поставленої мети дипломної роботи буде створення ідеї. Ідея прийшла в голову під час придумування теми дипломного проекту разом із науковим керівником. Однозначно було вирішено мною, що у застосунку буде використана механіка доповненої реальності, тому що вона мені дуже імпонує та в своєму житті я стикаюся із нею майже щодня. Далі було стверджено, що даний дипломний проект має вирішити проблему допомоги та зацікавленості у навчанні для студентів спеціальності комп'ютерна інженерія, кафедри комп'ютерних систем та мереж. Це означає, що потрібно в доповненій реальності показати якусь із тем, наприклад для лабораторної роботи або лекції одного із предметів, які ми вивчали під час навчання на даній спеціальності. Мною було обрано предмет мікроелектроніка. До розгляду було взято саме дискретні елементи, тому що всі вони чудово підходять для того, щоб показати їх за допомогою технології доповненої реальності.

Дискретні елементи – це такі елементи (електрорадіоелементи), які мають певне конструктивно закінчене виконання. Дискретні електрорадіоелементи призначені для виконання певних елементарних функцій в пристроях генерування і оброблення сигналів, запису, оброблення, зберігання та відтворення інформації.

Робота дискретних елементів визначається тими функціями, які вони виконують.

В даній роботі розглянемо класифікацію дискретних елементів за їх здатністю перетворювати енергію джерела живлення в енергію корисних сигналів. За цією ознакою всі дискретні елементи поділяють на пасивні та активні. Пасивні електрорадіоелементи неспроможні цього робити, активні спроможні. Всю довідкову інформацію було вирішено брати із перевіреного роками джерела – сайту *Wikipedia*.

Активний елемент або активний компонент електронної техніки (*Active Component*) – це різноманітні електронні прилади, що розрізняються за принципами дії й призначенням. Вони називаються активними тому, що їхнє функціонування пов'язане зі споживанням енергії від зовнішніх джерел живлення. Як правило, у радіоелектронних пристроях - це електрична енергія. Напруга таких джерел може бути постійною або змінною. Типовим активним компонентом є генератор, транзистор або інтегральна схема [22].

Пасивний елемент або пасивний компонент електронної техніки (*Passive Component*) – елемент, що функціонує без зовнішніх джерел живлення. Вхідні сигнали передаються на вихід, відтворюючи закон часової залежності, без підсилення. Властивості цих елементів (у більшості випадків) не залежать від полярності прикладеної напруги або напрямку струму, що протікає.

Номенклатура пасивних елементів досить широка. До них відносять резистори, конденсатори, індуктивні компоненти, елементи комутації та інші елементи. Вони можуть мати сталі й змінні параметри. З цією ознакою пов'язані принципові відмінності в їхніх конструкціях. Елементи зі змінними параметрами, як правило, значно дорожчі, мають більші габарити й масу [23].

На даному етапі була створена структура подачі інформації в додатку. Структура зображена на рис. 2.1. Основою додатка є дискретні елементи, які в свою чергу поділяються на активні та пасивні. Кожна група елементів має свої підгрупи. Було вирішено зобразити такі підгрупи:

- для активних елементів – діоди, підсилювачі, тиристори та транзистори;
- для пасивних елементів – конденсатори, котушки індуктивності, реле та резистори.

Для кожної підгрупи буде відведена своя окрема сторінка, де буде невеличкий опис та кнопка, щоб перейти до камери, щоб подивитись саме на ці об'єкти.

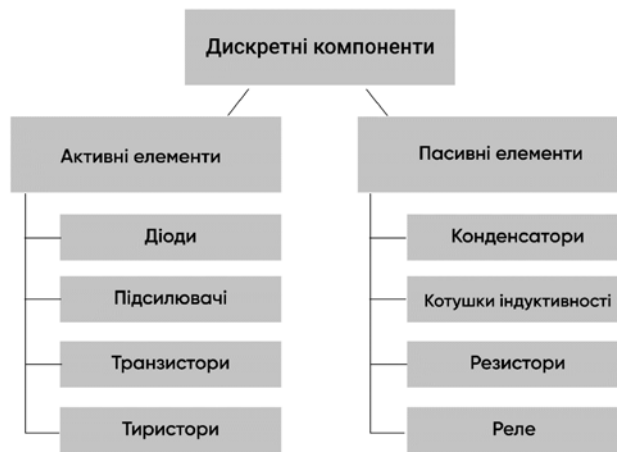


Рис. 2.1. Схема структури додатку

Назву додатку було придумано під час створення дизайну інтерфейсу, але вважаю доречним вказати її в цьому розділі.

Для того, щоб правильно назвати додаток потрібно під час придумування притримуватись таких правил:

- Назва має відображати основні функції додатка. Багато успішних додатків вибирають назву, яка описує те, що робить програма. Іноді користувачі оцінюють ваш додаток одразу на основі першого враження та не читаючи опису додатка - наявність імені, що пояснює особливості програми, може дати головну перевагу в цих випадках. Назва додатку не повинна бути надзвичайно розумною. Іноді «розумний» може бути складним. Просте ім'я легко запам'ятати і споживачам легко зрозуміти, про що йдеться в додатку.

- Назва повинна бути короткою та такою, що легко запам'ятовується. Довге ім'я з кількома словами та безліччю символів - це вірний спосіб досягти того, що користувачі забудуть ім'я додатка одразу ж. В ідеалі вам потрібно коротке ім'я з мінімальною кількістю символів.

- Потрібно зробити назву свого додатка заклик до дії. Хоча цих слів і немає у словнику, всі знають, що це означає, коли хтось каже «загугли». Можна вважати, що успіху досягнуто, коли назва додатку стає, словом, заклик до дії. Одна з основних підказок при вивченні того, як називати програму, - потрібно розглянути, як додаток звучить у повсякденній мові та як він працює.

– Потрібно обирати назву, яка шукається. Іноді ім'я можна вибрати відповідно до того як воно відповідає цілям підприємця щодо маркетингу та залучення клієнтів. Важливо враховувати той факт, що 63% користувачів смартфонів знаходять програми через магазин програм; вони вводять в пошуковик конкретні ключові слова, щоб знайти програму, яка вирішує проблему їх клієнтів. Оскільки така велика частина завантажень є результатом пошуку, розумним рішенням є ім'я, яке відповідає ключовим словам, які шукаються.

– Потрібно обирати очевидне ім'я. Іноді не потрібно вибирати ім'я, оскільки правильне ім'я може вибрати саме проект.

Незважаючи на те, що всі ці поради надзвичайно важливі для того, щоб навчитися називати програму правильно, ось одна порада, яка є найбільш важливою – потрібно переконатися в тому, що програма така ж хороша та якісна, як і сама назва. Немає нічого гіршого, ніж запустити якусь програму, назва якої дуже круто звучить, і лише розчаруватися не настільки дивовижним програмним рішенням. Вибираючи свою чудову назву програми, потрібно впевнитися, що вона справді представляє чудову програму [24].

Для даної роботи було обрано назву «AR Довідник», тому що вона лаконічна, одразу зрозуміла, коротка та одразу пояснює суть програми.

Отже, можна сказати, що основним посилом розроблюваного додатку є показати інформацію про дискретні елементи в цікавій формі.

Для реалізації проекту було обрано середовище розробки *Unity* та *AR SDK MAXST*.

## **2.2. Створення користувацького інтерфейсу програми**

Під час створення інтерфейсу потрібно завжди думати про кінцевого користувача, для якого розроблюється *UI*. Успіх у створенні інтерфейсу залежить від знання кінцевих користувачів, включаючи розуміння їх цілей, навичок, уподобань та тенденцій. Потрібно обов'язково врахувати наступні пункти при розробці інтерфейсу. Цільовою аудиторією розроблюваного додатку є студенти вищих

навчальних закладів, що навчаються на спеціальностях, пов'язаних із комп'ютерною інженерією, приблизно 1-4 курс, тобто це приблизно 17-21 років, переважно чоловіча стать.

Інтерфейс має бути простим, зручним, лаконічним. Потрібно уникати непотрібних елементів. Також важливим є створення та використання узгоджених, сталих елементів інтерфейсу, таких, які користувачам давно відомі та зручні. Якщо використовувати загальні елементи у своєму інтерфейсі, користувачі почуватимуться комфортніше і зможуть швидше виконувати необхідні справи [25].

Потрібно, щоб макет був цілеспрямованим. Під час створення інтерфейсу варто розглянути просторові взаємозв'язки між елементами на сторінці та структурувати сторінку по важливості. Ретельність під час розміщення предметів може допомогти привернути увагу до найважливіших частин інформації та допоможе її сканувати та зчитати.

Також важливо стратегічно використовувати колір і текстуру. Можна спрямувати увагу на або перенаправити увагу з предметів, використовуючи колір, світло, контраст і текстуру на свою користь. Потрібно використовувати можливості для відтворення візуальної ієрархії [26].

Ще основним є використання типографіки для створення ієрархії та чіткості. Необхідно уважно думати, як використовується гарнітура. Різні розміри, шрифти та розташування тексту сприяють збільшенню масштабованості, розбірливості та читабельності [25].

Важливим елементом для програми є її лого. Його можна використовувати як під час створення іконки додатку, так і безпосередньо в макетах самого додатку для більшої впізнаваності.

Для розроблюваної програми було зроблене лого, що можна побачити на рис. 2.2. Даний логотип символізує знання через розпираючий мозок та окуляри, досить примітивний ланцюжок думок, проте саме з таким підходом все більше людей зможуть зрозуміти основний посил.



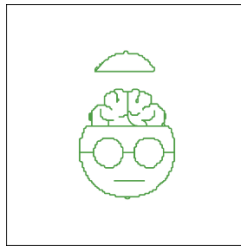


Рис. 2.2. Логотип для додатку AR Довідник

Вслід за логотипом можна зробити іконку для додатка, що буде відображатись на робочому столі девайсу на який буде встановлено додаток. Як це можна побачити на рисунку 2.3, під час створення було використано розроблений логотип та частину назви.

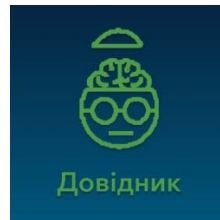


Рис. 2.3. Іконка для додатку AR Довідник

Час на завантаження додатка може варіюватись від моделі телефону та його потужності. Тому напевне не можна одразу сказати точний час очікування і необхідно для цієї дії розробити такий елемент інтерфейсу, щоб користувач розумів, що програма не просто зависла, а в даний час відбувається якась дія, обробка інформації, загрузка. Для такого випадку для цієї програми було розроблено послідовність кадрів, яку можна об'єднати в анімоване зображення саме на позначення, що дія відбувається. Ці кадри для анімації завантаження зображено на рисунку 2.4.

Далі в програмі має відкритись основний екран. Опираючись на структуру подачі інформації в додатку (див. рис. 2.1), було зроблено висновок, що на головному екрані доцільно одразу робити категоризацію контенту по типу активні елементи та пасивні елементи. В увесь дизайн було інтегровано лого додатку, що можна побачити на рисунку 2.5 для кращої впізнаваності. Також з головного екрану можна перейти на сторінку інформації про додаток. Цей екран можна побачити на рисунку 2.6, він містить загальну інформацію про мету цього розробленого додатка та про його можливості.

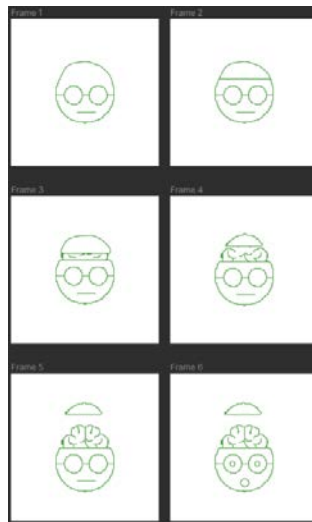


Рис. 2.4. Кадри анімованого зображення-позначення завантаження

По натисканні на кнопку «Активні елементи» на головному екрані (див. рис. 2.5) відкриється наступний екран додатку з категоризацією активних дискретних елементів відповідно до структури подачі інформації (див. рис. 2.1).

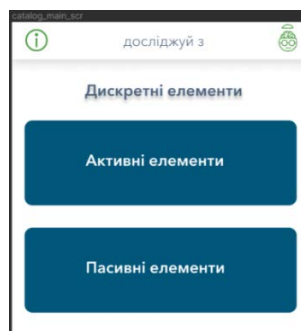


Рис. 2.5. Частина макету головного екрану додатку



Рис. 2.6. Частина макету сторінки інформації про додаток

Контент у додатку поділяється на сторінки типу просто інформація та з можливістю перейти до камери для перегляду контенту в доповненій реальності.

Відмінність між типами контенту буде одразу видно на цьому екрані за допомоги значків на кнопках контенту. Можна звернути на це увагу на рис. 2.7, інформаційні карточки контенту містять на собі іконку із стрілкою, що позначає просто перехід, а от карточки контенту із можливість перейти в AR камеру для відтворення інтерактивного контенту в доповненій реальності позначаються синьою іконкою із надписом AR.

На сторінці категорії активних елементів є такі кнопки (підкатегорії): Дискретні елементи, Активні елементи, Транзистори, Діоди, Тиристори, Підсилювачі. По натисканні на ці кнопки відбуватиметься перехід на відповідні сторінки інформації про даний тип дискретних елементів.

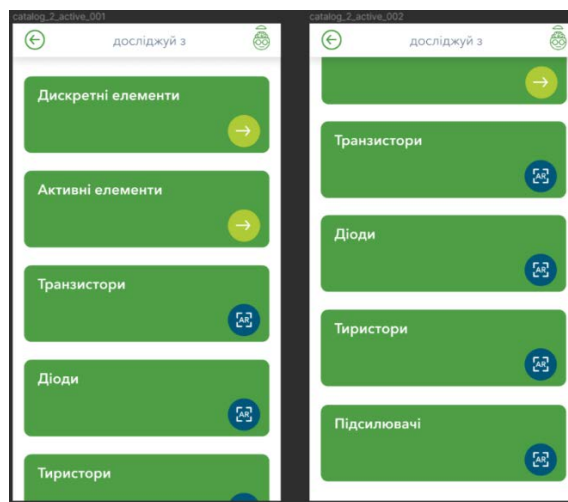


Рис. 2.7. Макет сторінки меню активних елементів

По натисканні на кнопку «Дискретні елементи» відбудеться перехід на інформаційну сторінку про дискретні елементи (див. рис. 4.8). Така ж сама кнопка буде доступною і на екрані меню пасивних елементів, тому що це загальна тема для обох категорій і її доречно продублювати. Дана карточка контенту є просто інформаційною, всі інші такі ж зроблені по такому ж принципу. Інформацію в текстовому вигляді про дискретні елементи було взято із джерела *Wikipedia* для всіх елементів, що розглянуто в програмному забезпеченні AR Довідник. На цьому екрані доступний такий інтерактивний елемент – кнопка «Більше». Якщо натиснути на цю кнопку, відбудеться перехід за посиланням, яке відкриє *Web*-сторінку Вікіпедії про дискретні елементи в браузері телефону. За таким ж принципом кнопка «Більше»

працює на всіх карточках контенту, присутня вона теж на всіх, адже повністю всю інформацію вміщувати в даному застосунку не має сенсу, тому що основною задачею є необхідність відійти від стандартної подачі інформації.

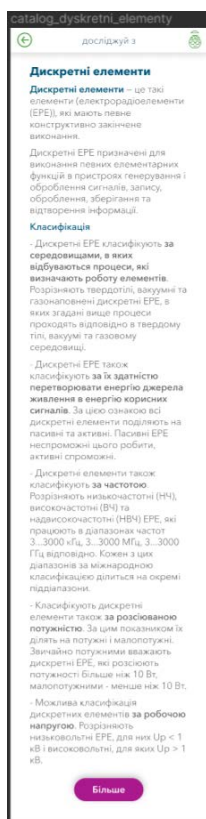


Рис. 2.8. Макет типового екрану просто інформаційної сторінки

По натисканні на екрані розділу активних елементів (див. рис. 2.7) на кнопку «Активні елементи» відбудеться перехід на типову звичайну інформаційну сторінку про активні дискретні елементи. Також на усіх інформаційних сторінках контенту є кнопка повернутись назад в лівому верхньому куті, вона повертатиме користувача на головний екран додатку (див. рис. 2.5).

По натисканні на екрані розділу активних елементів (див. рис. 2.7) на кнопку «Транзистори» відбудеться перехід на типову інформаційну сторінку про транзистори (див. рис. А.1) із можливістю запустити AR камеру для перегляду моделей транзисторів (чи будь-яких інших розглянутих в розробленому додатку елементів) у форматі доповненої реальності. Інформацію в текстовому вигляді про транзистори було теж взято із джерела *Wikipedia*. Для запуску екрану із включеною AR камерою необхідно на екрані карточки контенту натиснути кнопку «Сканувати».

При першій такій спробі обов'язково з'явиться системний поп-ап запит дозволу для відкриття камери девайсу, відповідно до правил розробки, адже не просто отримати доступ до камери телефону є незаконно. На інформаційних сторінках контенту, для якого доступний перегляд інформації з використанням технології доповненої реальності додатково можна побачити інструкцію як переглянути контент в режимі *AR*, а саме потрібно мати зображення-маркер, що, до речі, спеціально теж розташований є на цьому екрані, щоб розуміти яке саме зображення потрібне. Далі потрібно включити камеру для сканування саме цього зображення-тригера.

По натисканні на екрані розділу активних елементів (див. рис. 2.7) на кнопку «Діоди», відбудеться перехід на типову інформаційну сторінку про діоди з доступом до перегляду моделей діодів у форматі доповненої реальності. Якщо на цьому екрані натиснути кнопку «Сканувати» відбудеться перехід на екран із включеною *AR* камерою. Для показу більшої різноманітності об'єктів було вирішено розділити діоди на категорії. Відбувся поділ на *LED*, Діоди Шотки та інші діоди. Цей поділ буде видно лише на екрані *AR* камери. Таке ж саме оформлення поділу контенту буде прослідковуватись у візуалі і на інших деяких сторінках. Як можна побачити на рисунку 2.9, зверху на екрані розташовані інтерактивні елементи інтерфейсу із назвами підкатегорій. Зелений фон кнопки означає, що зараз обрано саме цю підкатегорію, чорний фон ж означає протилежне.

Екран *AR* камери містить в собі такі інтерактивні елементи інтерфейсу як сканер, кнопка увімкнення/вимкнення спалаху та назад. Елемент сканер необхідний для розуміння того, що користувачу потрібно навести камеру свого телефону на зображення, що слугує тригером для запуску доповненої реальності.



Рис. 2.9. Макет екрану включеної *AR* камери в розділі діодів

Як тільки користувач це зробить, сканер має зникнути, щоб не заважати перегляду контенту. Кнопка назад поверне користувача на сторінку про той дискретний елемент, який переглядався. Кнопка увімкнення/вимкнення спалаху є обов'язковим елементом для екранів AR камери, тому що неможливо передбачити в яких умовах користувач буде переглядати контент. Саме тому важливо надати можливість увімкнути спалах, щоб камера могла розпізнати зображення-тригер для AR в умовах поганої освітленості. За замовчуванням екрани AR камери будуть виглядати так, як на рисунку 2.10. Додатково на екрані камери будуть додаватись нові кнопки по мірі необхідності для деяких дискретних елементів, що дозволять переключатись між типами об'єктів (див. рис. 2.9). Також на екрані камери є кнопки вліво та вправо. Вони призначені для переключення між 3D об'єктами.



Рис. 2.10. Макет екрану включеної AR камери за замовчуванням

Дивитись контент в форматі доповненої реальності справді дуже цікаво, проте необхідно не забувати, що включена камера з'їдає дуже багато енергії телефону і відповідно розряджає його батарею. Розробники це чудово знають, але не всі користувачі можуть це розуміти, тому обов'язково необхідно їх попередити про це. Саме для цього було створено наступне поп-ап повідомлення-сповіщення про те, що програма працює, камера телефону включена і розряджає батарею телефону. Даний поп-ап, що зображено на рис. 2.11 буде з'являтися, якщо камера довго включена, а користувач не наводиться на необхідне зображення-тригер для AR.

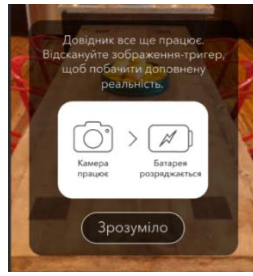


Рис. 2.11. Макет інформаційного поп-апу на екрані включеної AR камери

Другою, проте не менш важливою категорією дискретних елементів є пасивні елементи. Саме на їхню сторінку можна перейти по натисканні на кнопку «Пасивні елементи» на головному екрані (див. рис. 2.5). Категоризація пасивних дискретних елементів проведена відповідно до структури подачі інформації (див. рис. 2.1). На даному екрані також відмінність між типами контенту буде одразу видно за допомоги значків на кнопках контенту, як і в версії такого екрану про активні дискретні елементи.

В даному розділі, як говорилося раніше, теж можна потрапити на інформаційну сторінку про дискретні елементи (див. рис. 2.8).

На сторінці категорії пасивних елементів є такі кнопки (підкатегорії): Дискретні елементи, Пасивні елементи, Конденсатори, Резистори, Котушки індуктивності, Реле. По натисканні на ці кнопки відбуватиметься перехід на відповідні сторінки інформації про даний тип дискретних елементів.

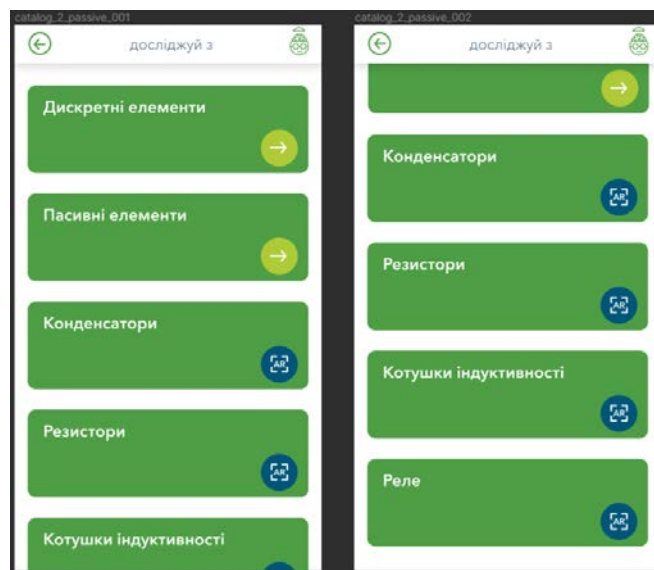


Рис. 2.12. Макет сторінки меню пасивних елементів

По натисканні на екрані розділу пасивних елементів (див. рис. 2.12) на кнопку «Пасивні елементи» відбудеться перехід на типову інформаційну сторінку про пасивні дискретні елементи.

По натисканні на екрані розділу пасивних елементів (див. рис. 2.12) на кнопки «Конденсатори», «Резистори», «Реле», «Котушки індуктивності» відбудеться перехід на типові відповідні інформаційні сторінки про обрані дискретні пасивні елементи з доступом до перегляду моделей цих дискретних елементів у форматі доповненої реальності. Інформацію в текстовому вигляді про пасивні дискретні елементи так само було взято із джерела *Wikipedia*. Тут теж присутня кнопки «Більше» та «Сканувати». Також на цьому екрані доступні будуть до перегляду зображення-тригери для *AR* камери для кожного контенту відповідно.

Для створення дизайну інтерфейсу для додатку *AR* Довідник було обрано інструментарій *Figma* через його зручність, доступність, можливість швидко працювати, створювати різні необхідні елементи за допомогою стандартних функцій. *Figma* – векторний онлайн-сервіс розробки інтерфейсів та прототипування. Працює у двох форматах: у браузері та як клієнтський додаток на десктопі користувача. Зберігає онлайн-версії файлів, з якими працював користувач.

Повний макет дизайну інтерфейсу можна переглянути за посиланням у фігмі: <https://www.figma.com/file/ObN6JyRop0iljF7FkWXqON/scenes?node-id=884%3A1>

### **2.3. Створення зображень-тригерів для відтворення контенту в доповненій реальності**

Для даного додатку було обрано маркерний тип *AR*. Про даний тип розповідалось у розділі 1. Розробка зображення-тригера є дуже важливим етапом розробки додатку з маркерною доповненою реальністю. Адже від якості тригерів буде залежати і якість користувацького досвіду взаємодії із контентом та доповненою реальністю загалом. Якщо зображення буде не достатньо хорошим по мірках тригерів для *AR*, то об'єкти, що повинні відтворитись у просторі будуть поводити себе не так, як це було задумано спочатку, зображення буде труситись, може злітати, або взагалі



не відтворюватись. Такого потрібно уникати, тому, що здавалось би, що зображення не мають такого дуже великого сенсу, легко можна десь взяти і якоесь потім, але, якщо не дотриматись вимог необхідних для *AR* – вся пророблена інша робота, яка займає чимало часу та зусиль, *3D*-моделювання, анімація, програмна частина, може піти коту під хвіст через своє неякісне відображення.

Про рекомендації для створення зображень-тригерів буде йтися далі, проаналізовано вимоги, що ставить саме обраний для роботи сервіс *MAXST*.

Програма відстеження зображень розпізнає та відстежує зображення як особливості цільового зображення (тригера), отже, можна відчутти надійне і стійке розпізнавання та відстеження без будь-яких тремтінь, використовуючи зображення з чітко розрізняючими характеристиками.

Під час завантаження тригера до диспетчера тригерів, сервер продовжує навчальний сеанс, а навчальна машина аналізує зображення для вилучення точок об'єкта та класифікує рівень якості тригера відповідно до кількості та розподілу вилучених точок об'єкта.

Рівень якості зображення відображається рейтингом по п'яти-бальній шкалі, залежно від впізнаваності та простежуваності. Зображення, оцінене як 1, може бути не впізнаваним, а саме, це означає, що з підвищенням рейтингу зображення, кількість вилучених точок рівномірно розподіляється. Тому рекомендується використовувати зображення з високим рівнем якості. Сервер може не вивчити зображення, які важко знайти в об'єкті. У цьому випадку потрібно буде завантажити інше зображення-тригер.

Зображення з неоднозначними межами між лініями та гранями важко розпізнати, оскільки важко виділити з них точки об'єкта, як наприклад зображення 2.13.



Рис. 2.13. Тестове зображення №1

Тестове зображення №1 (див. рис. 2.13) оцінено MAXST на 1. Це означає, що воно зовсім не підходить для потреб AR.

Якщо лише у певній частині зображення є точка об'єкта, як на рисунку 2.14, вона теж буде погано розпізнана.



Рис. 2.14. Тестове зображення №2

Тестове зображення №2 (див. рис. 2.14) оцінено MAXST на 1 ★. Це означає, що воно теж зовсім не підходить для потреб AR.

Якщо зображення складається з паттернів (рис. 2.15), воно не буде розпізнано як повторення подібних точок об'єкта.

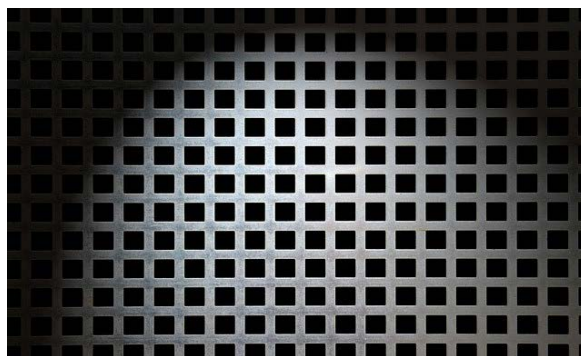


Рис. 2.15. Тестове зображення №3

Тестове зображення №3 (див. рис. 2.15) взагалі не розпізнано / опрацьовано сервером MAXST. Це означає, що воно його навіть не можна використати.

Як тільки зображення буде містити різноманітні грані та лінії, його можна чітко розрізнити. Можна розраховувати на кращу ефективність розпізнавання, якщо точки об'єктів розподіляються рівномірно, як на рисунку 2.16.



Рис. 2.16. Тестове зображення №4

Тестове зображення №4 (див. рис. 2.16) оцінено MAXST на ★★★★★. Тобто вищевказане зображення є чудовим прикладом того, який же тригер підходить для використання його в доповненій реальності. Рекомендується використовувати високоякісне зображення в якості тригера, оскільки при використанні низькоякісного зображення в якості тригера серверу не вдається добре його розпізнати [18].

Мною було створено 8 зображень-тригерів для доповненої реальності. По зображенню на підкатегорію дискретних елементів. Основу для зображень було взято на сайті *Shutterstock*. Також на зображення було накладено вставку (див. рис. 2.17) із логотипом додатка та значком, який означає, що це зображення потрібно зісканувати.

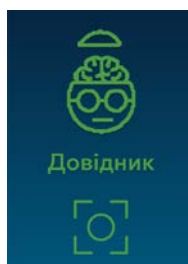


Рис. 2.17. Вставка на тигери

Усі зображення-тригери із підписами по підкатегоріям розміщено в додатку Б.

## 2.4. Інструкція для використання сервісу *MAXST AR SDK*

Як було вказано в розділі 1 для роботи було обрано *AR SDK MAXST*. Для початку роботи необхідно зареєструватися на сервері <https://developer.maxst.com/>

Тепер є можливість протестувати зображення-тригери, розроблені раніше (зазвичай тригери перевіряються прямо в момент роботи над зображеннями). Для цього необхідно перейти у вкладку *Target Manager* в *My Menu*, як це показано на рисунку 2.18.

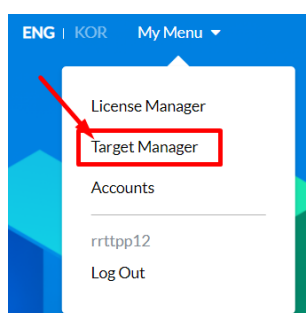


Рис. 2.18. Скріншот із сервера *MAXST*, де показано перехід в менеджер тригерів

Далі необхідно створити нову групу зображень, натиснувши на кнопку *Create New Target Group* в *Target Manager*, що зображено на рис. 2.19.

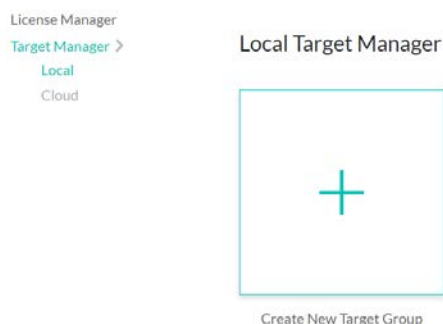


Рис. 2.19. Скріншот із сервера *MAXST*, частина 1

Наступним кроком буде дати назву групі зображень, що показано на рисунку 2.20

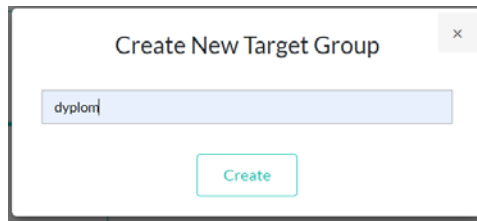


Рис. 2.20. Скріншот із сервера *MAXST*, частина 2

Коли групу зображень буде створено, вона буде відображатись на сторінці *Target Manager*, що видно на рисунку 2.21.



Рис. 2.21. Скріншот із сервера *MAXST*, де показано створену групу зображень

Після натискання на папку зображень відкриється наступна сторінка (рис 2.22), де уже можна додавати зображення, видаляти їх із сервера чи завантажувати для роботи з ними далі в середовищі розробки.



Рис. 2.22. Скріншот із сервера *MAXST*, де показано створену групу зображень зсередини

Щоб додати зображення на сервер необхідно натиснути на кнопку «*Upload Target Image*» (див. рис. 2.22). Відкриється вікно завантаження зображень – рисунок 2.23.

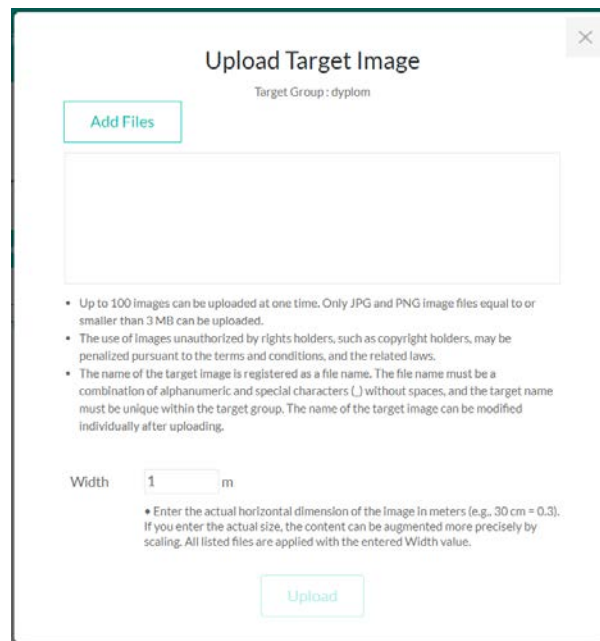


Рис. 2.23. Вікно завантаження зображень

Необхідно звернути увагу, що завантажувати зображення можна лише у форматі *JPG* або *PNG* та розмір кожного зображення не повинен перевищувати 3Мб. Також важливо правильно називати файли. Це правило стосується і повсякденного користувацького досвіду щодо файлів на комп'ютері, потрібно їх називати правильно завжди, необхідно виробити таку звичку. У назві файлів не повинно бути пробілів, потрібно використовувати лише латинські букви та цифри, без спеціальних символів. Для однієї групи зображень-тригерів назва кожного файлу повинна бути унікальною.

Щоб продовжити роботу, у вікні завантаження зображень (рис. 2.23) необхідно натиснути кнопку «*Add Files*». Після цього відкриється провідник комп'ютера, де необхідно обрати необхідні файли, підготовані зображення для тригерів. Коли файли підгрузяться одразу у вікні можна буде побачити чи можна їх завантажити, чи ні (рис. 2.24).

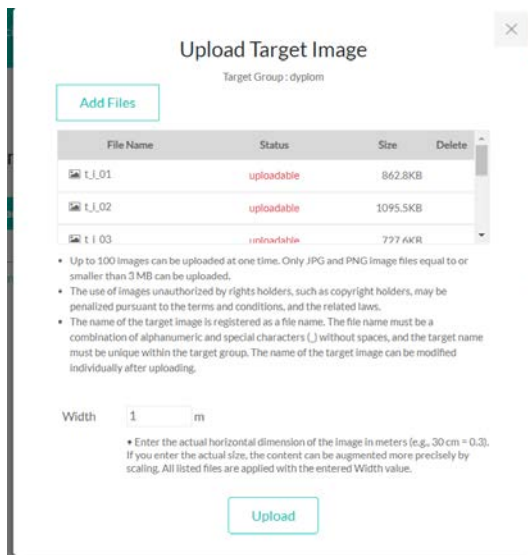


Рис. 2.24. Вікно завантаження зображень із підвантаженими файлами і їхніми характеристиками

Для запуску завантаження та розпізнавання файлів сервером необхідно натиснути кнопку «*Upload*». Після цього можна буде побачити статус завантаження зображень. Потрібно дочекатись, коли буде вказано, що 8 із 8 файлів завантажено. І далі потрібно натиснути кнопку підтвердження, що розташована знизу на рисунку 2.25.

Після цього відкриється папка, в яку було завантажено файли і можна буде побачити як оцінена якість кожного зображення (рис. 2.26).

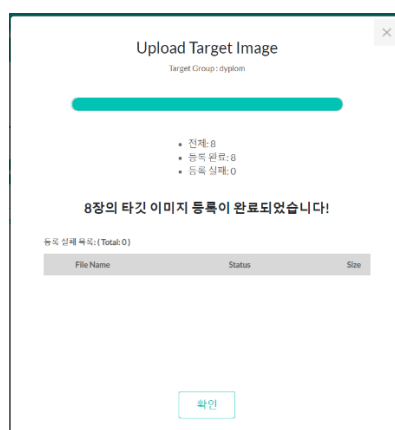


Рис. 2.25. Вікно завантаження зображень із відображенням статусу

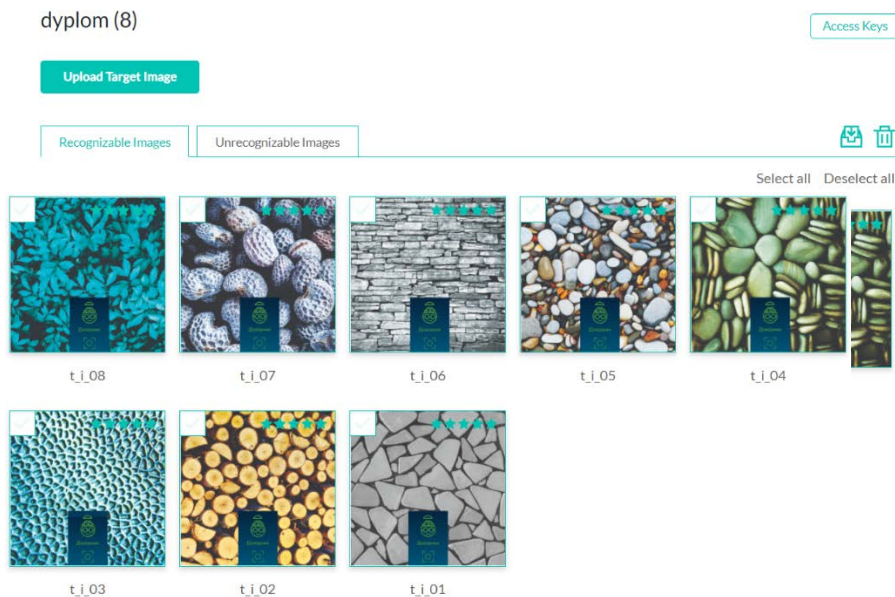


Рис. 2.26. Завантажені тригери із оцінкою якості

Усі завантажені зображення було оцінено найвищою кількістю зірочок, найвищою оцінкою рейтингу, тому що під час підготовки зображень було дотримано усіх вимог розробки тригерів.

Щоб дізнатись більше про тригер можна натиснути на нього і відкриється його сторінка (рис. 2.27), де можна переглянути назву файлу, оцінку його якості, дати створення та редагування, також є можливість його видалити. Якщо ж якість є недостатньою, біля оцінки є кнопка «*Tip*», яка стане в нагоді, щоб пригадати умови, яких потрібно дотримуватись, щоб картинка добре зчитувалась AR камерою.

Після переконання в тому, що усі зображення мають найвищий рейтинг, можна їх завантажувати для використання в проекті в *Unity*. Для цього у відповідній папці потрібно вибрати необхідні файли, за допомогою інтерактивного чекбоксу та натиснути на кнопку завантажити, що розташована біля кнопки видалити, як це показано на рис. 2.28.



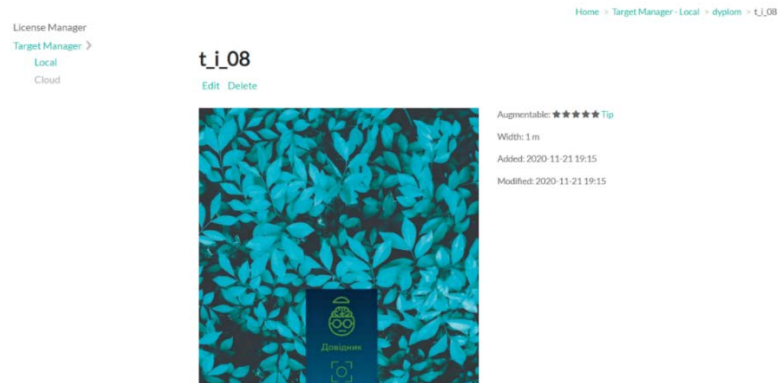


Рис. 2.27. Сторінка завантаженого зображення-тригера на сервері MAXST

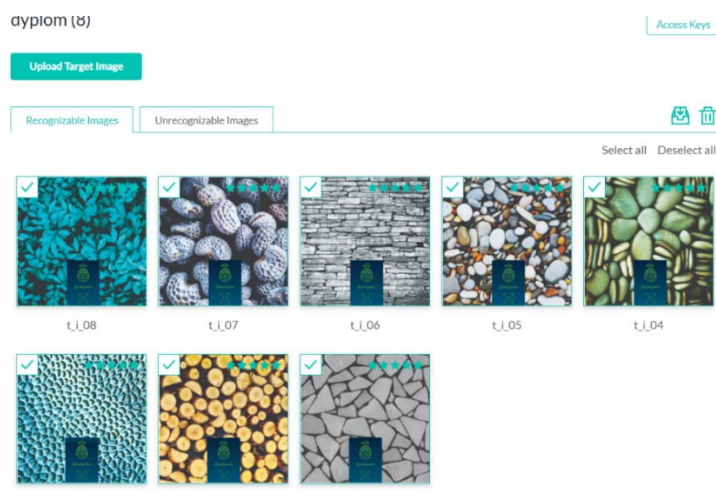


Рис. 2.28. Вибрані файли тригерів

Тоді з'явиться вікно, де потрібно обрати платформу для якої необхідно завантажити файли. Так як дана робота буде виконуватись у середовищі розробки *Unity* вибрано радіобаттон *Unity*, показано на рис. 2.29. Далі потрібно натиснути кнопку «*Download*». І після цього розпочнеться завантаження набору зображень (рис. 2.30) у форматі *Unity package* (рис. 2.31), що можна одразу відкривати, якщо встановлено та запущено середовище розробки *Unity*.

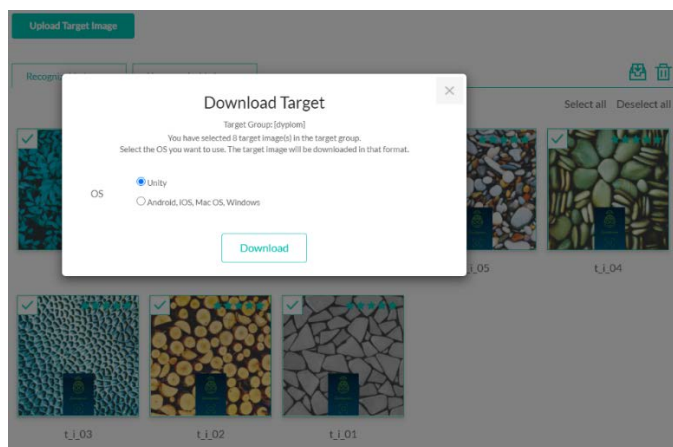


Рис. 2.29. Вікно налаштування завантаження файлів



Рис. 2.30. Вікно, де відображається процес завантаження файлів в форматі *Unity package*

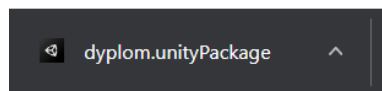


Рис. 2.31. Завантажений пакет *Unity* із зображеннями-тригерами

## 2.5. 3D контент додатку

Щоб зацікавити студентів у навчанні, важливо подавати інформацію в новому, захоплюючому вигляді. Розроблюваний додаток може врятувати викладача, якщо у нього під час лекції чи лабораторної роботи під руками не буде дискретних елементів, про які буде йти мова. Також цей додаток буде корисний у цьому випадку, щоб покращити запам'ятовування необхідної інформації та візуальних форм об'єктів. Саме для такої мети буде доречно використовувати 3D моделі дискретних елементів замість їхнього 2D зображення (хоч і це може справити теж непогане враження на студентів).

В більшості, дискретні елементи є простими по формі, тому їх можна змоделювати за досить невеликий період часу. Проте, зважаючи на малу кількість часу та чималу кількість об'єктів, які хотілося показати, було зроблене рішення скористатися сторонніми сервісами з уже готовими об'єктами. На щастя, було знайдено сайт *TraceParts*, що являється одним із ведучих постачальників цифрового 3D контенту для інженерії. Портал *TraceParts* доступний безкоштовно мільйонам користувачів САПР по всьому світу. Він забезпечує доступ до сотень каталогів постачальників і більш ніж 100 мільйонам моделей САПР і таблицями технічних даних, які ідеально відповідають конкретним потребам процесів і операцій проектування, закупівель, виробництва і технічного обслуговування.

Мною було завантажено всі необхідні моделі для показу в додатку згідно до структури (див. додаток В). Всі моделі були завантажені у форматі *OBJ* та мали додатково файл у форматі *MTL*. *OBJ* — формат файлу опису геометрії. Це відкритий файловий формат, який взято за стандарт.

Формат *OBJ* дуже простий, і задає тільки геометрію об'єкта - координати кожної вершини, її текстурні координати, нормалі, і грані, що задаються списками вершин багатокутників. Вершини багатокутників за замовчуванням задаються проти годинникової стрілки, роблячи явне задання нормалей необов'язковим. Матеріали, що описують візуальні аспекти моделі зберігаються в зовнішніх файлах *MTL*.

*OBJ* файли можна переглянути за допомогою стандартного переглядача 3D моделей від *Windows*, що показано на рисунку 2.32. Усі завантажені моделі необхідно імпортувати в проект в *Unity*. Матеріали мають підтягнутись самі.

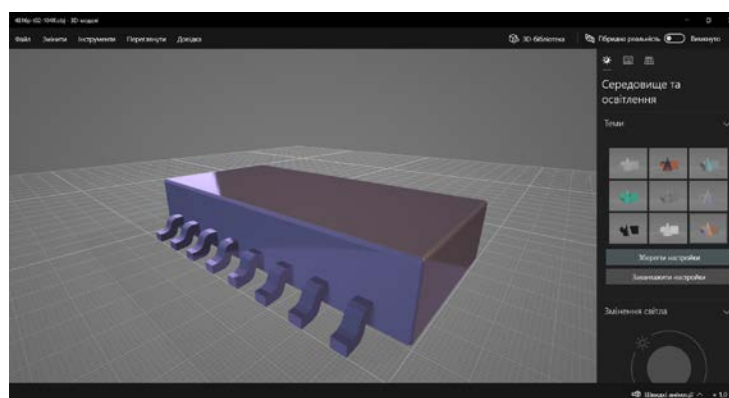


Рис. 2.32. Завантажена 3D модель

## 2.6. Розробка програмної частини

Тепер, коли уже всі підготовчі роботи проведені, а саме розроблений архітектура застосунку, розроблений дизайн інтерфейсу для застосунку, підготовані зображення-тригери для відтворення доповненої реальності, підготований увесь текстовий контент та завантажені на робочу машину усі необхідні 3D об'єкти для програми, можна приступати до створення проекту в *Unity* та безпосередньо самої розробки (кодингу) застосунку.

Як вказано у висновках розділу 1, після аналізу для даної роботи було обрано середовище розробки *Unity* через його властивості та зручність.

Для початку роботи необхідно встановити дану програму на свою робочу машину. *Unity Hub* - це основний спосіб встановити редактор *Unity*, створювати проекти та керувати своїм досвідом *Unity*. Він забезпечує центральне місце для керування вашими установками редактора, обліковими записами, ліцензіями та проектами. Також є можливість встановити редактор через командний рядок, але вважаю, що краще все ж це робити через *Unity Hub*. Для інсталяції потрібно перейти на офіційний сайт *Unity* і завантажити там установочний файл. Після встановлення *Unity Hub* потрібно завантажити необхідну версію *Unity*. Це можна зробити прямо в хабі. Для даного проекту було обрано найактуальнішу на дату початку роботи над проектом версію *Unity* 2019.4.11f1. Для її встановлення в юніті хабі необхідно натиснути на кнопку *Add* вибрати необхідну версію редактора та натиснути на кнопку *Next*, як вказано на рисунку 2.33.

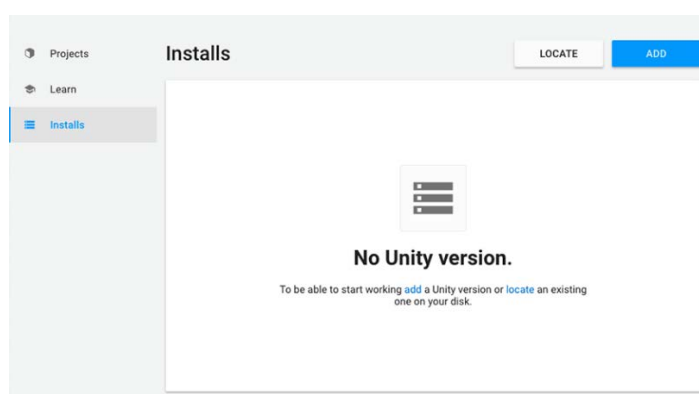


Рис. 2.33. Скріншот вікна програми *Unity Hub*

Далі потрібно вибрати необхідні модулі для встановлення і натиснути кнопку *DONE* (рис. 2.34). Якщо під час завантаження щось забудеться встановити, то можна буде зробити це потім, не потрібно переживати.

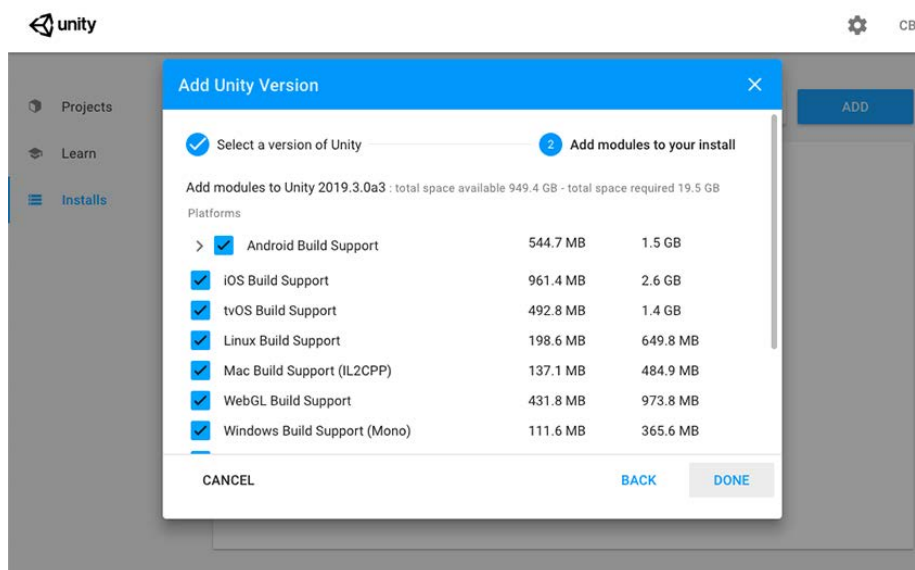


Рис. 2.34. Скріншот вікна налаштування встановлення необхідної версії редактора в програмі *Unity Hub*

В документації *MAXST* рекомендується встановлювати *Unity* версії 5.6.2, 2017 або новіші для використання *MAXST AR SDK* в *Unity*.

Наступним кроком після виконання налаштувань *Unity* є завантаження та встановлення *MAXST AR SDK* для *Unity*. Для цього потрібно з офіційного сайту *MAXST AR SDK* завантажити пакет *MAXST AR SDK*. Відкрити або створити новий проект *Unity* та імпортувати *MAXST AR SDK* для *Unity* клікнувши 2 рази на завантажений *MAXST AR SDK* пакет або обравши '*Assets / Import Package / Custom Package*' from the *Unity Editor* menu і обрати завантажений пакет.

Далі буде корисно завантажити зразковий проект від *MAXST* для *Unity*, щоб розібратись у всіх нюансах розробки із *MAXST AR SDK*. Для цього потрібно слідувати наступним крокам:

1. Потрібно встановити *MAXST AR SDK*
2. В *Unity Editor*, вибрати '*File > Build Settings*'.
3. В *Project View*, додати сцени із '*Assets > MaxsARSamples > Scenes*' папки (див рис 2.35).

4. Потрібно розмістити *Home scene* вгорі (див. рис. 2.36).
5. У вікні платформи потрібно натиснути на кнопку *Player Settings* (див. рис. 2.37).
6. У вікні *Inspector* потрібно ввести *Package Name* (див. рис. 2.38).

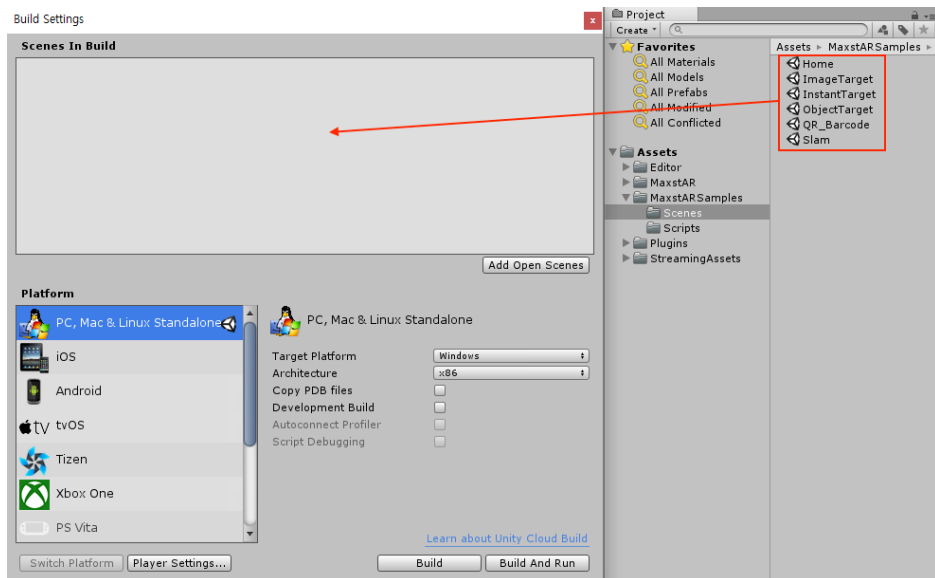


Рис. 2.35. Скріншот вікна *Unity Build Settings*, частина 1

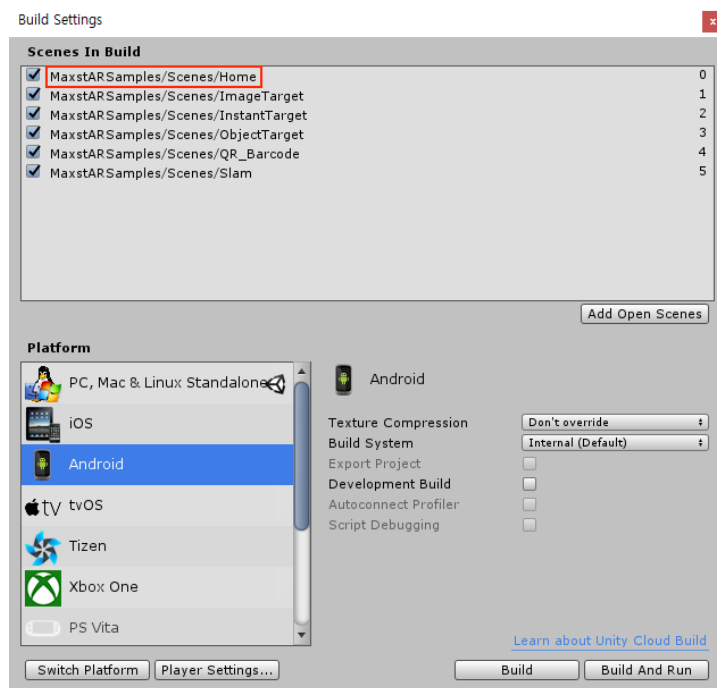


Рис. 2.36. Скріншот вікна *Unity Build Settings*, частина 2

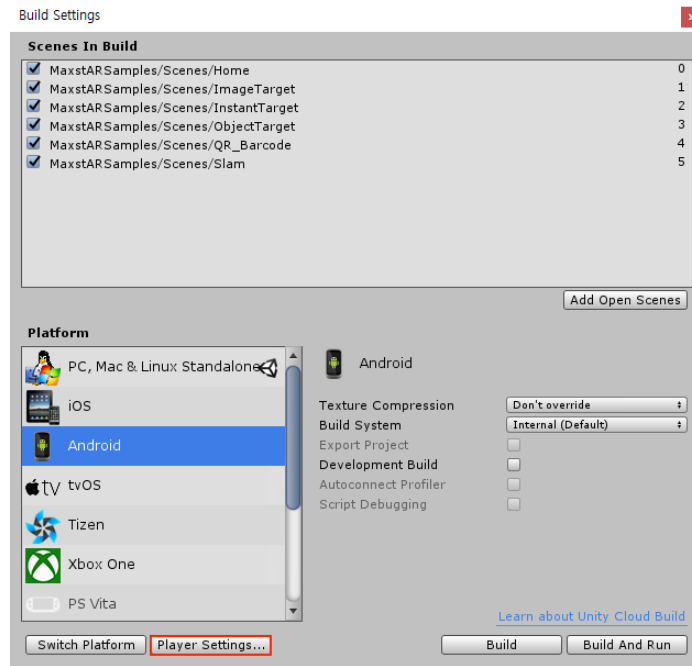


Рис. 2.37. Скріншот вікна *Unity Build Settings*, частина 3



Рис. 2.38. Скріншот вікна *Unity Build Settings*, частина 4

Далі потрібно ввести ключ ліцензії, а для цього її потрібно отримати. Сервіс *MAXST* пропонує такі види ліцензій (див. рисунок 2.39):



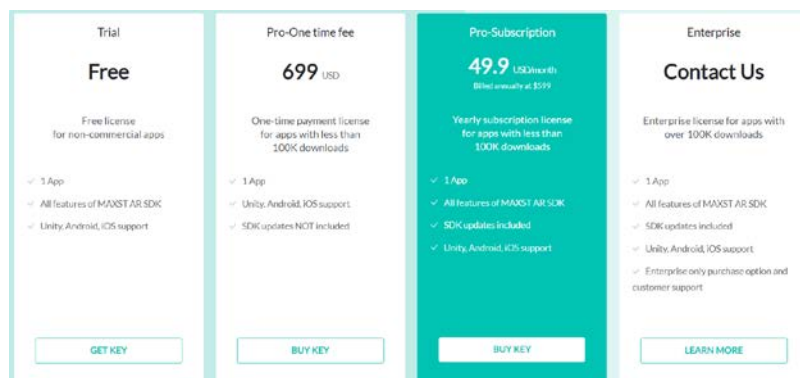


Рис. 2.39. Види ліцензій *MAXST AR SDK*

Для навчальних цілей достатньо буде пробної ліцензії типу *Free*. Також рекомендується під час розробки усіх проектів спочатку використовувати безкоштовну ліцензію для виключення ситуацій, що посередині розробки буде визначено, що *MAXST AR SDK* не підходить для досягнення поставлених цілей, а його вартість уже оплачена. Ліцензія типу *Free* призначена для некомерційних додатків та підтримує усі можливо необхідні платформи.

Щоб отримати ключ ліцензії можна у цьому вікні натиснути на кнопку *Get Key* на вибраному типі ліцензії, що перенесе на наступний екран оформлення ліцензії (див. рис. 2.40).

**Get SDK License**

Please select the license type you want to get.  
For more detailed information on licensing, please refer to the [Pricing](#) menu.

Free: For non-commercial purposes only. A watermark is included.  
 Pro - One-time fee (\$699): The one-time payment license doesn't include SDK engine updates. Minor updates such as bug fixes are included.  
 Pro - Subscription (\$599/year): The annual subscription license includes continual SDK engine updates, which reflect the frequent advances in the ever-evolving AR.  
 Enterprise: License that supports customization through separate consultation. Please first contact us for this option.

**App Information**

App Name:

App ID:

- You can only use one App ID for one license.
- The iOS Bundle Identifier and the Android Package Name must be the same so that you can distribute your app to both App Store and Google Play with one license.
- Please note that you can not modify App ID once it has been set.
- Please note that you can not use App ID that not compliant with store policies. (Check App ID naming policy of each store: [Google Play](#), [App Store](#))

[Get](#)

Рис. 2.40. Екран оформлення ліцензії *MAXST AR SDK* для розроблюваного додатку

У цьому вікні потрібно пересвідчитись, що обрано вірний тип ліцензії, в даному випадку *Free: For non-commercial purposes only. A watermark is included*. Далі потрібно






ввести назву додатку, його можна буде потім змінити. Для цього проекту додаток було названо *Rob\_T\_diplom*, що відповідає першим трьом буквам прізвища студента, першій букві імені та назві роботи.

Далі потрібно ввести *App ID*. Дуже важливо його правильно створити, потім буде використовуватись в юніті. Для даної роботи було створено такий ідентифікатор додатку *com.rob.tet.ua*.

Потрібно вкотре перевірити, що всі дані введено та вибрано вірно і натиснути на кнопку *Get*. Після цього відкриється вікно інформації про оформлену ліцензію (див. рис. 2.41)

### SDK License Details



License #	18935	License Type	Free Add
App Name	Rob_T_diplom 	App ID (Android Package Name or iOS Bundle Identifier)	com.rob.tet.ua
Registration Date	2020-11-21	Next billing date (estimated)	-
Payment Method	-	Subscription Status	-
License Key - version - 			

[Delete this SDK License](#)

Рис. 2.41. Вікно інформації про ліцензію

В даному вікні можна побачити назву додатка, яку за необхідності можна змінити, вона не впливає ні на що, також можна побачити дату створення ліцензії, тип ліцензії і найважливіше для наступного етапу, можна витягнути ключ ліцензії, який потрібно буде ввести в редакторі *Unity*. Ключ ліцензії можна обрати відповідно до встановленої версії *MAXST AR SDK*. На рисунку 2.42 показано саме вибір необхідної версії *MAXST AR SDK* для отримання ключа ліцензії

## SDK License Details

License #	18935	License Type	Free <a href="#">Add</a>
App Name	Rob_T_dyplom 	App ID (Android Package Name or iOS Bundle Identifier)	com.rob.tet.ua
Registration Date	2020-11-21	Next billing date (estimated)	-
Payment Method	-	Subscription Status	-
License Key	<input type="text" value=""/> 		



[Delete this SDK License](#)

[Privacy](#) [Terms of Use](#)

Рис. 2.42. Скріншот вікна інформації про ліцензію

Після вибору версії *MAXST AR SDK* для отримання ключа ліцензії збоку з'явиться ключ (рис. 2.43), який потрібно скопіювати та ввести у відповідне місце в редакторі *Unity*.

## SDK License Details

License #	18935	License Type	Free <a href="#">Add</a>
App Name	Rob_T_dyplom 	App ID (Android Package Name or iOS Bundle Identifier)	com.rob.tet.ua
Registration Date	2020-11-21	Next billing date (estimated)	-
Payment Method	-	Subscription Status	-
License Key	<input type="text" value="Gv3r3I7fuNFpWX0D8Jh/rm3nLBxR9paZntTTC8HZ59g="/> 		

[Delete this SDK License](#)

Рис. 2.43. Скріншот вікна інформації про ліцензію, де тепер видно ключ ліцензії

Під час створення проекту в *Unity* потрібно внести такі дані, як назва компанії, назва продукту та версія, також є можливість додати іконку додатка, яку користувач буде бачити на робочому столі свого девайсу, також там відобразатиметься назва продукту, що показано на рисунку 2.44.

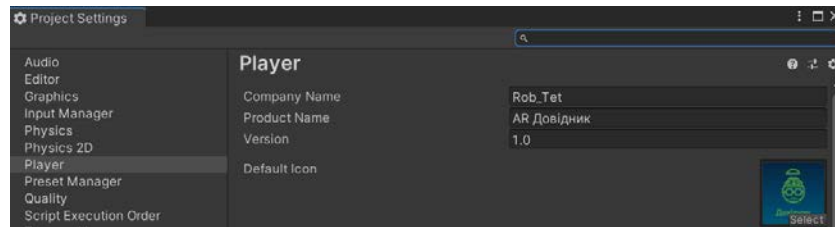


Рис. 2.44. Вікно налаштувань проекту в *Unity*, частина 1

Прогорнувши в налаштуваннях нижче, потрібно заповнити такі дані як назва пакету, що має відповідати *App ID*, яке було введено на сервері *MAXST* під час створення ліцензії та ключа (рис. 2.45).



Рис. 2.45. Вікно налаштувань проекту в *Unity*, частина 2

Далі до проекту необхідно додати *AR* камеру та ввести ключ ліцензії, як показано на рисунку 2.46.

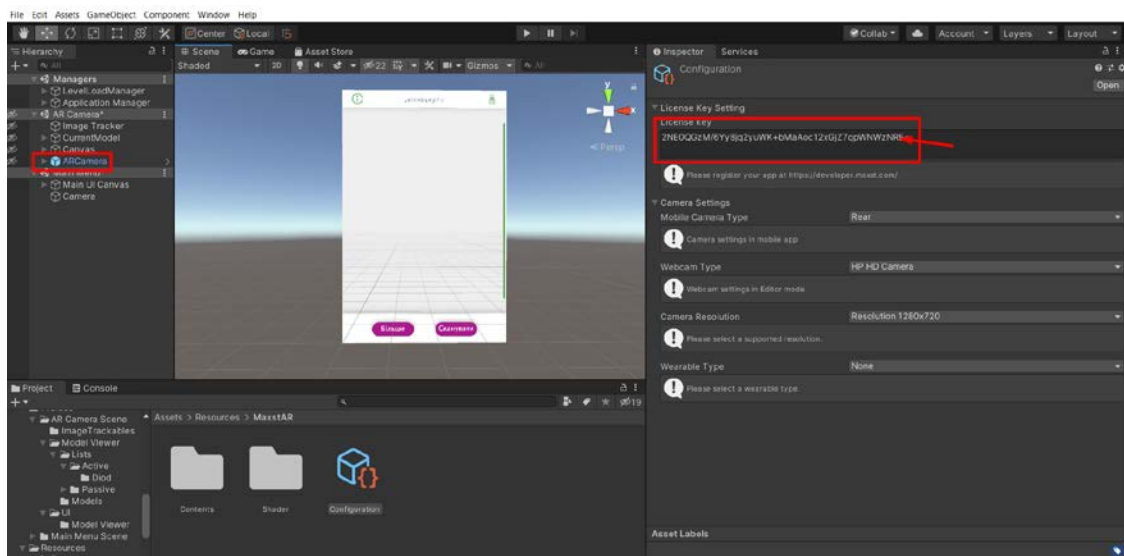


Рис. 2.46. Вікно налаштування *AR* камери в *Unity*

В даному проекті було зроблено 3 сцени – менеджер, головний екран і *AR* камера. При запуску білда одразу запускається сцена менеджера, що містить у собі 2 менеджера, а саме *LevelLoadManager* і *Application Manager*. Менеджер *LevelLoadManager* призначений для переключення між трьома сценами, як це можна побачити на рисунку 2.47.

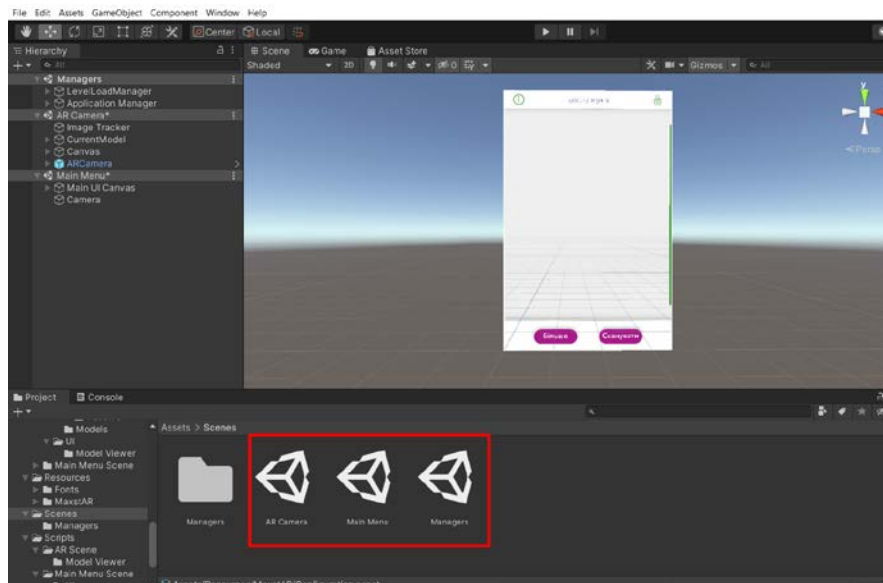


Рис. 2.47. Скріншот вікна *Unity* з менеджерами

*Application Manager* призначений для завантаження наступної сцени при запуску білда, сцени головного меню. Обидва менеджери описані вище, мають змінну *Don't Destroy On Load – true*, тому вони не знищуються при зміні сцени.

В сцені головного меню знаходяться такі об'єкти як канвас інтерфейсу і камера, що показує цей канвас. Інтерфейс головного меню складається із чотирьох об'єктів: бекграунд (задній фон), верхній край (хедер, тобто всі елементи, що знаходяться зверху), частина посередині (тут знаходиться найбільша частина контенту, також тут є скрол і вьюпорт, який обмежує краї для контенту, тобто, якщо контенту більше і він виходить за краї, то за краями вьюпорта він показуватись не буде), та нижній край (футер). На початку запуску додатка, скрипт об'єкта *Main Menu* запускає перший вказаний контент у вьюпорті, а саме *Main Catalog* (головний екран додатку, див. рис. 2.5). Кожен контент має свій скрипт контенту, в якому визначається чи є попередній контент і, якщо він є, то в хедері зліва буде відобразитись кнопка назад, якщо нема попереднього контенту, то буде відобразитись кнопка інформації. Всередині контенту *Main Catalog* знаходяться 2 кнопки. На кожній із кнопок встановлений префаб *Main Catalog Button*, де можна побачити активний контент та той, на який переключиться програма при натисканні на кнопку (див рис. 2.48).

При переході на наступний контент, наприклад *Active Elements Catalog* тепер можна побачити 2 типи контенту: *Read i Scan*. *Read* відповідає за тип інформаційної

сторінки, де розміщено лише текст, а *Scan* – за ті сторінки, де є можливість перейти на екран *AR* камери.

Для контенту типу *Read* доступні такі налаштування як назва, текст, посилання для кнопки «Більше» та назва кнопки (див. рис. 2.48).

Для контенту типу *Scan* доступні всі ті ж самі налаштування, що вказані вище і додаються ще зображення тригера, що показується у вікні вьюпорта, зображення інструкція для використання *AR*, файл зчитуваного зображення від *MAXST*, список моделей(див. рис. 2.49). Важливо всі завантажені компоненти від *MAXST* засовувати в папку *Streaming Assets*, тому що це єдина папка, яка не конвертується при білді.

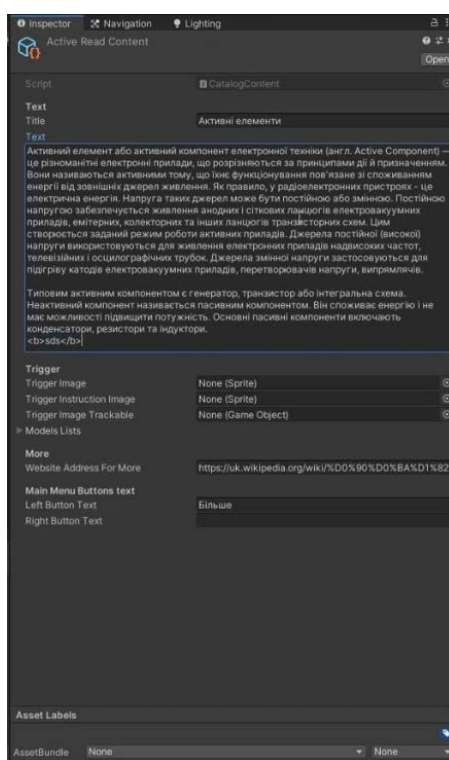


Рис. 2.48. Скріншот вікна *Unity* з налаштуванням контенту типу *Read*

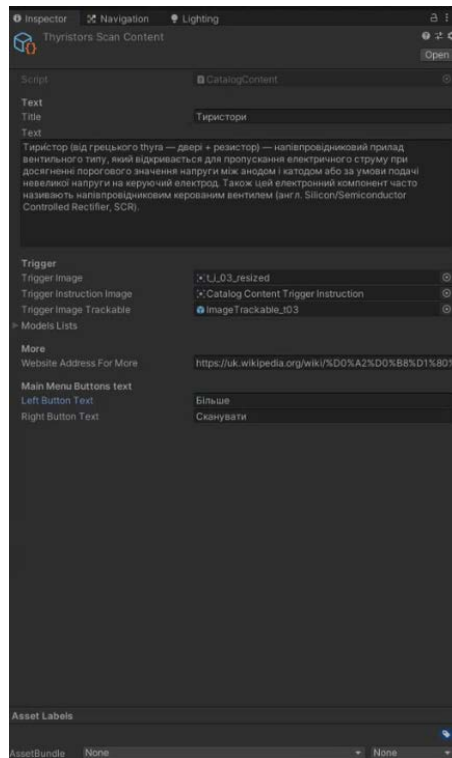


Рис. 2.49. Скріншот вікна *Unity* з налаштуванням контенту типу *Scan*

У складі футера знаходяться 3 кнопки: ліва, права та середня. Середня кнопка зроблена з метою показу її, якщо в налаштуваннях контенту функціонал наданий лише одній лівій кнопці. Тобто, якщо заповнене лише поле для ліво кнопки, то ліва і права кнопки стають невидимі, а показується середня.

Із вікна контенту типу *Scan* можна запустити третю сцену – *AR Camera* по натисканні на кнопку «Сканувати».

Сцена *AR Camera* складається з *Image Tracker* (стандартний скрипт від *Maxst*, який ініціалізує камеру і.т.д), *AR Camera* (теж стандартна штука від *Maxst*, куди було введено ключ ліцензії). Всередині *AR Camera* знаходиться об'єкт *Current Model*, це об'єкт, в який потім будуть підтягуватись моделі із списку моделей. *Current Model* буде ініціалізовуватись в об'єкті, наприклад, *Image\_Trackable\_t04* і, поки камерою не буде затригерено необхідну картинку, об'єкти, що знаходяться всередині *Image\_Trackable\_t04* не будуть видимі. Коли камера зчитала тригер, починає працювати *Model Viewer* із *Canvas AR Camera*, в якому можна контролювати, наприклад, швидкість повороту.

Також канвас *AR* камери містить кнопку назад, що викликає *Application Manager*, який грузить *LoadMainScene* та кнопку спалаху, яка бере функцію з стандартного *MAXST sdk* і ставить її *true* або *false*.

Приклад роботи програми наведено на скріншотах рисунків 2.50 та 2.51.

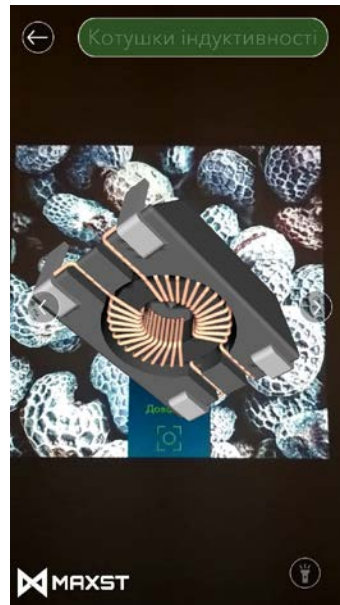


Рис. 2.50. Приклад роботи програми, показ моделі котушок індуктивності



Рис. 2.51. Приклад роботи програми, показ моделі транзисторів

Програмний код буде розміщено в додатку Г.

## Висновки за розділом

Під час виконання даної дипломної роботи було створено ідею для додатка для використання в навчальному процесі; розроблено користувацький інтерфейс відповідно до загальновідомих стандартів. Також для використання маркерної технології доповненої реальності було розроблено зображення-тригери для відтворення доповненої реальності відповідно до вимог сервісу *MAXST*. Також описано покрокову інструкцію для використання сервісу *Maxst* та імплементація технології доповненої реальності в проект *Unity*.

Результатом даної роботи є працюючий додаток для девайсів з операційною системою *Android*. Додаток розроблено відповідно до створеної структури подачі інформації про дискретні елементи, до розробленого дизайну на основі створеної структури, із використанням усіх завантажених *3D* моделей, із використанням сервісу від *MAXST* та в середовищі розробки *Unity*. Даний додаток можна вважати основою для подальшого розширення функціоналу програми та доповнення бази дискретних елементів або і всієї іншої інформації, що буде необхідна для навчального процесу з дисципліни Мікроелектроніка. Також даний додаток може бути прикладом для споріднених програм для використання в навчальному процесі з інших предметів. Дуже важливо використовувати нові технології для росту студентів, для надання їм можливостей дізнаватись інформацію в цікавому форматі, отримувати знання про нові технології, зацікавлюватись в них.



## РОЗДІЛ 3

### ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ В НАВЧАЛЬНОМУ ПРОЦЕСІ

Використання технології доповненої реальності уже визнано важливою частиною освіти. Дана технологія рекомендується як рішення для забезпечення ефективного навчання в школах та університетах [27]. Дослідження показали, що використання технологій у навчанні підвищує рівень досягнень студентів, пропонує важливий інструмент навчання, що привертає увагу студентів до змісту курсу та робить процес навчання активним та ефективним [28]. Швидкий розвиток технологій призвів до того, що матеріали, що використовуються в навчальних цілях, постійно потрібно оновлювати. Зараз на додаток до дошок та експериментів у лабораторіях використовуються інтерактивні дошки, програмне забезпечення для моделювання та середовища віртуальної реальності.

Хоча *AR* і використовується в інших галузях протягом уже тривалого часу, проте спостерігається, що дослідження про потенціал використання даної технології у навчальних закладах почали проводитись лише недавно.

*AR* є простим у використанні, проте йому уже вдалося привернути увагу з точки зору його використання в освіті завдяки своїм педагогічним перевагам. Помічено, що інструменти *AR* використовуються широким колом учасників - від учнів дошкільних шкіл до аспірантів та викладачів у широкому діапазоні областей – від навчання грамоти до астрономічної освіти. Таке інтенсивне використання *AR* за короткий час дає уявлення про потенціал, який технологія *AR* пропонує для освітніх середовищ.

#### **3.1. Дослідження ефективності використання технології доповненої реальності для навчального процесу**

Дослідження повідомляють про такі переваги використання *AR* в освітніх середовищах:

- Підвищення інтересу студентів у навчанні [29-33];
- Підвищення мотивації до навчання в студентів [30, 32, 34-35];
- Забезпечення захоплення навчанням [32, 36-37];
- Полегшення навчання [30, 33-34, 38-42];
- Конкретизування абстрактних понять [43-44];
- Розвиток просторового розуміння, активності [45-47];
- Збагачення досвіду взаємодії [42, 48];
- Підвищення рівня залучення [49-52].

Дослідження, підкреслюють, що *AR* може розвивати навички, які очікуються від сучасних учнів, таких як вирішення проблем, робота в групах, різнобічне оцінювання та розуміння різних точок зору. Як описано вище, використання *AR* в освітніх середовищах має багато переваг. Однак можна констатувати, що найважливішою перевагою *AR* є інтеграція внеску віртуального середовища в реальність класу. *AR* включає всі ці переваги в моделюванні та добавляє в реальне середовище, в якому ми існуємо, що разом перевершує віртуальну реальність. Тому це дозволяє студентам брати активну участь у одночасній взаємодії з реальним середовищем та з віртуальними об'єктами. *AR* може бути використаний як ефективний інструмент для студентів для досягнення цілей навчання.

Технологія *AR* забезпечує важливу можливість візуалізації в освітньому середовищі. Тривимірний вміст уроку можна розробити за допомогою технології *AR*. Тривимірний вміст уроку надає студентам можливість розглядати предмети з такими характеристиками, як розташування, кут, обертання та поворот.

Завдяки *AR* можна отримати умови таких середовищ, які неможливо створити в реальних умовах та, наприклад для небезпечних експериментів [34, 43].

Ці всі переваги роблять *AR* важливим інструментом викладання абстрактних понять. Наука – одна з галузей, на якій зосередились дослідження *AR*, оскільки вона включає в себе численні абстрактні поняття.

Досліджень використання технології *AR* для навчальних процесів із кожним роком стає все більше, отже, все більше вчених цікавляться цією темою та вважають за необхідне її розвивати. Найбільше досліджень є про використання технології

доповненої реальності саме для науки, гуманітарних наук та мистецтва. Найменш вивченими галузями для використання *AR* є використання даної технології для підготовки вчителів та для сільського господарства.

*AR* в основному застосовується у вищих навчальних закладах та на обов'язкових рівнях освіти для мотивації студентів. Основною метою використання *AR* було пояснення цікавої теми та надання додаткової інформації. Також зростає частота використання технології доповненої реальності для виконання лабораторних робіт та експериментів.

Більшість досліджень використання технології доповненої реальності для освіти розглядали середні зразки досліджень (від 30 до 200 учасників), і більшість досліджень використовували змішані методи оцінки. Найпопулярнішими методами збору даних були анкетування, інтерв'ю та опитування, і більшість досліджень були перехресними.

У цьому дослідженні мобільне програмне забезпечення використовується для комп'ютерних наук. Через сферу теоретичних та прикладних досліджень з основ інформації та обчислень, Комп'ютерні науки мають широкий спектр абстрактних та технічних предметів, які, як відомо, важко вивчити серед студентів. Отже, важливо дати можливість студентам отримати необхідні знання та практичні аспекти на курсах цієї галузі, використовуючи кращі навчальні методи. Використання інструменту електронного навчання на базі *AR* призведе до ефективного підходу до досягнення освітніх цілей.

Зважаючи на розроблену програму, на даний момент нею можна користуватись лише із девайсів із операційною системою *Android*. Студенти можуть користуватись цим ПЗ із своїх девайсів. Використання програми *AR* Довідник дуже легке. В даному додатку використана тривимірна технологія зображення об'єктів для кращого сприйняття. Також помічено, звертання уваги в реальному світі на ті ж об'єкти, що були в побачені в доповненій реальності та їхню впізнаваність, що знову ж таки свідчить про покращене запам'ятовування поданої інформації.

Експериментальна оцінка:

Всього в експерименті взяло участь 200 студентів, що навчаються за освітніми програмами, пов'язаними з комп'ютерними науками з різних університетів. Половина студентів (100), формували експериментальну групу, де контент, що базується на технології доповненої реальності використовувався для супроводу навчального процесу. Для іншої половини студентів, що формували контрольну групу, використовувались для супроводу навчального процесу лише наукові відео. Групи були створені збалансовано, відповідно до їхньої академічної успішності. Після експерименту було зроблено такі висновки, що занесено в таблиці 3.1 [54]:

Таблиця 3.1

Група	Кількість студентів	К-сть студентів, що успішно пройшли курс*	Середня оцінка	Медіана	Стандартне відхилення
Експериментальна	100	84	73,88	74	14,19
Контрольна	100	61	60,28	60,80	19,80

\* Кількість студентів, оцінка успіху яких (яка обчислюється за  $(0,4 * \text{оцінка за іспитом посередині курсу}) + (0,6 * \text{оцінка за підсумковим іспитом})$ ) дорівнює або перевищує 60 зі 100.

Вкінці навчального семестру також студенти із експериментальної групи були опитані таким чином (табл. 3.2) [54].

Результати, отримані в ході експериментальних та опитувальних сесій, показують, що розроблене мобільне програмне забезпечення покращило навчальні досягнення студентів та забезпечило ефективний спосіб, що спрощує вивчення абстрактних технічних предметів, що стосуються галузі комп'ютерних наук.

Таблиця 3.2

№	Судження	Оцінка				
		1	2	3	4	5
1	Мені сподобався навчальний процес, проведений за підтримки цього інструменту - програми.	0	0	7	10	83
2	Завдяки цьому інструменту - додатку було легше вивчати важкі предмети.	0	1	5	14	80
3	Було важко використовувати цей інструмент.	78	11	9	2	0
4	За допомогою цього інструменту - програми я почувався впевненіше у вивченні предметів курсу	0	0	6	13	81
5	Використовуючи цей інструмент - додаток, було ефективніше вивчати предмет.	0	3	5	16	76
6	Я вважаю за краще використовувати цей інструмент - додаток, а не інші підходи до електронного навчання	0	2	6	13	79
7	Я не хочу брати участь знову в такому навчальному процесі	80	17	3	0	0
8	Матеріали (анімація, відео ... тощо), переглянуті на інструменті, були привабливими.	0	0	1	18	81
9	Рівень моїх навчальних досягнень покращений завдяки цьому інструменту – додатку	0	0	3	17	80
10	Цей інструмент - додаток дозволив мені швидше вчитися.	0	2	5	16	77
*Шкала оцінювання: 1 – категорично не згоден; 2 – не погоджуюся; 3 – немає думки; 4 – погоджуюсь; 5 – повністю згоден.						

Важливість дослідження пов'язана з науково-дослідницькими зусиллями, що проводяться для оцінки ефективності AR щодо покращення досвіду навчання серед абстрактних або технічних курсів. Згідно з результатами, отриманими в результаті оціночних робіт, програмна система продемонструвала ефективну та успішну роботу щодо вдосконалення навчального досвіду та була прийнята як кращий та приємніший спосіб ніж інші можливі підходи до електронного навчання. На майбутнє

залишається вдосконалювати систему інструментів програмного забезпечення, додаючи нові функції. Крім того, також потрібно проводити ще більше досліджень використання такого програмного забезпечення для оцінки ефективності в різних випадках.

Поява нових технологічних інновацій, таких як технології доповненої реальності, змогло продемонструвати слабкі сторони традиційних методів навчання, а також потенціал їх вдосконалення [55].

Хочу поділитись ще одним цікавим дослідженням, о було проведено університетом Маккензі в Бразилії. Було відібрано студентів комп'ютерних наук, інформаційних систем, цифрового дизайну та мультимедійного виробництва. Ці курси бакалаврату були обрані, оскільки всі вони досліджують людські та програмні аспекти, слідуючи сучасним технологічним та технічним тенденціям розвитку. Незважаючи на те, що Інформаційні системи не зосереджуються на розробці систем як Комп'ютерні науки, цей курс має напрямок досліджень та вивчення в інтерактивних медіа. Курси з дизайнерських дисциплін (цифровий дизайн та мультимедіа-виробництво) мають велику участь в інноваційних цифрових технологіях. Для студентів було підготовано опитування, результати яких будуть вказані нижче на рисунках 3.1, 3.2, 3.3.

По результатам опитування стало відомо, що більшість студентів були вже знайомі з AR (рис. 3.1), проте не всі і частина тих, що не знали, була із комп'ютерної галузі, це важливо зауважити. Також дізнались про те, що велика частина опитуваних, хоч і знала технологію, проте не користувалась нею, зазвичай через те. Що не було можливості, хоч і було одностайно виголошено бажання.

На рис. 3.1 зібрано в графік відповіді на запитання студентів чи знають вони про технологію AR.

На рис. 3.2 зібрано в графік відповіді студентів чи вважають вони, що технологія доповненої реальності допомагає освоїти складну інформацію.

На рис. 3.3. зібрано результати опитування чи вважають студенти, що використання технологій доповненої реальності підвищує мотивацію до навчання.

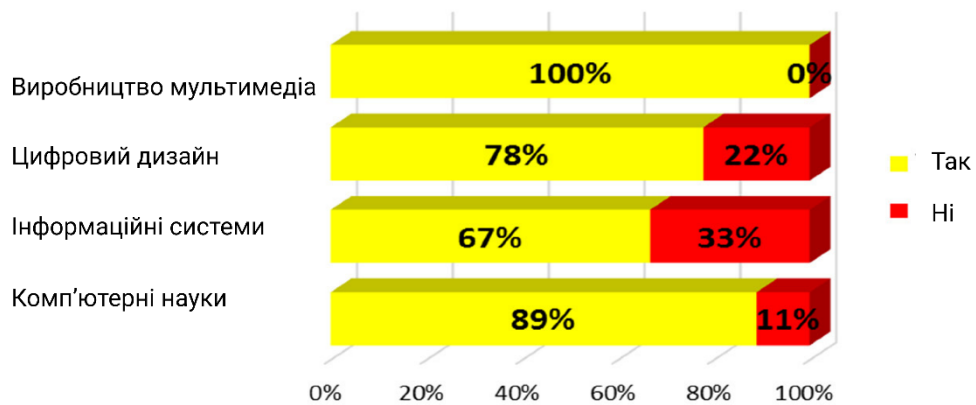


Рис. 3.1. Результати опитування студентів чи знають вони технологію AR

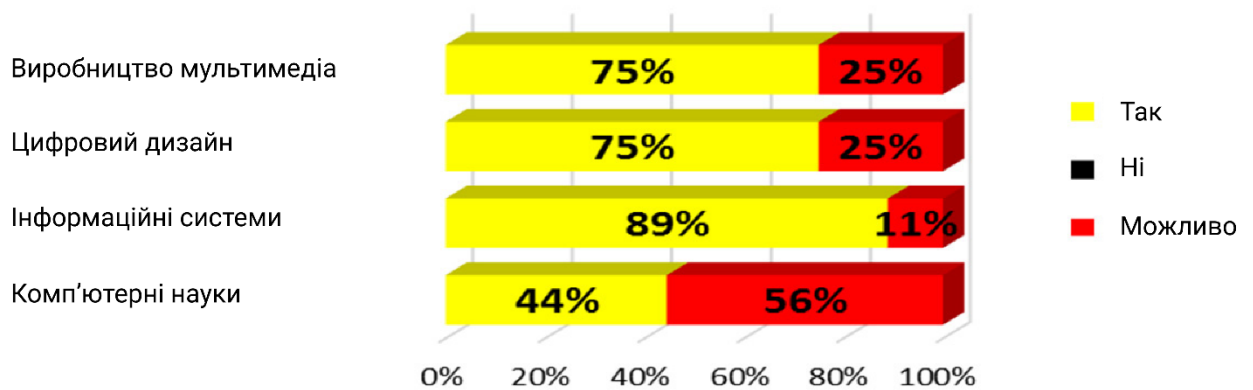


Рис. 3.2. Результати опитування студентів чи допомагає AR в освоєнні складного матеріалу

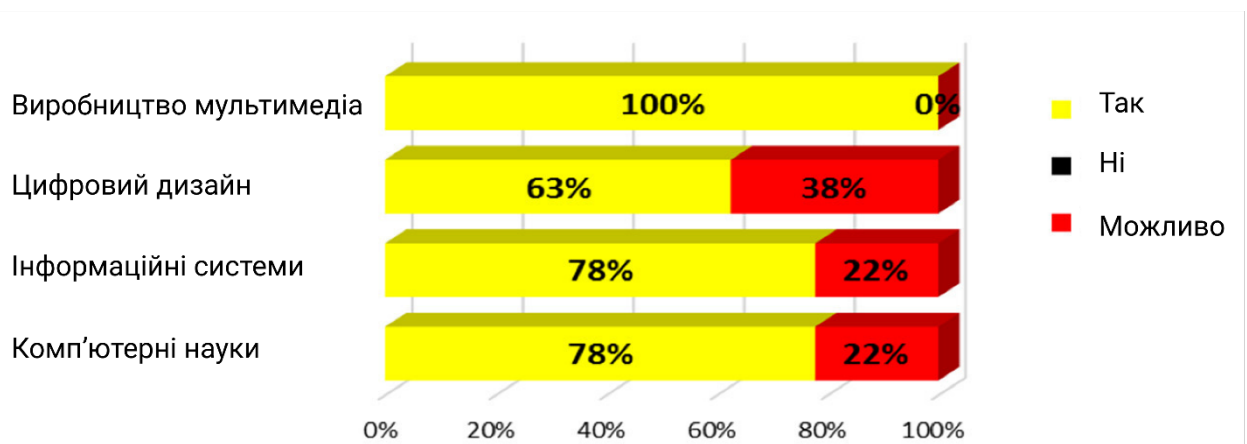


Рис. 3.3. Результати опитування студентів чи підвищує AR мотивацію

По результатах опитування, можна зробити висновок, що дії, які дають змогу краще зрозуміти технологію і забезпечують навчання, можуть призвести до значного

збільшення існуючої кількості розробників *AR*, оскільки зацікавленість студентів велика.

Результати також демонструють сильну тенденцію до використання та розвитку додатків *AR* у сфері ігор та розваг, і, як очікувалося, можна підтвердити вагомий вплив мобільних пристроїв у цьому поколінні розробників.

Очевидно, що є консенсус щодо важливості цієї технології у процесі викладання та навчання, що полегшує розуміння складних питань, забезпечує розширення вмісту та підвищує мотивацію та інтерес користувачів. Однак залишається незрозумілим використання *AR* як засобу для сприяння спільній роботі [56].

Також було зроблено дослідження Кейп-Таунським університетом, без поділу на контрольну та експериментальну групи. Дослідження вивчало відмінності в мотивації навчання студентів до та після використання мобільного додатка з доповненою реальністю. Загалом 78 учасників скористались мобільним додатком із доповненою реальністю та заповнили анкети для до та після взаємодії із додатком із технологією *AR*. Результати показали, що використання мобільного додатка з доповненою реальністю підвищує мотивацію навчання учнів. Фактори уваги, задоволеності та впевненості у мотивації були підвищені, і ці результати виявились значними. Хоча коефіцієнт релевантності продемонстрував зменшення, він виявився незначним. Результати дослідження можна побачити у таблиці 3.3.

Таблиця 3.3

	До	Після	Різниця у відсотках
Увага	2.93	3.83	30.72% підвищення
Відповідність	3.37	3.26	3.26% зниження
Впевненість	2.98	3.30	10.74% підвищення
Задоволення	2.96	3.33	12.50% підвищення
Загально	3.05	3.49	14.43% підвищення

Результати цього дослідження додаються до попередніх досліджень, проведених для вимірювання мотивації навчання учнів після використання



навчального інструменту *AR*. Подібні дослідницькі дослідження слід проводити протягом тривалого періоду часу, щоб зменшити ефект новизни, який міг виступити тривожним фактором [57].

### **3.2. Дослідження впливу використання *2D*, *3D* та *AR* технологій у навчальному процесі**

Також доцільно в даній пояснювальній записці згадати дослідження із Тайванського університету, що вивчало використання двовимірного зображення, тривимірного зображення та доповненої реальності. Це дослідження мало на меті обговорити вплив матеріалів, заснованих на моделюванні, на ефективність навчання. Квазіекспериментальне дослідження було прийняте, щоб відтворити ефекти навчання студентів у цьому дослідженні. Крім того, за допомогою Т-тестів на незалежних зразках це дослідження отримало статистичні дані щодо ефективності різних навчальних конструкцій навчального блоку між студентами із використанням *2D*-анімаційних матеріалів (*2D*), *3D*-моделювальних матеріалів (*3D*) та матеріалів доповненої реальності (*AR*).

Учасниками були 104 студенти університету на півночі Тайваню, а саме 41 особа чоловічої статі (39,42%) та 63 жіночої (60,58%), які навчались на двох вибіркового курсу "Комп'ютерна анімація та додатки". Зразки були розділені на три групи та випадковим чином віднесені до експериментальних умов (група *AR*) та контрольної групи (*2D*-анімація та *3D*-моделювання). На підставі моделювання навчальної конструкції експериментальна група та контрольна група прийняли абсолютно однакові процедури, але різні допоміжні матеріали. Експериментальна група (група *AR*) проводила одиничні заходи за допомогою матеріалів *AR*. Тим часом контрольна група (*2D* та *3D* група) проводила заходи з використанням *2D* анімації та *3D* моделюючих матеріалів.

У рис. 3.4 наведено результати дослідження, описова статистика та підсумок навчальних досягнень при використанні різних підходів, вказано показники до та після імплементації допоміжних матеріалів різними варіаціями.

По-перше, тест на рівність був проведений на оцінці успішності учасників до тестування на основі моделювання. Результат не був значущим. Припущення про однорідність, що дисперсія похибки залежної змінної однакова для груп, було збережено. Результати показали, що вплив різного навчального підходу на ставлення учнів був дуже значним.

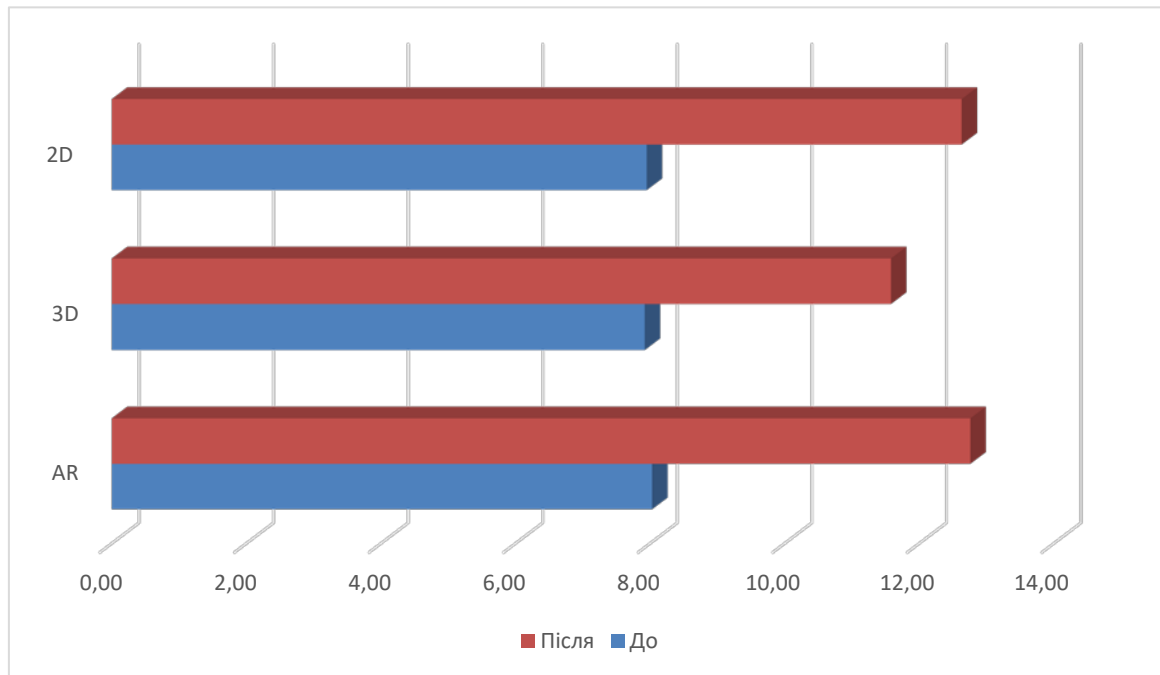


Рис. 3.4. Результат дослідження використання різних підходів до навчання

Це дослідження мало на меті обговорити вплив різних підходів до навчання (*2D*-анімація, *3D*-моделювання та матеріали *AR*) на ефективність навчання та проаналізувати досвід студентів під час використання навчальних матеріалів *AR*. Під час цього дослідження було проведено два експериментальні уроки по 70 хвилин. Відповідно до даних, зібраних з попереднього і післятестового випробувань, були отримані наступні висновки.

По-перше, з точки зору впливу різного підходу до навчального проектування на ефективність навчання, *2D*-анімація, *3D*-моделювання та *AR*-моделювання корисні для поліпшення навчання. Однак відмінності між трьома групами не досягли значного рівня. Це означає, що матеріали в доповненій реальності, розроблені в цьому дослідженні, можуть покращити ефективність навчання, але переваги матеріалів *AR* вимагають подальших досліджень. Пропозиція для майбутніх дослідників полягає у

повному розумінні та достовірному відображенні вмісту знань під час виготовлення матеріалів AR. Що ще важливіше, слід забезпечити повний інтерактивний режим роботи, який може покращити навчальні інтереси студентів [58].

### **Висновки за розділом**

Під час виконання даної роботи було проведено дослідження ефективності використання програмного забезпечення із використанням технології доповненої реальності у навчальному процесі. Більшість досліджень показують, що використання технології доповненої реальності мають такі переваги, як підвищення інтересу, мотивації, захвату студентів у навчанні, полегшення навчання. Також важливо зазначити, що технологія доповненої реальності є дуже ефективна у використанні під час навчання абстрактних понять та наук, особливо, коли раніше завжди все вчилось із тексту та двовимірних зображень. Перевага тривимірних зображень та технології, що дозволяє перенести віртуальні зображення, об'єкти у реальний світ очевидна.

Доповнена реальність сприяє розвитку просторового розуміння, активності. Також дослідження показали, що із використанням даної технології значно збагачується досвід взаємодії і підвищується рівень залучення, адже це, зазвичай, щось нове, небувале та цікаве. Хоч деякі дослідження вказують на те, що багато студентів вже чули і ознайомлені із обговорюваною технологією, проте лише менша частина їх справді мала з нею взаємодію, незважаючи на величезний рівень зацікавлення, як у взаємодії, так і в розробці таких продуктів із використанням технології доповненої реальності.

Отже, інтерес до технології присутній і це означає, що її потрібно більше інтегрувати в наше повсякденне життя, а особливо для вивчення абстрактних наук, для кращого розуміння предмету. Важливо розповсюджувати нові технології, як і для того, щоб знайти нових користувачів, так і для того, щоб створювати нових розробників для цієї галузі.

## ВИСНОВКИ

Технологічний розвиток за останні кілька років досяг рівня, коли неможливо уникнути натрапляння на візуальні трюки *3D*, створені та відображені комп'ютерами. Будь то стереоскопічний (чи інший стерео) показ у кінотеатрі чи комп'ютерна графіка в режимі реального часу, яка постійно розвивається, форма цих віртуальних світів стала частиною нашого повсякденного життя. Технологічний розвиток (будь то з апаратного чи програмного секторів) дає можливість кожному отримати доступ до *3D*-вмісту у доступних формах. Для широкого загалу доступний цілий ряд доцільних програм, але їх обробка та використання далеко не тривіальні.

Новою галуззю інформатики, що динамічно розвивається, є технологія доповненої реальності, яка, окрім поточних програм, в майбутньому відіграватиме значну роль у сферах реклами та маркетингу, освіти, розподіленої праці, охорони здоров'я та програм для проведення вільного часу. Напрямок розвитку технологій *AR* - це виведення мобільних платформ на відоме місце. Додатки, що мають насамперед описові цілі, стають надзвичайно інтерактивними, а ефективні мультимедійні пристрої з'являються у багатьох сферах життя завдяки розвитку внутрішньої підтримки на основі штучного інтелекту.

Під час виконання даної дипломної роботи виконано детальний аналіз технології доповненої реальності, розібрано аспекти розробки *AR*, а саме типи, пристрої, на яких можна відтворювати контент у форматі доповненої реальності, сфери застосування. Також було проведено аналіз різних наборів готових рішень (*SDK*), за допомогою яких можна легко додати в свій проект трішки доповненої реальності. Для виконання даної дипломної роботи було обрано сервіс від *MAXST*. Даний сервіс є простим та зручним у використанні, містить добре оформлені рекомендації для розробки зображень-тригерів, що було важливо, тому, що розроблюваний додаток містить у собі маркерну технологію *AR*. Також в даній пояснювальній записці можна знайти вичерпну інструкцію, що необхідна для початку роботи із *MAXST*. Це важливо, адже розроблений застосунок можна допрацьовувати, збільшувати базу довідкової та інтерактивної інформації. Також під час виконання

даної дипломної роботи було проведено дослідження та втілено в життя процеси розробки програмного забезпечення. Вагомим етапом перед безпосереднім процесом написання коду, є розробка користувацького інтерфейсу, яка була виконана відповідно до стандартних рекомендацій для розробки інтерфейсу мобільних додатків та розробки інтерфейсу для використання в доповненій реальності. Створений користувацький інтерфейс є лаконічним, зрозумілим, має хороше кольорове рішення, що дозволяє візуально ділити інформацію, також інтерфейс є зручним та орієнтованим на кінцевого користувача, що є найважливішим під час розробки інтерфейсів.

Проведено аналіз існуючих платформ для розробки програм та було відібрано найкращий інструмент, ігровий рушій для даного проекту, а саме *Unity 3D Engine*, що дозволяє швидко почати співпрацювати з ним та містить величезну кількість вичерпної інформації, що необхідна під час розробки проектів.

Результатом даної дипломної роботи є працюючий застосунок *AR Довідник*, що дозволяє отримати базову інформацію з предмету Мікроелектроніка у текстовому форматі та з можливістю перегляду *3D* моделей дискретних елементів у режимі маркерної доповненої реальності. Розроблені маркери (зображення-тригери) рекомендується вкладати у друковані навчальні матеріали, щоб ними могли користуватись студенти.

Даний додаток можна допрацьовувати, а саме, збільшувати базу моделей та і взагалі інформації, яку можна там переглянути. Як кажуть, нема межі досконалості, так і тут.

Також в рамках виконання даної дипломної роботи було проведено дослідження, що базується на різних наукових статтях з усього світу на предмет ефективності використання технології доповненої реальності під час навчального процесу. Результатами дослідження є визначення основних переваг, що надає використання *AR* технології в навчальних цілях, а саме, найчастіше прослідковувалось збільшення зацікавлення, уваги, інтересу, запам'ятовування інформації, інтерактиву.

Також важливо зазначити, що технологія доповненої реальності є дуже ефективною для використання її для навчання абстрактних наук. За допомогою доповненої реальності у студентів краще розвивається просторове уявлення.

Отже, інтерес до даної технології присутній, проте вона ще не достатньо увійшла у життя студентів. Дуже важливо саме в навчальних закладах використовувати технологію доповненої реальності як для збільшення кількості користувачів нових технологій, так і для створення нових розробників саме для такого інтерактивного контенту, адже дана технологія має величезну кількість сфер застосування, наприклад, для воєнних проблем, чи медицини, так і є успішним інструментом для використання на лабораторних роботах, особливо в умовах дистанційного навчання. Також вважаю дуже цю технологію дуже корисною, особливо для людей із різними вродженими вадами. Саме завдяки даній технології вони зможуть отримати якіснішу освіту.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Genetic epistemology* [Інтернет-ресурс] Режим доступу <https://enacademic.com/dic.nsf/enwiki/7742991> вільний
2. Brooks, Frederick P. Jr. The Computer Scientist as Toolsmith II. *CACM* 39, 3 (March 1996), 61-68.
3. Azuma, R., Gary, B. A frequency-domain analysis of head-motion prediction. Proceedings of SIGGRAPH '95 (Los Angeles, CA, 6-11 August 1995). In *Computer Graphics, Annual Conference Series*, 1995, pp. 401-408
4. Azuma, R.T., Baillet, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B. Recent advances in augmented reality. *IEEE Computer Graphics & Applications*, 21:6, 34-47, 2001.
5. *DEPTH CAMERAS FOR MOBILE AR* [Інтернет-ресурс] Web-сайт *Medium* Режим доступу <https://medium.com/@DAQRI/depth-cameras-for-mobile-ar-from-iphones-to-wearables-and-beyond-ea29758ec280> вільний
6. *Augmented reality issues* [Інтернет-ресурс] Режим доступу <https://theappsolutions.com/blog/development/augmented-reality-challenges/> вільний
7. *Software* [Інтернет-ресурс] Web-сайт *techopedia* Режим доступу <https://www.techopedia.com/definition/4356/software> вільний
8. *Embedded software* [Інтернет-ресурс] Web-сайт *Wikipedia* Режим доступу [https://en.wikipedia.org/wiki/Embedded\\_software](https://en.wikipedia.org/wiki/Embedded_software) вільний
9. IBM Software development [Інтернет-ресурс] Web-сайт IBM Режим доступу <https://www.ibm.com/topics/software-development> вільний
10. *Low-Code Development Platforms For AD&D Pros* [Інтернет-ресурс] Режим доступу <https://www.forrester.com/report/The+Forrester+Wave+LowCode+Development+Platforms+For+ADD+Pros+Q4+2017/-/E-RES137262> вільний
11. *Software development* [Інтернет-ресурс] Web-сайт *IBM* Режим доступу <https://www.ibm.com/topics/software-development> вільний

12. Офіційний сайт *Unity* [Інтернет-ресурс] Режим доступу <https://unity.com/> вільний
13. Офіційний сайт *Unreal Engine* [Інтернет-ресурс] Режим доступу <https://www.unrealengine.com/> вільний
14. Офіційний сайт розробників *Apple* [Інтернет-ресурс] Режим доступу <https://developer.apple.com/augmented-reality/> вільний
15. Офіційний сайт розробників *Google* [Інтернет-ресурс] Режим доступу <https://developers.google.com/ar> вільний
16. Офіційний сайт *AR SDK EasyAR* [Інтернет-ресурс] Режим доступу <https://www.easyar.com/> вільний
17. Офіційний сайт *AR SDK Vuforia* [Інтернет-ресурс] Режим доступу <https://developer.vuforia.com/> вільний
18. Офіційний сайт девелоперів *MAXST* [Інтернет-ресурс] Режим доступу <https://developer.maxst.com/MD/doc/g/tacondition> вільний
19. Офіційний сайт *AR SDK Wikitude* [Інтернет-ресурс] Режим доступу <https://www.wikitude.com/> вільний
20. *12 Best Augmented Reality SDKs* [Інтернет-ресурс] Режим доступу <https://dzone.com/articles/12-best-augmented-reality-sdks> вільний
21. Дискретний елемент *Web-сайт Wikipedia* Режим доступу [https://uk.wikipedia.org/wiki/Дискретний\\_елемент](https://uk.wikipedia.org/wiki/Дискретний_елемент) вільний
22. Активний елемент *Web-сайт Wikipedia* Режим доступу [https://uk.wikipedia.org/wiki/Активний\\_елемент\\_\(електроніка\)](https://uk.wikipedia.org/wiki/Активний_елемент_(електроніка)) вільний
23. Пасивний елемент *Web-сайт Wikipedia* Режим доступу [https://uk.wikipedia.org/wiki/Пасивний\\_елемент\\_\(електроніка\)](https://uk.wikipedia.org/wiki/Пасивний_елемент_(електроніка)) вільний
24. *How to come up with the right name for your app* [Інтернет-ресурс] Режим доступу <https://www.thinklions.com/blog/how-to-name-an-app/> вільний
25. *User Interface Design Basics* [Інтернет-ресурс] Режим доступу <https://www.usability.gov/what-and-why/user-interface-design.html> вільний
26. *User Interface Design Fundamentals* [Інтернет-ресурс] Режим доступу <https://blog.teamtreehouse.com/10-user-interface-design-fundamentals> вільний



27. Seferoglu, S. S. (2009). *Ilkogretim okullarında teknoloji kullanımı ve yoneticilerin bakis acilari. Akademik Bilişim '09* (с. 403–410). Şanlıurfa, Turkey.
28. Sumadio, D. D. & Rambli, D. R. A. (2010). *Preliminary evaluation on user acceptance of the augmented reality use for education. Proceedings of Second International Conference on Computer Engineering and Applications*  
[Інтернет-ресурс] Режим доступу  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=544569> вільний
29. Cai, S., Chiang, F. K., Sun, Y., Lin, C., & Lee, J. J. (2017). *Applications of augmented reality-based natural interactive learning in magnetic field instruction. Interactive Learning Environments*, 25(6), 778-791.
30. Delello, J. A. (2014). *Insights from pre-service teachers using science-based augmented reality. Journal of Computers in Education*, 1(4), 295-311.
31. Perez-Lopez, D., & Contero, M. (2013). *Delivering educational multimedia contents through an augmented reality application: A case study on its impact on knowledge acquisition and retention. Turkish Online Journal of Educational Technology*, 12(4), 19-28.
32. Tomi, A. & Rambli, D. R. A. (2013). *An interactive mobile augmented reality magical playbook: Learning number with the thirsty crow. Procedia Computer Science*, 25, 123-130.
33. Yen, J.-C., Tsai, C.-H., & Wu, M. (2013). *Augmented reality in the higher education: Students' science concept learning and academic achievement in astronomy. Procedia - Social and Behavioral Sciences*, 103, 165-173.
34. Kerawalla, L., Luckin, R., Seljeflot, S., & Woolard, A. (2006). *Making it real: exploring the potential of augmented reality for teaching primary school science. Virtual Reality*, 10(3-4), 163-174.
35. Sotiriou, S. & Bogner, F. X. (2008). *Visualizing the Invisible: augmented reality as an innovative science education scheme. Journal of Computational and Theoretical Nanoscience*, 1(1), 114-122.
36. Yilmaz, R. M. (2016). *Educational magic toys developed with augmented reality technology for early childhood education. Computers in Human Behavior*, 54, 240-248.

37. Zarzuela, M. M., Pernas, F. J. D., Martínez, L. B., Ortega, D. G., & Rodríguez, M. A. (2013). *Mobile serious game using augmented reality for supporting children's learning about animals*. *Procedia Computer Science*, 25, 375-381.
38. Akcayir, M., Akcayir, G., Pektas, H. M., & Ocak, M. A. (2016). *Augmented reality in science laboratories: The effects of augmented reality on university students' laboratory skills and attitudes toward science laboratories*. *Computers in Human Behavior*, 57, 334-342.
39. Ivanova, M. & Ivanov, G. (2011). *Enhancement of learning and teaching in computer graphics through marker augmented reality technology*. *International Journal of New Computer Architectures and Their Applications (IJNCAA)*, 1(1), 176-184.
40. Rosenbaum, E., Klopfer, E., & Perry, J. (2006). *On location learning: Authentic applied science with networked augmented realities*. *Journal of Science Education and Technology*, 16(1), 31-45.
41. Yoon, S., Anderson, E., Lin, J., & Elinich, K. (2017). *How augmented reality enables conceptual understanding of challenging science content*. *Educational Technology & Society*, 20 (1), 156-168.
42. Cai, S., Wang, X., & Chiang, F.-K. (2014). *A case study of augmented reality simulation system application in a chemistry course*. *Computers in Human Behavior*, 37, 31-40. [Интернет-ресурс] Режим доступа <http://doi.org/http://dx.doi.org/10.1016/j.chb.2014.04.018> вільний
43. Shelton, B. E. & Hedley, N. R. (2002). *Using augmented reality for teaching earth-sun relationships to undergraduate geography students: Augmented reality toolkit*. *The First IEEE International Workshop* (ст. 8). Darmstadt, Germany.
44. Shelton, B. E. & Stevens, R. (2004). *Using coordination classes to interpret conceptual change in astronomical thinking*. *Proceedings of the 6th international conference for the learning sciences*. Mahweh, NJ: Lawrence Erlbaum & Associates.
45. Bujak, K. R., Radu, I., Catrambone, R., MacIntyre, B., Zheng, R., & Golubski, G. (2013). *A psychological perspective on augmented reality in the mathematics classroom*. *Computers & Education*, 68, 536-544.

46. Cheng, K.-H. & Tsai, C.-C. (2013). *Affordances of augmented reality in science learning: Suggestions for future research*. *Journal of Science Education and Technology*, 22(4), 449-462.
47. Medicherla, P. S., Chang, G., & Morreale, P. (2010). *Visualization for increased understanding and learning using augmented reality*. *Proceedings of the International Conference on Multimedia Information Retrieval* [Інтернет-ресурс] Режим доступу <http://doi.org/10.1145/> вільний
48. Wojciechowski, R. & Cellary, W. (2013). *Evaluation of learners' attitude toward learning in ARIES augmented reality environments*. *Computers & Education*, 68, 570-585.
49. Abdusselam, M. S. & Karal, H. (2012). *The effect of mixed reality environments on the students' academic achievement in physics education: 11th grade magnetism topic example*. *Journal of Research in Education and Teaching*, 1(4), 170-181.
50. Cai, S., Chiang, F.-K., & Wang, X. (2013). *Using the augmented reality 3D technique for a convex imaging experiment in a physics course*. *International Journal of Engineering Education*, 29(4), 856-865.
51. Dunleavy, M., Dede, C., & Mitchell, R. (2009). *Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning*. *Journal of Science Education and Technology*, 18(1), 7-22.
52. Yusoff, Z. & Dahlan, H. M. (2013). *Mobile based learning: An integrated framework to support learning engagement through augmented reality environment*. *Research and Innovation in Information Systems (ICRIIS)*, Kuala Lumpur, Malaysia: IEEE.
53. Bacca, J., Baldiris, S., Fabregat, R., Graf, S., & Kinshuk. (2014). *Augmented Reality Trends in Education: A Systematic Review of Research and Applications*. *Educational Technology & Society*, 17 (4), 133–149.
54. Utku Kose, Durmus Koc, Suleyman Anil Yucesoy (2013) *An Augmented Reality Based Mobile Software to Support Learning Experiences in Computer Science Courses*. *Procedia Computer Science* 25 ( 2013 ) 370 – 374

55. *Liarokapis F. Augmented Reality Interfaces for Assisting Computer Games University Students. Bulletin of the IEEE Technical Committee on Learning Technology 2012*

56. *Ilana de A. Souza-Concilio, Beatriz A. Pacheco The Development of Augmented Reality Systems in Informatics Higher Education 2013 International Conference on Virtual and Augmented Reality in Education*

57. *Tasneem Khan, Kevin Johnston, and Jacques Ophoff The Impact of an Augmented Reality Application on Learning Motivation of Students*

[Стаття-дослідження]                      Режим                      доступу                      ВІЛЬНИЙ

<https://www.hindawi.com/journals/ahci/2019/7208494/>

58. *Jung-Chuan Yena , Chih-Hsiao Tsaib , Min Wua Augmented reality in the higher education: Students' science concept learning and academic achievement Procedia - Social and Behavioral Sciences 103 ( 2013 ) 165 – 173*

# ДОДАТКИ

## Додаток А



Рис. А.1 Макет типової інформаційної сторінки із доступом до AR

## Додаток Б



Рис. Б.1 Зображення-тригер для діодів



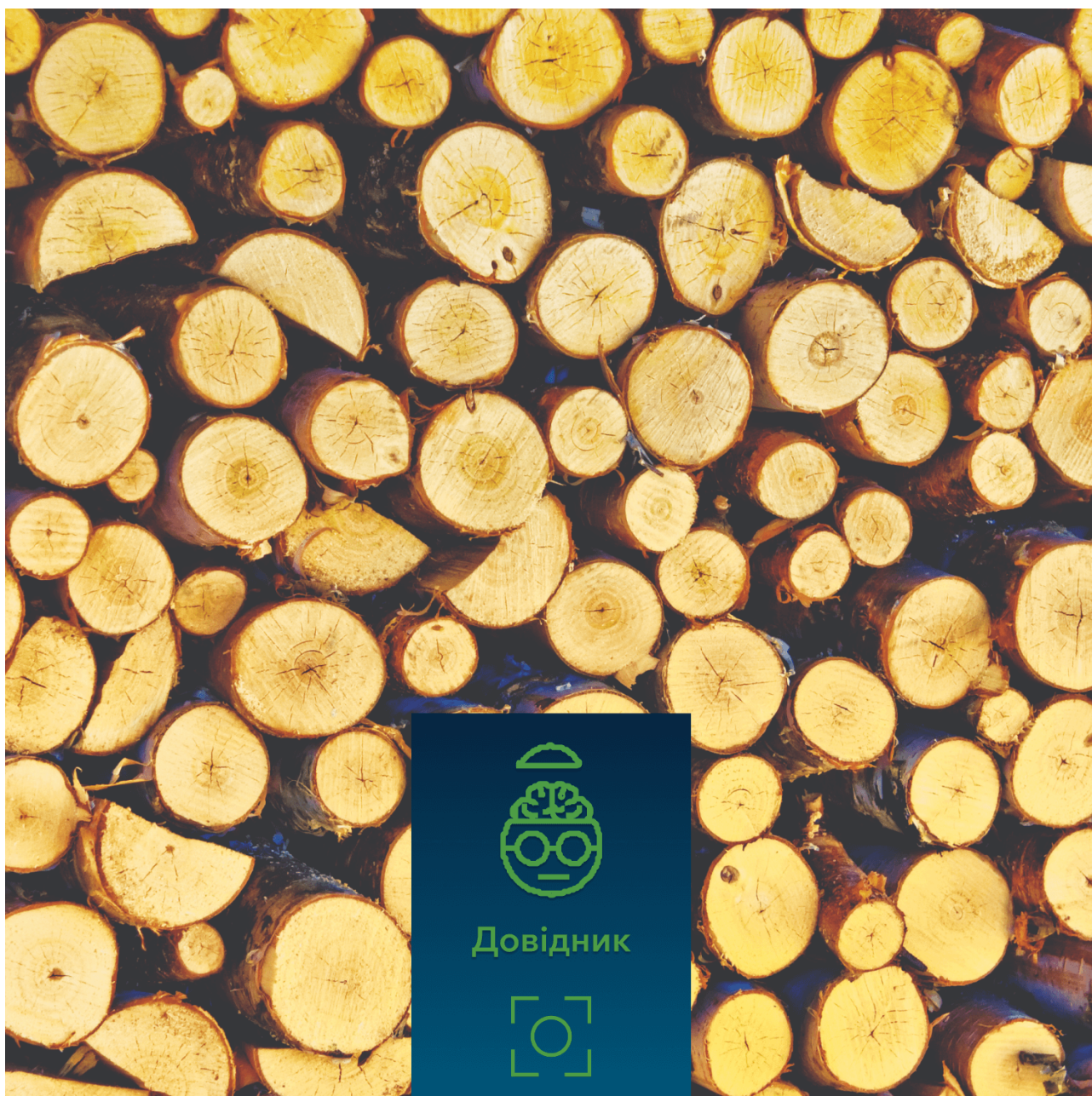


Рис. Б.2 Зображення-тригер для транзисторів



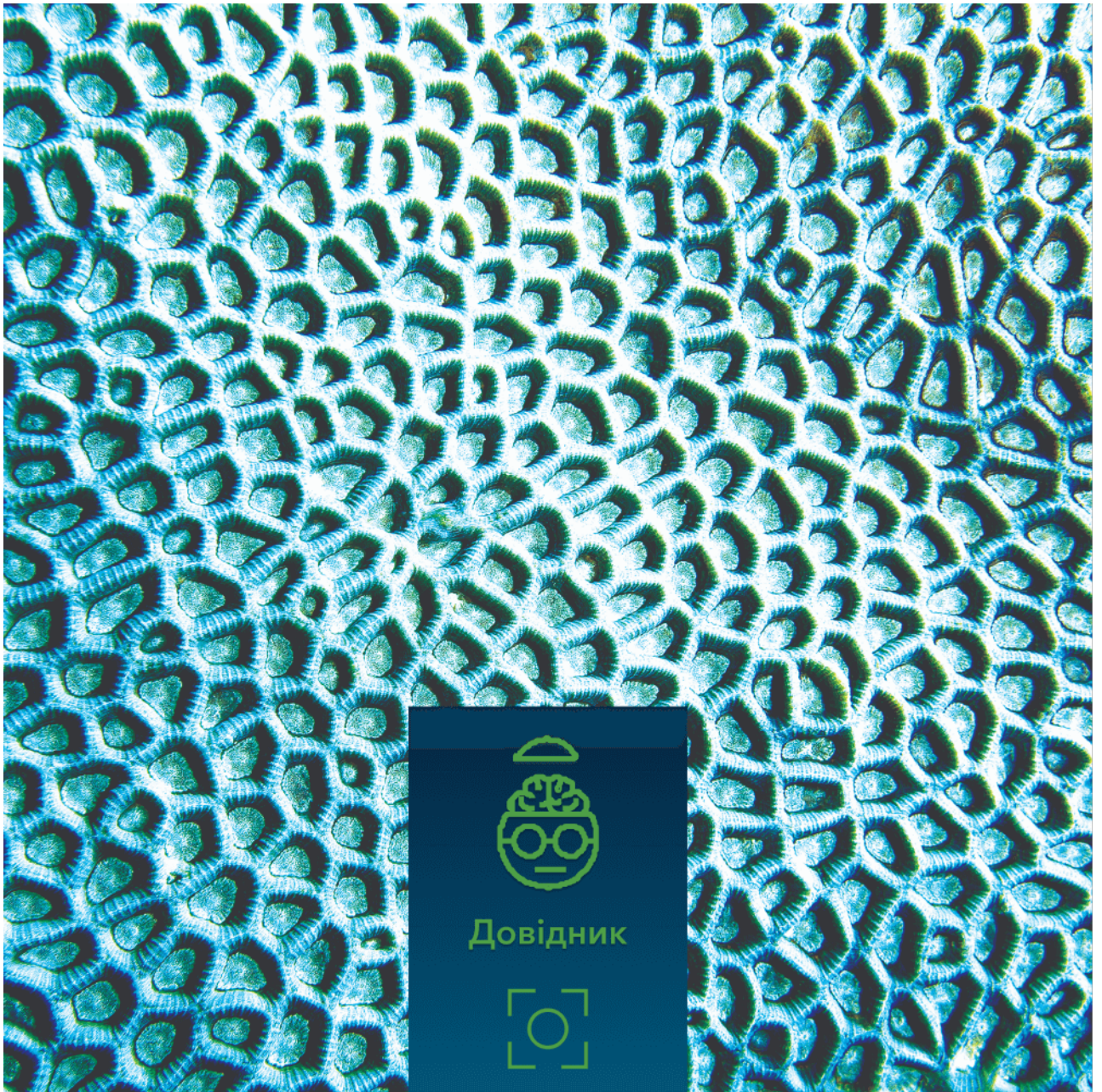


Рис. Б.3 Зображення-тригер для тиристорів





Рис. Б.4 Зображення-тригер для підсилювачів



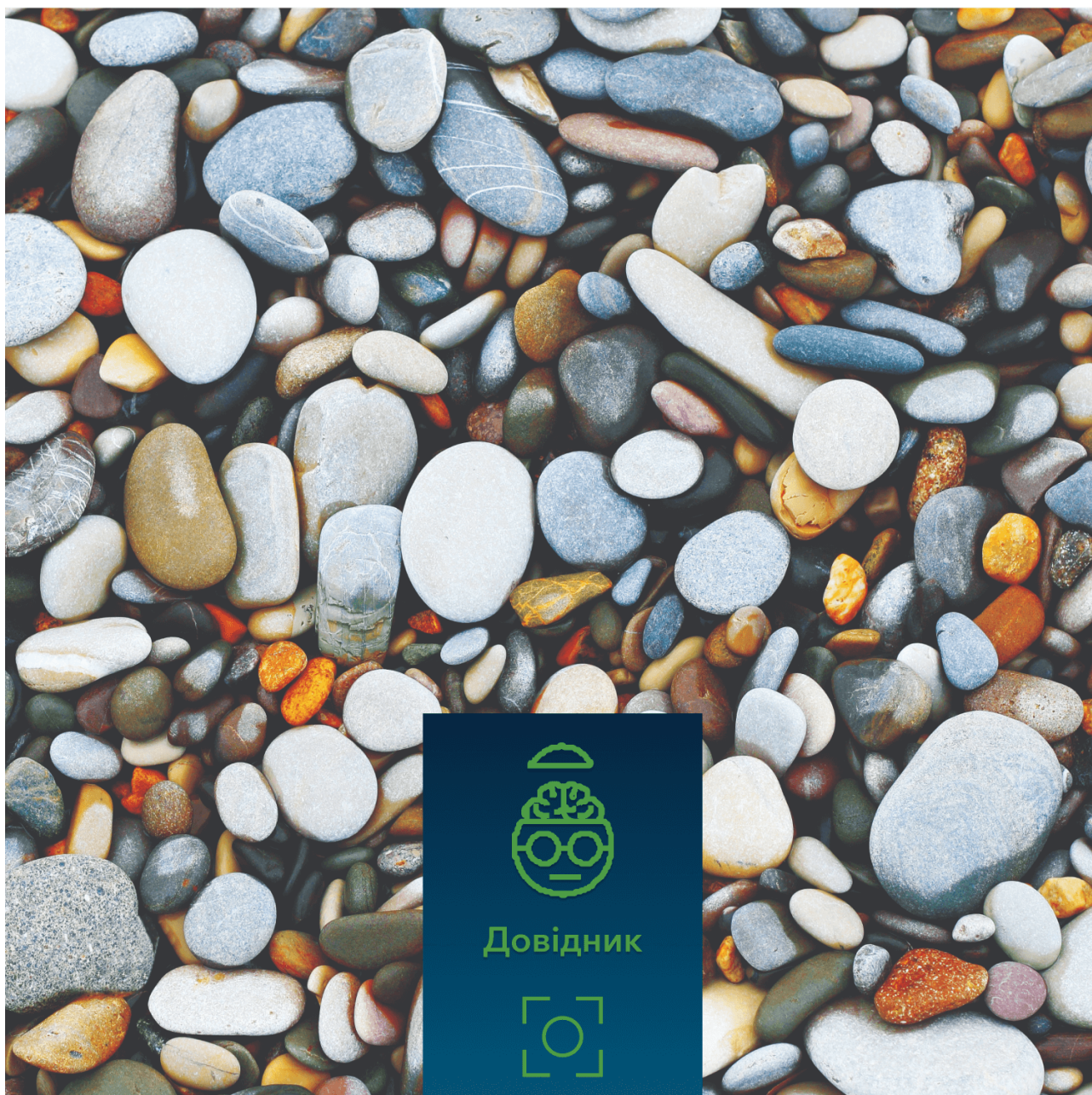


Рис. Б.5 Зображення-тригер для конденсаторів





Рис. Б.6 Зображення-тригер для резисторів





Рис. Б.7 Зображення-тригер для катушок індуктивності



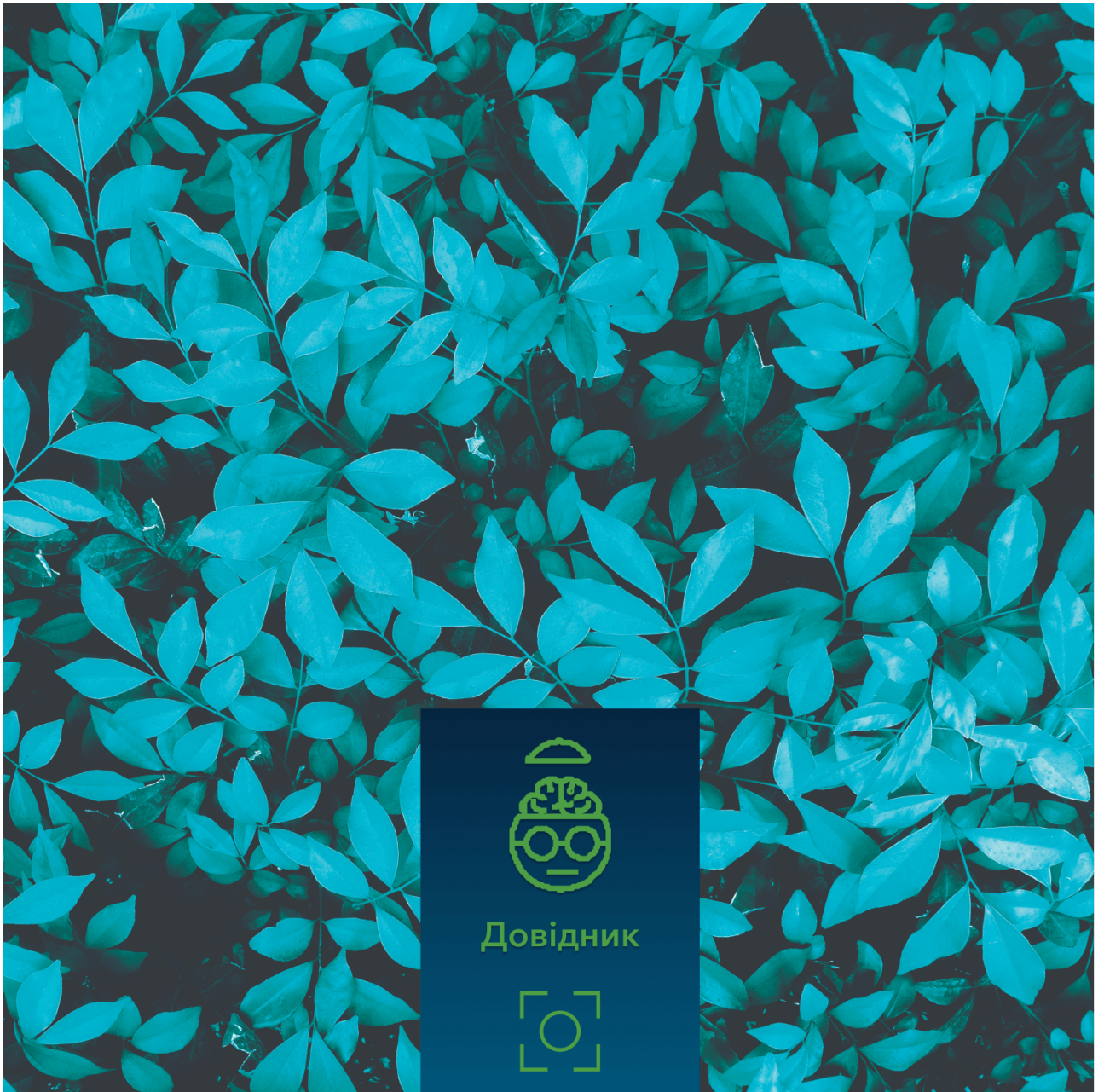


Рис. Б.8 Зображення-тригер для реле

## Додаток В

Завантажені 3D файли моделей дискретних елементів згідно до структури (рис 2.1)

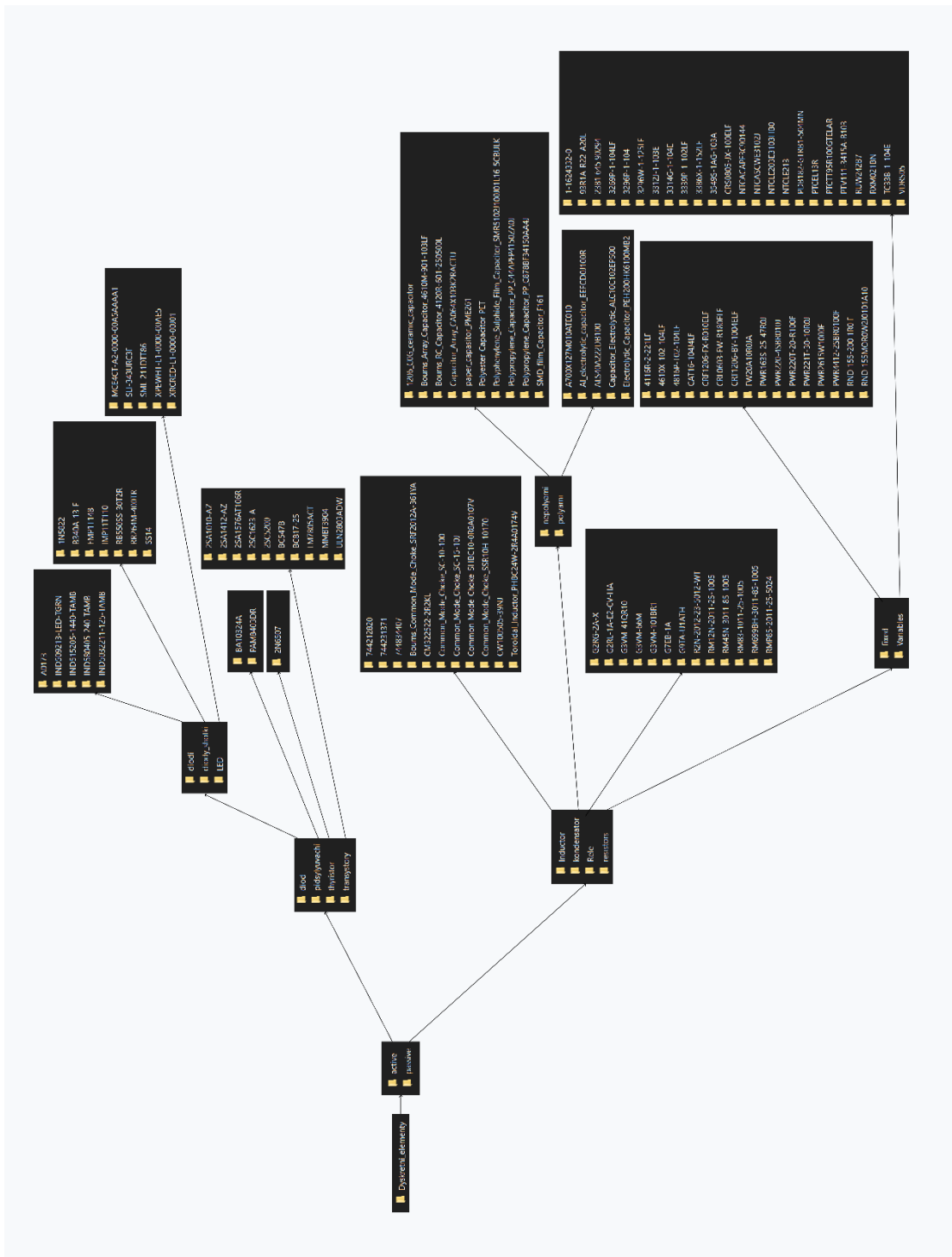


Рис. В.1 Структурна схема завантажених файлів 3D

## Додаток Г

LevelLoadManager.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

namespace Managers
{
    public class LevelLoadManager : Additional.Singleton<LevelLoadManager>
    {
        #region Events
        public delegate IEnumerator LoadEventHandler();
        /// <summary>
        /// Add new delegate in Awake if u want call it before close loading canvas
        /// </summary>
        public event LoadEventHandler LoadCompleteEvent;
        public event LoadEventHandler LoadStartedEvent;
        public IEnumerator OnLoadCompleteEvent()
        {
            if (LoadCompleteEvent != null)
            {
                //for (int i = LoadCompleteEvent.GetInvocationList().Length - 1; i >= 0; i--)
                //{
                //    LoadEventHandler handler = LoadCompleteEvent.GetInvocationList()[i] as
                LoadEventHandler;
                //    yield return handler.Invoke();
                //}
                //For 1 to N, for first added to Last added
                foreach (LoadEventHandler handler in LoadCompleteEvent.GetInvocationList())
                {
                    yield return handler.Invoke();
                }
            }
        }
        public IEnumerator OnLoadStartedEvent()
        {
            if (LoadStartedEvent != null)
            {
                foreach (LoadEventHandler handler in LoadStartedEvent.GetInvocationList())
                {
                    yield return handler.Invoke();
                }
            }
        }
        #endregion
        [SerializeField] private Canvas LoadingScreenCanvas;
        [SerializeField] private Image LoadingFillImage;
        [SerializeField] private Camera LevelLoadCamera;

        private bool disableCanvasGo = true;
        private bool isStarted = false;
        private bool isEnded = true;
        public bool IsLoading { get; set; } = false;

        protected override void Awake()
        {
```

```

    base.Awake();
    if(LoadingScreenCanvas != null)
        LoadingScreenCanvas.enabled = false;
    if(disableCanvasGo && LoadingScreenCanvas != null)
        LoadingScreenCanvas.gameObject.SetActive(true);
    if (LevelLoadCamera != null)
        LevelLoadCamera.gameObject.SetActive(false);
    isStarted = false;
    isEnded = true;
    LoadStartedEvent += EnableLoadingScreen;
}
protected void Start()
{
    LoadCompleteEvent += DisableLoadingScreen;
}
public IEnumerator LoadSceneAsync(int sceneBuildIndex, float timeToWait = 0)
{
    yield return LoadSceneRoutine(sceneBuildIndex, timeToWait);
}
/// <summary>
/// Load scene by scene index after timeToWait seconds
/// </summary>
/// <param name="sceneBuildIndex"></param>
/// <param name="timeToWait"></param>
public void LoadScene(int sceneBuildIndex, float timeToWait, int unloadSceneIndex = -1)
{
    StartCoroutine(LoadSceneRoutine(sceneBuildIndex, timeToWait, unloadSceneIndex));
}

/// <summary>
/// Loading Scene by scene index
/// </summary>
/// <param name="sceneBuildIndex">Loading scene index</param>
/// <param name="withPreLoad">if true, Loading PreLoadScene, then Loading scene by
sceneBuildIndex </param>
/// <param name="gameObjectName">GameObject's name, on Position which Player will be set
</param>
public void LoadScene(int sceneBuildIndex)
{
    StartCoroutine(LoadSceneRoutine(sceneBuildIndex));
}
private IEnumerator UnLoadSceneAsyncRoutine(int sceneBuildIndex)
{
    AsyncOperation async =
UnityEngine.SceneManagement.SceneManager.UnLoadSceneAsync(sceneBuildIndex);
    if (async != null)
    {
        while (!async.isDone)
        {
            yield return null;
        }
    }
}
private IEnumerator LoadSceneRoutine(int sceneBuildIndex, float timeToWait = 0, int
unloadSceneIndex = -1)
{
    if (timeToWait > 0)
        yield return new WaitForSeconds(timeToWait);
    //set variable to Load
    IsLoading = true;
    //wait for event to complete
    if(!isStarted)
        yield return OnLoadStartedEvent();
    //if unload first

```



```

        if(unLoadSceneIndex > 0)
        {
            yield return UnLoadSceneAsyncRoutine(unLoadSceneIndex);
        }
        //Loading scene
        AsyncOperation async =
UnityEngine.SceneManagement.SceneManager.LoadSceneAsync(sceneBuildIndex);
        while (!async.isDone)
        {
            float progress = async.progress / .9f;
            if (LoadingFillImage != null)
                LoadingFillImage.fillAmount = progress;
            yield return null;
        }
        //Wait for end of frame
        yield return null;
        //Temp wait
        yield return new WaitForSeconds(1);
        //Wait for event to complete
        if(!isEnded)
            yield return OnLoadCompleteEvent();
        IsLoading = false;
    }

    /// <summary>
    /// Enable all GO for Loading Screen
    /// </summary>
    private IEnumerator EnableLoadingScreen()
    {
        //
        isStarted = true;
        isEnded = false;
        //
        if (LoadingScreenCanvas != null)
            LoadingScreenCanvas.enabled = true;
        if (disableCanvasGo && LoadingScreenCanvas != null)
            LoadingScreenCanvas.gameObject.SetActive(true);
        if (LoadingFillImage != null)
            LoadingFillImage.fillAmount = 0;
        if(Camera.main == null)
        {
            LevelLoadCamera.gameObject.SetActive(true);
        }
        Debug.Log("Enable Load screen,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,");
        yield return null;
    }

    /// <summary>
    /// Disabling all GO for LS
    /// </summary>
    private IEnumerator DisableLoadingScreen()
    {
        //
        isStarted = false;
        isEnded = true;
        //
        if (LoadingScreenCanvas != null)
            LoadingScreenCanvas.enabled = false;
        if (disableCanvasGo && LoadingScreenCanvas != null)
            LoadingScreenCanvas.gameObject.SetActive(false);
        if (LevelLoadCamera.gameObject.activeSelf)
        {
            LevelLoadCamera.gameObject.SetActive(false);
        }
    }

```

```

        Debug.Log("Canvas disabled");
        yield return null;
    }
}
}

```

ApplicationManager.cs

```

using AR_Camera_Scene.Model_Viewer;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
namespace Managers
{
    public class ApplicationManager : Additional.Singleton<ApplicationManager>
    {
        //Temp Variables for add components to another scene
        //Game Object that will be added to AR Scene (Image Tracker Behaviour)
        private GameObject tempGo = null;
        //List of models list that will be initialized in AR Scene.ModelViewer
        private List<ModelsList> tempLists = null;

        private void Start()
        {
            LoadMainScene();
        }
        /// <summary>
        /// Loading Main Scene
        /// </summary>
        public void LoadMainScene()
        {
            //Loading Main Menu Scene. (Index in .LoadScene() - Index of scene in Build Settings
            (Ctrl + Shift + B)
            LevelLoadManager.Instance.LoadScene(1);
        }
        /// <summary>
        /// Loading AR Scene
        /// </summary>
        /// <param name="imageTrackable"></param>
        /// <param name="modelsLists"></param>
        public void LoadARScene(GameObject imageTrackable, List<ModelsList> modelsLists)
        {
            //Set temp go for image Trackable. Or Copy it from Object to Object Variable.
            tempGo = imageTrackable;
            //Set temp Lists
            tempLists = modelsLists;
            //Add to Load Complete Event function that add Temp Game Object to scene and init
            Model Viewer with list of models
            LevelLoadManager.Instance.LoadCompleteEvent += AddGameObjectToScene;
            //Loading AR Scene
            LevelLoadManager.Instance.LoadScene(2);
        }

        private IEnumerator AddGameObjectToScene()
        {
            //Instantiate Game Object = From Variable to Real Game Object in Scene
            GameObject gameObject = Instantiate(tempGo);

            //Set Parent for Game Object that contains models
            ModelViewer.Instance.CurrentModelTransform.SetParent(gameObject.transform);
            //Init Image Tracker after add Image Tracker Behaviour to scene
            AR_Camera_Scene.ImageTrackerSample.Instance.InitializeTrackableMap();
            //Init Model Viewer
        }
    }
}

```

```

    ModelViewer.Instance.ModelsLists = tempLists;
    ModelViewer.Instance.InitializeViewer();
    //Reset
    tempGo = null;
    tempLists = null;
    LevelLoadManager.Instance.LoadCompleteEvent -= AddGameObjectToScene;
    //
    yield return null;
}
/// <summary>
/// Opens Url link
/// </summary>
/// <param name="url"></param>
public void OpenUri(string url)
{
    Application.OpenURL(url);
}
}
}

```

*UniversalAdditionalCameraData.cs*

```

using System;

using System.Collections.Generic;

using System.Linq;

using UnityEditor;

using UnityEngine.Scripting.APIUpdating;

using UnityEngine.Serialization;

using UnityEngine.Rendering;

using System.ComponentModel;

namespace UnityEngine.Rendering.LWRP
{
    [Obsolete("LWRP -> Universal (UnityUpgradable) ->
    UnityEngine.Rendering.Universal.UniversalAdditionalCameraData", true)]

    public class LWRPAdditionalCameraData

    {

    }

}

```

```

namespace UnityEngine.Rendering.Universal
{
    /// <summary>
    /// Holds information about whether to override certain camera rendering options from the
    render pipeline asset.
    /// When set to <c>Off</c> option will be disabled regardless of what is set on the pipeline
    asset.
    /// When set to <c>On</c> option will be enabled regardless of what is set on the pipeline
    asset.
    /// When set to <c>UsePipelineSetting</c> value set in the <see
    cref="UniversalRenderPipelineAsset"/>.
    /// </summary>
    [MovedFrom("UnityEngine.Rendering.LWRP")] public enum CameraOverrideOption
    {
        Off,
        On,
        UsePipelineSettings,
    }

    //[Obsolete("Renderer override is no longer used, renderers are referenced by index on the
    pipeline asset.")]
    [MovedFrom("UnityEngine.Rendering.LWRP")] public enum RendererOverrideOption
    {
        Custom,
        UsePipelineSettings,
    }

    /// <summary>
    /// Holds information about the post-processing anti-aliasing mode.

```

*/// When set to <c>None</c> no post-processing anti-aliasing pass will be performed.*

*/// When set to <c>Fast</c> a fast approximated anti-aliasing pass will render when resolving the camera to screen.*

*/// When set to <c>SubpixelMorphologicalAntiAliasing</c> SMAA pass will render when resolving the camera to screen. You can choose the SMAA quality by setting <seealso cref="AntialiasingQuality"/>*

*/// </summary>*

*public enum AntialiasingMode*

*{*

*None,*

*FastApproximateAntialiasing,*

*SubpixelMorphologicalAntiAliasing,*

*//TemporalAntialiasing*

*}*

*/// <summary>*

*/// Holds information about the render type of a camera. Options are Base or Overlay.*

*/// Base rendering type allows the camera to render to either the screen or to a texture.*

*/// Overlay rendering type allows the camera to render on top of a previous camera output, thus compositing camera results.*

*/// </summary>*

*public enum CameraRenderType*

*{*

*Base,*

*Overlay,*

*}*

*/// <summary>*

*/// Holds information about the output target for a camera.*

```

/// Only used for cameras of render type Base. <seealso cref="CameraRenderType"/>.
/// </summary>
[Obsolete("This enum is deprecated.")]
[EditorBrowsable(EditorBrowsableState.Never)]
public enum CameraOutput
{
    Screen,
    Texture,

    [Obsolete("Use CameraOutput.Screen instead.", false)]
    [EditorBrowsable(EditorBrowsableState.Never)]
    Camera = Screen,
}

/// <summary>
/// Controls SMAA anti-aliasing quality.
/// </summary>
public enum AntialiasingQuality
{
    Low,
    Medium,
    High
}

/// <summary>
/// Contains extension methods for Camera class.
/// </summary>

```

```

public static class CameraExtensions
{
    /// <summary>
    /// Universal Render Pipeline exposes additional rendering data in a separate component.
    /// This method returns the additional data component for the given camera or create one
    if it doesn't exists yet.
    /// </summary>
    /// <param name="camera"></param>
    /// <returns>The <c>UniversalAdditinalCameraData</c> for this camera.</returns>
    /// <see cref="UniversalAdditionalCameraData"/>
    public static UniversalAdditionalCameraData GetUniversalAdditionalCameraData(this Camera
camera)
    {
        var gameObject = camera.gameObject;

        bool componentExists = gameObject.TryGetComponent<UniversalAdditionalCameraData>(out
var cameraData);

        if (!componentExists)
            cameraData = gameObject.AddComponent<UniversalAdditionalCameraData>();

        return cameraData;
    }
}

static class CameraTypeUtility
{
    static string[] s_CameraTypeNames = Enum.GetNames(typeof(CameraRenderType)).ToArray();

    public static string GetName(this CameraRenderType type)

```

```

{
    int typeInt = (int)type;

    if (typeInt < 0 || typeInt >= s_CameraTypeNames.Length)

        typeInt = (int)CameraRenderType.Base;

    return s_CameraTypeNames[typeInt];
}
}

```

```
[DisallowMultipleComponent]
```

```
[RequireComponent(typeof(Camera))]
```

```
[ImageEffectAllowedInSceneView]
```

```
[MovedFrom("UnityEngine.Rendering.LWRP")] public class UniversalAdditionalCameraData :
MonoBehaviour, ISerializationCallbackReceiver
```

```
{
```

```
[Tooltip("If enabled shadows will render for this camera.")]
```

```
[FormerlySerializedAs("renderShadows"), SerializeField]
```

```
bool m_RenderShadows = true;
```

```
[Tooltip("If enabled depth texture will render for this camera bound as
_CameraDepthTexture.")]
```

```
[SerializeField]
```

```
CameraOverrideOption m_RequiresDepthTextureOption =
CameraOverrideOption.UsePipelineSettings;
```

```
[Tooltip("If enabled opaque color texture will render for this camera and bound as
_CameraOpaqueTexture.")]
```

```
[SerializeField]
```

```
CameraOverrideOption m_RequiresOpaqueTextureOption =
CameraOverrideOption.UsePipelineSettings;
```



```

[SerializeField] CameraRenderType m_CameraType = CameraRenderType.Base;

    [SerializeField] List<Camera> m_Cameras = new List<Camera>();

    [SerializeField] int m_RendererIndex = -1;

[SerializeField] LayerMask m_VolumeLayerMask = 1; // "Default"

[SerializeField] Transform m_VolumeTrigger = null;

[SerializeField] bool m_RenderPostProcessing = false;

[SerializeField] AntialiasingMode m_Antialiasing = AntialiasingMode.None;

[SerializeField] AntialiasingQuality m_AntialiasingQuality = AntialiasingQuality.High;

[SerializeField] bool m_StopNaN = false;

[SerializeField] bool m_Dithering = false;

[SerializeField] bool m_ClearDepth = true;

// Deprecated:

[FormerlySerializedAs("requiresDepthTexture"), SerializeField]

bool m_RequiresDepthTexture = false;

[FormerlySerializedAs("requiresColorTexture"), SerializeField]

bool m_RequiresColorTexture = false;

[HideInInspector] [SerializeField] float m_Version = 2;

public float version => m_Version;

static UniversalAdditionalCameraData s_DefaultAdditionalCameraData = null;

```

```

internal static UniversalAdditionalCameraData defaultAdditionalCameraData
{
    get
    {
        if (s_DefaultAdditionalCameraData == null)
            s_DefaultAdditionalCameraData = new UniversalAdditionalCameraData();

        return s_DefaultAdditionalCameraData;
    }
}

/// <summary>
/// Controls if this camera should render shadows.
/// </summary>
public bool renderShadows
{
    get => m_RenderShadows;
    set => m_RenderShadows = value;
}

/// <summary>
/// Controls if a camera should render depth.
/// The depth is available to be bound in shaders as _CameraDepthTexture.
/// <seealso cref="CameraOverrideOption"/>
/// </summary>
public CameraOverrideOption requiresDepthOption
{

```

```
    get => m_RequiresDepthTextureOption;

    set => m_RequiresDepthTextureOption = value;
}
```

```
/// <summary>
```

```
/// Controls if a camera should copy the color contents of a camera after rendering opaques.
```

```
/// The color texture is available to be bound in shaders as _CameraOpaqueTexture.
```

```
/// </summary>
```

```
public CameraOverrideOption requiresColorOption
```

```
{
```

```
    get => m_RequiresOpaqueTextureOption;
```

```
    set => m_RequiresOpaqueTextureOption = value;
```

```
}
```

```
/// <summary>
```

```
/// Returns the camera renderType.
```

```
/// <see cref="CameraRenderType"/>.
```

```
/// </summary>
```

```
public CameraRenderType renderType
```

```
{
```

```
    get => m_CameraType;
```

```
    set => m_CameraType = value;
```

```
}
```

```
#region deprecated
```

```
/// <summary>
```

```
/// Returns the camera output type. Only valid for Base cameras.
```

```
/// <see cref="CameraOutput"/>.
```

```
/// <seealso cref="CameraRenderType"/>.
```

```
/// <seealso cref="Camera"/>
```

```
/// </summary>
```

```
[Obsolete("CameraOutput has been deprecated. Use Camera.targetTexture instead.")]
```

```
[EditorBrowsable(EditorBrowsableState.Never)]
```

```
public CameraOutput cameraOutput
```

```
{
```

```
    get
```

```
    {
```

```
        gameObject.TryGetComponent<Camera>(out var camera);
```

```
        if (camera?.targetTexture == null)
```

```
            return CameraOutput.Screen;
```

```
        return CameraOutput.Texture;
```

```
    }
```

```
    set { }
```

```
}
```

```
[Obsolete("AddCamera has been deprecated. You can add cameras to the stack by calling  
<c>cameraStack</c> property and modifying the camera stack list.")]
```

```
[EditorBrowsable(EditorBrowsableState.Never)]
```

```
public void AddCamera(Camera camera)
```

```
{
```

```
    m_Cameras.Add(camera);
```

```
}
```

```
[Obsolete("cameras property has been deprecated. Use cameraStack property instead.")]
```

```

[EditorBrowsable(EditorBrowsableState.Never)]

public List<Camera> cameras => cameraStack;

#endregion

/// <summary>
/// Returns the camera stack. Only valid for Base cameras.
/// Overlay cameras have no stack and will return null.
/// <seealso cref="CameraRenderType"/>.
/// </summary>

public List<Camera> cameraStack
{
    get
    {
        if (renderType != CameraRenderType.Base)
        {
            var camera = gameObject.GetComponent<Camera>();

            Debug.LogWarning(string.Format("{0}: This camera is of {1} type. Only Base
cameras can have a camera stack.", camera.name, renderType));

            return null;
        }

        if (scriptableRenderer.supportedRenderingFeatures.cameraStacking == false)
        {
            var camera = gameObject.GetComponent<Camera>();

            Debug.LogWarning(string.Format("{0}: This camera has a ScriptableRenderer
that doesn't support camera stacking. Camera stack is null.", camera.name));

            return null;
        }
    }
}

```

```

        return m_Cameras;
    }
}

/// <summary>
/// If true, this camera will clear depth value before rendering. Only valid for Overlay
cameras.
/// </summary>
public bool clearDepth
{
    get => m_ClearDepth;
}

/// <summary>
/// Returns true if this camera needs to render depth information in a texture.
/// If enabled, depth texture is available to be bound and read from shaders as
_CameraDepthTexture after rendering skybox.
/// </summary>
public bool requiresDepthTexture
{
    get
    {
        if (m_RequiresDepthTextureOption == CameraOverrideOption.UsePipelineSettings)
        {
            return UniversalRenderPipeline.asset.supportsCameraDepthTexture;
        }
        else

```

```

        {
            return m_RequiresDepthTextureOption == CameraOverrideOption.On;
        }
    }

    set { m_RequiresDepthTextureOption = (value) ? CameraOverrideOption.On :
CameraOverrideOption.Off; }
}

/// <summary>
/// Returns true if this camera requires to color information in a texture.
/// If enabled, color texture is available to be bound and read from shaders as
_CameraOpaqueTexture after rendering skybox.
/// </summary>

public bool requiresColorTexture
{
    get
    {
        if (m_RequiresOpaqueTextureOption == CameraOverrideOption.UsePipelineSettings)
        {
            return UniversalRenderPipeline.asset.supportsCameraOpaqueTexture;
        }
        else
        {
            return m_RequiresOpaqueTextureOption == CameraOverrideOption.On;
        }
    }

    set { m_RequiresOpaqueTextureOption = (value) ? CameraOverrideOption.On :
CameraOverrideOption.Off; }
}

```

```

}

/// <summary>
/// Returns the <see cref="ScriptableRenderer"/> that is used to render this camera.
/// </summary>

public ScriptableRenderer scriptableRenderer
{
    get => UniversalRenderPipeline.asset.GetRenderer(m_RendererIndex);
}

/// <summary>
/// Use this to set this Camera's current <see cref="ScriptableRenderer"/> to one listed
on the Render Pipeline Asset. Takes an index that maps to the list on the Render Pipeline Asset.
/// </summary>

/// <param name="index">The index that maps to the RendererData list on the currently
assigned Render Pipeline Asset</param>

public void SetRenderer(int index)
{
    m_RendererIndex = index;
}

public LayerMask volumeLayerMask
{
    get => m_VolumeLayerMask;
    set => m_VolumeLayerMask = value;
}

public Transform volumeTrigger

```



```

{
    get => m_VolumeTrigger;
    set => m_VolumeTrigger = value;
}

/// <summary>
/// Returns true if this camera should render post-processing.
/// </summary>
public bool renderPostProcessing
{
    get => m_RenderPostProcessing;
    set => m_RenderPostProcessing = value;
}

/// <summary>
/// Returns the current anti-aliasing mode used by this camera.
/// <see cref="AntialiasingMode"/>.
/// </summary>
public AntialiasingMode antialiasing
{
    get => m_Antialiasing;
    set => m_Antialiasing = value;
}

/// <summary>
/// Returns the current anti-aliasing quality used by this camera.
/// <seealso cref="antialiasingQuality"/>.

```

```
/// </summary>

public AntialiasingQuality antialiasingQuality
{
    get => m_AntialiasingQuality;
    set => m_AntialiasingQuality = value;
}

public bool stopNaN
{
    get => m_StopNaN;
    set => m_StopNaN = value;
}

public bool dithering
{
    get => m_Dithering;
    set => m_Dithering = value;
}

public void OnBeforeSerialize()
{
}

public void OnAfterDeserialize()
{
    if (version <= 1)
    {
```

```

        m_RequiresDepthTextureOption = (m_RequiresDepthTexture) ? CameraOverrideOption.On
: CameraOverrideOption.Off;

        m_RequiresOpaqueTextureOption = (m_RequiresColorTexture) ?
CameraOverrideOption.On : CameraOverrideOption.Off;

    }

}

public void OnDrawGizmos()

{

    string path = "Packages/com.unity.render-pipelines.universal/Editor/Gizmos/";

    string gizmoName = "";

    Color tint = Color.white;

    if (m_CameraType == CameraRenderType.Base)

    {

        gizmoName = $"{path}Camera_Base.png";

    }

    else if (m_CameraType == CameraRenderType.Overlay)

    {

        gizmoName = $"{path}Camera_Overlay.png";

    }

}

#if UNITY_2019_2_OR_NEWER

#if UNITY_EDITOR

    if (Selection.activeObject == gameObject)

    {

        // Get the preferences selection color

        tint = SceneView.selectedOutlineColor;

    }

}

}

```

```

        }
#endif

        if (!string.IsNullOrEmpty(gizmoName))
        {
            Gizmos.DrawIcon(transform.position, gizmoName, true, tint);
        }

        if (renderPostProcessing)
        {
            Gizmos.DrawIcon(transform.position, $"{path}Camera_PostProcessing.png", true,
tint);
        }
#else
        if (renderPostProcessing)
        {
            Gizmos.DrawIcon(transform.position, $"{path}Camera_PostProcessing.png");
        }

        Gizmos.DrawIcon(transform.position, gizmoName);
#endif
    }
}

}

MainMenu.cs

using System;
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

namespace Main_Menu_Scene.Main_Menu
{
    public class MainMenu : Additional.Singleton<MainMenu>

```

```

{
    [Header("References")]
    [Header("Buttons")]
    [SerializeField]
    private Button backButton = null;
    [SerializeField]
    private Button infoButton = null;
    [SerializeField]
    private Button bottomLeftButton = null;
    [SerializeField]
    private Button bottomRightButton = null;
    [SerializeField]
    private Button bottomMiddleButton = null;
    [Header("Game Objects and Content")]
    [SerializeField]
    private GameObject bottomBorderGo = null;
    [SerializeField, Tooltip("List of Main Menu Contents")]
    private List<MenuContent> menuContents = new List<MenuContent>();
    [Header("Scroll View")]
    [SerializeField, Tooltip("Content View in Scroll View. In Middle Game Object of Main
Menu")]
    private RectTransform middleContentView = null;
    [SerializeField]
    private Scrollbar scrollbar = null;

    /// <summary>
    /// Min Anchor for control middle content View With Or Without Bottom Border
    /// </summary>
    private Vector2 middleContentViewMinAnchor = Vector2.zero;

    public MenuContent CurrentContent { get; set; } = null;

    protected override void Awake()
    {
        base.Awake();
        //Set Default Anchor to var
        if (middleContentView != null)
        {
            middleContentViewMinAnchor = middleContentView.anchorMin;
        }
        //Hide Bottom Border
        ShowBottomBorder(false);
        //Show First Content
        if(menuContents != null && menuContents.Count > 0)
        {
            HideAllContent();
            //Show First Available Content
            ShowContent(menuContents[0]);
        }
    }
    private void Update()
    {
        ///For Android KeyCode.Escape - back button on phone
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            //Call Back button functions
            if (backButton != null) backButton.onClick.Invoke();
        }
    }
    /// <summary>
    /// Hides ALL Content that contains Main Menu
    /// </summary>
    public void HideAllContent()

```

```

    {
        if (menuContents != null) menuContents.ForEach(mc => HideContent(mc));
    }

    public void ShowContent(MenuContent content)
    {
        //Hide ALL
        HideAllContent();
        //If Back Button not null
        if(backButton != null)
        {
            //If content has previous content, set back button to show previous
            if(content.PreviousContent != null)
            {
                ShowBackButton(true, () => ShowContent(content.PreviousContent));
            }
            else
            {
                //If previous is null, back button disabled, but quit application.
                ShowBackButton(false, () => Application.Quit());
            }
        }
        if (scrollbar != null) scrollbar.value = 1f;
        //Show content
        content.Show();
        //Set as Current
        CurrentContent = content;
    }
    /// <summary>
    /// Hide concrete content
    /// </summary>
    /// <param name="content"></param>
    public void HideContent(MenuContent content)
    {
        content.Hide();
    }
    /// <summary>
    /// Shows Catalog(ue) Content
    /// </summary>
    /// <param name="menuContent"></param>
    /// <param name="catalogContent"></param>
    /// <param name="previousMenu"></param>
    public void ShowCatalogContent(CatalogMenuContent menuContent, CatalogContent
catalogContent, MenuContent previousMenu)
    {
        menuContent.InitializeContent(catalogContent, previousMenu);
        ShowContent(menuContent);
    }
    /// <summary>
    /// Show or Hide Back Button with function in it
    /// </summary>
    /// <param name="enable"></param>
    /// <param name="backButtonFunction"></param>
    public void ShowBackButton(bool enable, Action backButtonFunction = null)
    {
        if(backButton)
        {
            backButton.gameObject.SetActive(enable);
            backButton.onClick.RemoveAllListeners();
            if (backButtonFunction != null)
                backButton.onClick.AddListener(backButtonFunction.Invoke);
        }
        if(infoButton)
        {

```

```

        infoButton.gameObject.SetActive(!enable);
    }
}
/// <summary>
/// Show or Hide Bottom Border (with buttons)
/// </summary>
/// <param name="enable"></param>
public void ShowBottomBorder(bool enable)
{
    bottomBorderGo.SetActive(enable);
    if (middleContentView != null) {
        if (!enable)
        {
            middleContentView.anchorMin = new Vector2(middleContentView.anchorMin.x, 0);
        }
        else
        {
            middleContentView.anchorMin = middleContentViewMinAnchor;
        }
    }
}
/// <summary>
/// Show or Hide bottom buttons in cases of available Action in this buttons. If action
is one, and another is null, middle button will be shown.
/// </summary>
/// <param name="LeftButton"></param>
/// <param name="rightButton"></param>
/// <param name="LeftButtonText"></param>
/// <param name="rightButtonText"></param>
public void ShowBottomButtons(Action leftButton, Action rightButton, string
LeftButtonText = "", string rightButtonText = "")
{
    //if all null, hide Border
    if (leftButton == null && rightButton == null)
    {
        ShowBottomBorder(false);
    }
    //if all buttons have action, show left + right
    else if (leftButton != null && rightButton != null)
    {
        ShowBottomBorder(true);
        if(bottomMiddleButton != null)
            bottomMiddleButton.gameObject.SetActive(false);
        if (bottomLeftButton != null)
        {
            bottomLeftButton.onClick.RemoveAllListeners();
            bottomLeftButton.onClick.AddListener(leftButton.Invoke);
            TextMeshProUGUI text =
bottomLeftButton.GetComponentInChildren<TextMeshProUGUI>();
            if(text != null && LeftButtonText != null)
            {
                text.SetText(LeftButtonText);
            }
            bottomLeftButton.gameObject.SetActive(true);
        }
        if (bottomRightButton != null)
        {
            bottomRightButton.onClick.RemoveAllListeners();
            bottomRightButton.onClick.AddListener(rightButton.Invoke);
            TextMeshProUGUI text =
bottomRightButton.GetComponentInChildren<TextMeshProUGUI>();
            if (text != null && rightButtonText != null)
            {
                text.SetText(rightButtonText);
            }
        }
    }
}

```

```

    }
    bottomRightButton.gameObject.SetActive(true);
}
}
//if one of two have action, show middle
else if(leftButton != null || rightButton != null)
{
    ShowBottomBorder(true);
    if (bottomLeftButton != null) bottomLeftButton.gameObject.SetActive(false);
    if (bottomRightButton != null) bottomRightButton.gameObject.SetActive(false);
    if (bottomMiddleButton != null)
    {
        bottomMiddleButton.onClick.RemoveAllListeners();
        Action notNullAction = rightButton != null ? rightButton : leftButton;
        string notNullText = rightButton != null ? rightButtonText : leftButtonText;
        bottomMiddleButton.onClick.AddListener(notNullAction.Invoke);
        TextMeshProUGUI text =
bottomMiddleButton.GetComponentInChildren<TextMeshProUGUI>();
        if (text != null && notNullText != null)
        {
            text.SetText(notNullText);
        }
        bottomMiddleButton.gameObject.SetActive(true);
    }
}
}
}
}
}
}
}

```

#### CatalogContent.cs

```

using AR_Camera_Scene.Model_Viewer;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace Main_Menu_Scene.Main_Menu
{
    /// <summary>
    /// Scriptable Class for content variables.
    /// </summary>
    [CreateAssetMenu(fileName = "Catalog Content", menuName = "ScriptableObjects/Main
Menu/Catalog Content", order = 0)]
    public class CatalogContent : ScriptableObject
    {
        [Header("Text")]
        [SerializeField, Tooltip("Catalog Title")]
        private string title = null;
        [SerializeField, TextArea(10, 100), Tooltip("Catalog Text")]
        private string text = null;
        [Header("Trigger")]
        [SerializeField, Tooltip("Catalog Trigger Image that will be shown in content")]
        private Sprite triggerImage = null;
        [SerializeField, Tooltip("Catalog Instruction Image, that will be shown after trigger
image")]
        private Sprite triggerInstructionImage = null;
        [SerializeField, Tooltip("Image Trackable Behaviour Game Object, that contains image map
from MaxST SDK website.")]
        private GameObject triggerImageTrackable = null;
        [SerializeField, Tooltip("List of Models List")]
        private List<ModelsList> modelsLists = null;
        [Header("More")]
        [SerializeField, Tooltip("Website link that border button will open")]
        private string websiteAddressForMore = null;
    }
}

```



```

[Header("Main Menu Buttons text")]
[SerializeField, Tooltip("Text for left button (Website Link)")]
private string LeftButtonText = null;
[SerializeField, Tooltip("Text for right button (Scan, Open AR Scene)")]
private string rightButtonText = null;

public string Title => title;
public string Text => text;
public Sprite TriggerImage => triggerImage;
public Sprite TriggerInstructionImage => triggerInstructionImage;

public string WebsiteAddressForMore => websiteAddressForMore;

public string RightButtonText => rightButtonText;
public string LeftButtonText => LeftButtonText;

public GameObject TriggerImageTrackable => triggerImageTrackable;

public List<ModelsList> ModelsLists => modelsLists;
}
}
CatalogMenuContent.cs

```

```

using Managers;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using TMPPro;
using UnityEngine;
using UnityEngine.UI;

namespace Main_Menu_Scene.Main_Menu
{
    public class CatalogMenuContent : MenuContent
    {
        [Header("Catalog")]
        [SerializeField, Tooltip("Catalog Title")]
        private TextMeshProUGUI titleText = null;
        [SerializeField, Tooltip("Catalog Content")]
        private TextMeshProUGUI contentText = null;
        [SerializeField, Tooltip("Catalog Trigger Image for AR Scene")] //Trigger image for AR
        Scene sets in image tracker behaviour in prefab, not here. It's here for content
        private Image triggerImage = null;
        [SerializeField, Tooltip("Trigger Instruction image that")]
        private Image triggerInstructionImage = null;

        public CatalogContent CurrentContent { get; set; } = null;

        protected override void ShowBottomBorderButton()
        {
            if(CurrentContent != null)
            {
                Action moreAction = null;
                if (!string.IsNullOrEmpty(CurrentContent.WebsiteAddressForMore))
                    moreAction = () =>
ApplicationManager.Instance.OpenUri(CurrentContent.WebsiteAddressForMore);
                Action scanAction = null;
                if (CurrentContent.TriggerImage != null)
                    scanAction = () =>
ApplicationManager.Instance.LoadARScene(CurrentContent.TriggerImageTrackable,
CurrentContent.ModelsLists);
            }
        }
    }
}

```

```

        MainMenu.Instance.ShowBottomButtons(moreAction, scanAction,
CurrentContent.LeftButtonText, CurrentContent.RightButtonText);
    }
}
/// <summary>
/// Initialize Menu Content with Catalog Content
/// </summary>
/// <param name="catalogContent">Content of catalog</param>
/// <param name="previousMenu">Menu that open this content</param>
public void InitializeContent(CatalogContent catalogContent, MenuContent previousMenu)
{
    //Set Previous menu that open this content
    previousContent = previousMenu;
    //Set Content
    titleText.SetText(catalogContent.Title);
    contentText.SetText(catalogContent.Text);
    //Set Trigger Pictures
    if (triggerImage != null)
    {
        triggerImage.sprite = catalogContent.TriggerImage;
        if(triggerImage.sprite != null)
        {
            triggerImage.gameObject.SetActive(true);
        }
        else
        {
            triggerImage.gameObject.SetActive(false);
        }

        if (triggerInstructionImage != null)
        {
            triggerInstructionImage.sprite = catalogContent.TriggerInstructionImage;
            if (triggerInstructionImage.sprite != null && triggerImage.sprite != null)
            {
                triggerInstructionImage.gameObject.SetActive(true);
            }
            else
            {
                triggerInstructionImage.gameObject.SetActive(false);
            }
        }
    }
}
//Enable Content for Rebuild LayoutGroup
gameObject.SetActive(true);
//Canvas.ForceUpdateCanvases();
//List<ContentSizeFitter> cs = GetComponentInChildren<ContentSizeFitter>().ToList();
//cs.ForEach(c => c.enabled = false);
//cs.ForEach(c => c.enabled = true);
//Instant Update Layout Group (Need it, 'cause Unity has problem with multiply
Content Size Fitter in children objects that contains LayoutGroupComponent
List<VerticalLayoutGroup> vg =
GetComponentInChildren<VerticalLayoutGroup>().ToList();
//disable group
vg.ForEach(c => c.enabled = false);
//enable group
vg.ForEach(c => c.enabled = true);
//reset layout
vg.ForEach(l => l.SetLayoutVertical());
//Call in Canvas that we need to Update Canvas Now
Canvas.ForceUpdateCanvases();
//Disable Content
gameObject.SetActive(false);
//Set Current as new.
CurrentContent = catalogContent;

```

```

    }
}

```

MenuButton.cs

```

using UnityEngine;
using System.Collections;
using UnityEngine.UI;
using System.Collections.Generic;

namespace Main_Menu_Scene.Main_Menu
{
    [RequireComponent(typeof(Button))]
    public class MenuButton : MonoBehaviour
    {
        [SerializeField, Tooltip("Menu Content in which this button is setted")]
        private MenuContent currentMenuContent = null;
        [Header("On Click")]
        [SerializeField, Tooltip("Menu Content that will be shown, after click")]
        private MenuContent menuContentToShow = null;
        [SerializeField, Tooltip("Catalog Content that will be initialized in Catalog Menu, after click")]
        private CatalogContent catalogContent = null;

        private Button button = null;
        private MainMenu mainMenu = null;

        private void Awake()
        {
            //Set Instance of main menu
            mainMenu = MainMenu.Instance;
            //Get Button
            button = GetComponent<Button>();
            //Set button function
            button.onClick.AddListener(OnButtonClick);
        }

        protected virtual void OnButtonClick()
        {
            //Show Content on button click
            if(menuContentToShow != null)
            {
                if (catalogContent == null)
                {
                    mainMenu.ShowContent(menuContentToShow);
                }
                else
                {
                    mainMenu.ShowCatalogContent(menuContentToShow as CatalogMenuContent,
                    catalogContent, currentMenuContent);
                }
            }
        }
    }
}

```

MenuContent.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```

```

namespace Main_Menu_Scene.Main_Menu
{
    public class MenuContent : MonoBehaviour
    {
        [SerializeField, Tooltip("Reference to previous Menu Content")]
        protected MenuContent previousContent = null;

        public MenuContent PreviousContent => previousContent;
        /// <summary>
        /// Show Content
        /// </summary>
        public virtual void Show()
        {
            gameObject.SetActive(true);
            ShowBottomBorderButton();
        }
        /// <summary>
        /// Initialize(Show or Hide) Bottom Botder Buttons
        /// </summary>
        protected virtual void ShowBottomBorderButton()
        {
            MainMenu.Instance.ShowBottomButtons(null, null);
        }
        /// <summary>
        /// Hide Content
        /// </summary>
        public virtual void Hide()
        {
            gameObject.SetActive(false);
        }
    }
}

```

UIFlippable.cs

```

using UnityEngine;
using UnityEngine.UI;
using System.Collections.Generic;

[RequireComponent(typeof(RectTransform)), RequireComponent(typeof(Graphic)),
DisallowMultipleComponent,
AddComponentMenu("UI/Flippable", 8)]
#if UNITY_5_2 || UNITY_5_3_OR_NEWER
public class UiFlippable : MonoBehaviour, IMeshModifier
{
#else
    public class UIFlippable : MonoBehaviour, IVertexModifier {
#endif
    [SerializeField] private bool m_Horizontal = false;
    [SerializeField] private bool m_Veritical = false;

    private Graphic cachedGraphic;
    private Graphic Graphic
    {
        get
        {
            if (cachedGraphic == null)
            {
                cachedGraphic = GetComponent<Graphic>();
            }

            return cachedGraphic;
        }
    }
}

```

```

}
private Transform cachedTransform;
private Transform Transform
{
    get
    {
        if (cachedTransform == null)
        {
            cachedTransform = transform;
        }

        return cachedTransform;
    }
}

/// <summary>
/// Gets or sets a value indicating whether this <see cref="UnityEngine.UI.UIFlippable"/>
should be flipped horizontally.
/// </summary>
/// <value><c>true</c> if horizontal; otherwise, <c>false</c>.</value>
public bool horizontal
{
    get { return this.m_Horizontal; }
    set
    {
        this.m_Horizontal = value;
        Graphic.SetVerticesDirty();
    }
}

/// <summary>
/// Gets or sets a value indicating whether this <see cref="UnityEngine.UI.UIFlippable"/>
should be flipped vertically.
/// </summary>
/// <value><c>true</c> if vertical; otherwise, <c>false</c>.</value>
public bool vertical
{
    get { return this.m_Veritical; }
    set
    {
        this.m_Veritical = value;
        Graphic.SetVerticesDirty();
    }
}

#if UNITY_EDITOR
protected void OnValidate()
{
    Graphic.SetVerticesDirty();
}
#endif

#if UNITY_5_2 || UNITY_5_3_OR_NEWER
public void ModifyMesh(VertexHelper vertexHelper)
{
    if (!this.enabled)
        return;

    List<UIVertex> list = new List<UIVertex>();
    vertexHelper.GetUIVertexStream(list);

    ModifyVertices(list); // calls the old ModifyVertices which was used on pre 5.2

    vertexHelper.Clear();
}

```

```

        vertexHelper.AddUIVertexTriangleStream(List);
    }

    public void ModifyMesh(Mesh mesh)
    {
        if (!this.enabled)
            return;

        List<UIVertex> List = new List<UIVertex>();
        using (VertexHelper vertexHelper = new VertexHelper(mesh))
        {
            vertexHelper.GetUIVertexStream(List);
        }

        ModifyVertices(List); // calls the old ModifyVertices which was used on pre 5.2

        using (VertexHelper vertexHelper2 = new VertexHelper())
        {
            vertexHelper2.AddUIVertexTriangleStream(List);
            vertexHelper2.FillMesh(mesh);
        }
    }
}
#endif

public void ModifyVertices(List<UIVertex> verts)
{
    if (!this.enabled)
        return;

    RectTransform rt = Transform as RectTransform;

    for (int i = 0; i < verts.Count; ++i)
    {
        UIVertex v = verts[i];

        // Modify positions
        v.position = new Vector3(
            (this.m_Horizontal ? (v.position.x + (rt.rect.center.x - v.position.x) * 2) :
v.position.x),
            (this.m_Veritical ? (v.position.y + (rt.rect.center.y - v.position.y) * 2) :
v.position.y),
            v.position.z
        );

        // Apply
        verts[i] = v;
    }
}
}
}

```

Model.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
namespace AR_Camera_Scene.Model_Viewer
{
    /// <summary>
    /// Class for Model that will be controlled in Model Viewer
    /// </summary>
    public class Model : MonoBehaviour
    {
        /// <summary>
        /// Model ID.
    }
}

```

```

    /// </summary>
    public int Id { get; set; } = -1;
    /// <summary>
    /// Rotation inertia will be reduced for this amount (divided)
    /// </summary>
    public float InertiaReducedByTimeSpeed { get => reduceByTimeSpeed; set =>
reduceByTimeSpeed = value; }
    /// <summary>
    /// Rotation Inertia seek to zero speed
    /// </summary>
    public float InertiaSeekToZeroSpeed { get => inertiaSeekToZeroSpeed; set =>
inertiaSeekToZeroSpeed = value; }

    private Vector3 rotationInertia;
    ///[SerializeField]
    private float reduceByTimeSpeed = 1000f;
    ///[SerializeField]
    private float inertiaSeekToZeroSpeed = 2f;
    /// <summary>
    /// Rotating Model with difference in touch position
    /// </summary>
    /// <param name="rot"></param>
    public void Rotate(Vector3 rot)
    {
        //Reset Inertia
        rotationInertia = Vector3.zero;
        //Init rotation
        float val = Mathf.Deg2Rad * 5f;
        Vector3 newRot = rot * val;
        //
        //
        //Here Rotation
        transform.Rotate(Vector3.back, newRot.y, Space.World);
        transform.Rotate(Vector3.Left, newRot.x, Space.World);
    }
    /// <summary>
    /// Set inertia with difference and time of touching
    /// </summary>
    /// <param name="difference"></param>
    /// <param name="time"></param>
    public void SetInertia(Vector3 difference, float time)
    {
        //Init rotation Inertia
        rotationInertia = difference / time / InertiaReducedByTimeSpeed;
        //Debug.Log($"{difference} : {time} : {InertiaReducedByTimeSpeed} :
{rotationInertia}");
    }

    private void Update()
    {
        //Update rotation Inertia
        rotationInertia = Vector3.MoveTowards(rotationInertia, Vector3.zero,
InertiaSeekToZeroSpeed * Time.deltaTime);
        //
        //
        //Here Rotation
        transform.Rotate(Vector3.back, rotationInertia.y, Space.World);
        transform.Rotate(Vector3.Left, rotationInertia.x, Space.World);
    }
}
}
}

```

ModelListUI.cs

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;

namespace AR_Camera_Scene.Model_Viewer
{
    /// <summary>
    /// List UI - On Top of UI List of Model List.
    /// </summary>
    public class ModelListUI : MonoBehaviour
    {
        [SerializeField, Tooltip("List Title")]
        private TextMeshProUGUI listTitle = null;
        [SerializeField, Tooltip("List Button for click")]
        private Button button = null;
        [SerializeField, Tooltip("Button image for select\ deselect")]
        private Image buttonImage = null;
        [SerializeField, Tooltip("Default Image sprite (when not selected)")]
        private Sprite defaultSprite = null;
        [SerializeField, Tooltip("Selected Image sprite")]
        private Sprite selectSprite = null;

        public TextMeshProUGUI ListTitle => listTitle;

        public ModelViewer ModelViewer { get; private set; } = null;
        public ModelsList ModelsList { get; private set; } = null;
        /// <summary>
        /// Initialize UI with reference to modelViewer that init this and with List
        /// </summary>
        /// <param name="modelViewer"></param>
        /// <param name="modelsList"></param>
        public void Initialize(ModelViewer modelViewer, ModelsList modelsList)
        {
            ModelViewer = modelViewer;
            ModelsList = modelsList;
            button.onClick.RemoveAllListeners();
            button.onClick.AddListener(OnButtonClick);
        }

        private void OnButtonClick()
        {
            if(ModelViewer != null && ModelsList != null)
            {
                ModelViewer.SelectModelList(ModelsList);
            }
        }

        public void Select()
        {
            if(ButtonImage != null && defaultSprite != null && selectSprite != null)
            {
                ButtonImage.sprite = selectSprite;
            }
        }

        public void Deselect()
        {
            if (ButtonImage != null && defaultSprite != null && selectSprite != null)
            {
                ButtonImage.sprite = defaultSprite;
            }
        }
    }
}
```



```

    }
}
}
}

```

ModelsList.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
namespace AR_Camera_Scene.Model_Viewer
{
    [CreateAssetMenu(fileName = "Models List", menuName = "ScriptableObjects/Models", order = 1)]
    public class ModelsList : ScriptableObject
    {
        [SerializeField, Tooltip("List Title")]
        private string listTitle = "";
        [SerializeField, Tooltip("List of Models")]
        private List<Model> models = null;
        public string ListTitle => listTitle;
        public List<Model> Models => models;

        /// <summary>
        /// Initialize List with models ID (For future search)
        /// </summary>
        public void InitializeList()
        {
            if (Models == null) return;
            for(int i = 0; i < Models.Count; i++)
            {
                Models[i].Id = i;
            }
        }
    }
}

```

ModelViewer.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
namespace AR_Camera_Scene.Model_Viewer
{
    public class ModelViewer : Additional.Singleton<ModelViewer>
    {
        [Header("References")]
        [SerializeField, Tooltip("Model List UI Prefab")]
        private ModelListUI ModelListUIPrefab = null;
        [SerializeField, Tooltip("Content in Scroll View for Models Lists")]
        private Transform modelViewContent = null;
        [SerializeField, Tooltip("Transform(Game Object) in which models will be Instantiated (Created)")]
        private Transform currentModelTransform = null;
        [Header("Model View Settings")]
        [SerializeField, Range(0f, 10f)]
        private float rotationSpeed = 2f;
        [SerializeField, Range(1f, 2000f)]
        private float inertiaReducedWhenSet = 1000f;
        [SerializeField, Range(0f, 10f)]
        private float inertiaSeekToZeroSpeed = 2f;
        [Header("For Test")]
    }
}

```

```

[SerializeField, Tooltip("For debug models, just add models in this List and Start Play
with scene that contains Model Viewer")]
private List<ModelsList> tempModelsList = null;

public List<ModelsList> ModelsLists { get; set; } = null;

public ModelsList CurrentModelList { get; private set; } = null;
public Model CurrentModel { get; private set; } = null;

public List<ModelListUI> ModelListUIs { get; private set; } = new List<ModelListUI>();
public Transform CurrentModelTransform => currentModelTransform;
/// <summary>
/// Cached Main Camera
/// </summary>
private Camera camera = null;
/// <summary>
/// Var that contains Previous Touch Position. (previous IN Frame Update touch position,
not previous touch position when "Clicked")
/// </summary>
private Vector3 previousTouchPosition = Vector3.zero;
//Position when touch is pressed
private Vector3 touchDownPosition = Vector3.zero;
//Control touch pressed
private bool touchPressed = false;
//Timer for touch time
private float pressedTimer = 0;
protected override void Awake()
{
    camera = Camera.main;
    if (camera == null) Debug.LogError("Camera is null");
    //Initialize Debug List
    if (tempModelsList != null)
    {
        ModelsLists = tempModelsList;
        InitializeViewer();
    }
}
/// <summary>
/// Initialize Current Model Viewer with Models Lists
/// </summary>
public void InitializeViewer()
{
    //Reset old
    ResetViewer();
    //Create new
    if(ModelsLists != null)
    {
        //Init all lists
        ModelsLists.ForEach(ml => ml.InitializeList());
        //Create uis
        foreach(ModelsList modelsList in ModelsLists)
        {
            CreateModelsList(modelsList);
        }
    }
    //Find available model
    if (ModelsLists != null)
    {
        foreach (ModelsList modelsList in ModelsLists)
        {
            if(modelsList != null && modelsList.Models != null && modelsList.Models.Count
> 0)
            {
                SelectModelList(modelsList);
            }
        }
    }
}

```

```

        break;
    }
}
}
}
/// <summary>
/// Reset viewer
/// </summary>
public void ResetViewer()
{
    ModelListUIs.ForEach(ui => Destroy(ui.gameObject));
    ModelListUIs.Clear();
    if (CurrentModel != null) Destroy(CurrentModel.gameObject);
    CurrentModel = null;
    touchPressed = false;
}
/// <summary>
/// Create Model List in UI
/// </summary>
/// <param name="modelsList"></param>
private void CreateModelsList(ModelsList modelsList)
{
    if(modelsList != null && modelsList.Models != null && modelsList.Models.Count > 0)
    {
        ModelListUI modelListUI = Instantiate(ModelListUIPrefab, modelViewContent);
        modelListUI.ListTitle.text = modelsList.ListTitle;
        modelListUI.Initialize(this, modelsList);
        ModelListUIs.Add(modelListUI);
    }
}
private void Update()
{
    if(CurrentModel != null)
    {
        HandleCurrentModelView();
    }
}
/// <summary>
/// Handle Input for control and view current model
/// </summary>
/// <param name="model"></param>
private void HandleCurrentModelView()
{
    if(Input.GetMouseButtonDown(0))
    {
        touchPressed = true;
        previousTouchPosition = Input.mousePosition;
        touchDownPosition = Input.mousePosition;
    }
    if(Input.GetMouseButtonUp(0))
    {
        touchPressed = false;
        if (CurrentModel != null) CurrentModel.SetInertia(touchDownPosition -
Input.mousePosition, pressedTimer);
        pressedTimer = 0;
    }
    if(touchPressed && CurrentModelList != null && CurrentModel != null)
    {
        pressedTimer += Time.deltaTime;
        Vector3 currentTouchPosition = Input.mousePosition;
        //bool isRight = currentTouchPosition.x > previousTouchPosition.x;
        CurrentModel.Rotate((previousTouchPosition -
currentTouchPosition)*rotationSpeed);
        previousTouchPosition = currentTouchPosition;
    }
}

```

```

    }
}

public void SelectModelList(ModelsList modelList)
{
    CurrentModelList = modelList;
    ChangeCurrentModel(CurrentModelList.Models.Find(m => m));
    ModelListUIs.ForEach(ml => ml.Deselect());
    ModelListUI modelListUI = ModelListUIs.Find(ml => ml.ModelsList == modelList);
    if(modelListUI != null)
    {
        modelListUI.Select();
    }
}

public void ChangeCurrentModel(Model model)
{
    if (CurrentModel != null) Destroy(CurrentModel.gameObject);
    Model instantiatedModel = Instantiate(model, CurrentModelTransform);
    instantiatedModel.Id = model.Id;
    CurrentModel = instantiatedModel;
    CurrentModel.InertiaReducedByTimeSpeed = inertiaReducedWhenSet;
    CurrentModel.InertiaSeekToZeroSpeed = inertiaSeekToZeroSpeed;
}

public void ChangeToNextModel()
{
    if (CurrentModel != null && CurrentModelList != null && CurrentModelList.Models !=
null)
    {
        int currentIndex = -1;
        for (int i = 0; i < CurrentModelList.Models.Count; i++)
        {
            if(CurrentModelList.Models[i].Id == CurrentModel.Id)
            {
                currentIndex = i;
                break;
            }
        }
        if(currentIndex >= 0)
        {
            currentIndex = (currentIndex + 1) % CurrentModelList.Models.Count;
            ChangeCurrentModel(CurrentModelList.Models[currentIndex]);
        }
        else
        {
            Debug.LogError("Model Not Found...");
        }
    }
}

public void ChangeToPreviousModel()
{
    if (CurrentModel != null && CurrentModelList != null && CurrentModelList.Models !=
null)
    {
        int currentIndex = -1;
        for (int i = 0; i < CurrentModelList.Models.Count; i++)
        {
            if (CurrentModelList.Models[i].Id == CurrentModel.Id)
            {
                currentIndex = i;
                break;
            }
        }
        if (currentIndex >= 0)
    }
}

```

```
{
  currentIndex = (currentIndex - 1);
  if(currentIndex < 0)
  {
    currentIndex = CurrentModelList.Models.Count + currentIndex;
  }
  ChangeCurrentModel(CurrentModelList.Models[currentIndex]);
}
else
{
  Debug.LogError("Model Not Found...");
}
}
}
}
}
```