

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Доповнена реальність	6
1.1.1 Передмова	6
1.1.2 Класифікація систем доповненої реальності.....	7
1.1.3 Мобільні системи доповненої реальності.....	10
1.1.4 Технології для роботи з AR.....	13
1.2 Мобільні пристрої.....	15
1.3 Програми-аналоги.....	17
Висновки до розділу 1	21
РОЗДІЛ 2. ОПИС ЗАСОБІВ РОЗРОБКИ ТА РОБОТИ ДОДАТКУ	22
2.1 Постановка задачі дипломного проектування.....	22
2.2 Інструментальні засоби для реалізації	22
2.2.1 ARToolkit.....	22
2.2.2 Java.....	26
2.2.3 Android SDK.....	26
2.2.4 Android NDK	27
2.2.5 Java Native Interface.....	28
2.2.6 Android Studio	30
2.2.7 Open GL	31
2.2.8 Формат .obj і .mtl.....	32
2.3 Вхідні та вихідні дані.....	33
Висновки до розділу 2.....	34
РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	35
3.1 Опис математичної моделі	35
3.1.1 Геометричні перетворення в тривимірному просторі	35
3.1.2 Розрахунок параметрів матриці	37
3.2 Підготовка проекту	39
3.2.1 Інсталяція та налаштування Android Studio	39
3.2.2 Створення проекту та інтеграція JNI.....	40

					ІАЛЦ.467100.003 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Сухина			Система візуалізації інтер'єру з використанням доповненої реальності		
Перевір.		Радченко					
Н. контр.		Сімоненко					
Затверд.		Стіренко					
					Літ.	Арк.	Аркуші
						1	64
					НТУУ КПІ ФІОТ Група ІО-32		

3.2.3	Налаштування ARToolkit	43
3.2.4	Налаштування маніфест файлу	44
3.3	Архітектура програмного забезпечення	44
3.3.1	MVC	44
3.3.2	Модель	46
3.3.3	Контролер	47
3.3.4	Представлення	48
3.3.5	C/C++ функціонал	49
3.4	Інтеграція ресурсів в додаток	50
3.4.1	Завантаження 3D моделей	51
3.4.2	Генерація ARToolkit маркера	51
3.5	Збірка проекту	54
3.5.1	Збірка нативної бібліотеки	54
3.5.2	Генерація APK	55
3.6	Опис реалізованого програмного забезпечення	56
	Висновки до розділу 3	61
	ВИСНОВКИ	62
	ЛІТЕРАТУРА	63

ВСТУП

Розвиток технологій у ХХІ столітті є невимовно швидким. Грань між науковою фантастикою і наукою нівелюється все більше і більше з кожним днем. Ще 200 років тому люди були б в шоці від технічних досягнень ХХІ століття. Літаки, автомобілі та смартфони - цим сьогодні не здивуєш нікого. Кожен рік на світових конференціях присвячених технологіям представляється величезна кількість нових винаходів та технологій.

Однією з найпопулярніших технологій останнього часу є доповнена реальність. Доповнена реальність (Augmented Reality скор. AR) - це технологія об'єднання реального і віртуального світів, коли цифрова інформація у вигляді тексту, зображення, відео, звуку доповнює об'єкти і явища фізичного світу. Простими словами, доповнена реальність - накладення віртуальних об'єктів поверх реальних.

Ця технологія має безпосередній вектор у майбутнє. Її застосування можна легко представити у навігації. Наприклад, використовуючи смартфон (обладнаний камерою), можна буде без проблем знайти необхідний шлях, навіть якщо у вас є проблеми з орієнтацією на карті. Такий результат може бути досягнений завдяки синхронній роботі систем глобального позиціонування, завантаженням, на смартфон, картам, і алгоритмам, що зможуть прив'язати данні з карти на зображення, що отримується з камери смартфона. Як результат, ми зможемо побачити виділений шлях (наприклад, зеленою лінією), що буде проектуватися у реальність і вказуватиме необхідну користувачеві дорогу.

Використання технології доповненої реальності підвищує ефективність розробки нового продукту. Замість того, щоб знову і знову створювати фізичні прототипи, AR дозволяє компаніям використовувати віртуальні моделі САПР, суміщені з реальними пристроями. Це економить час і допомагає виявити помилки на ранніх етапах проектування, дає зрозуміти як працюватиме чергове удосконалення продукту. Зовнішні вимірювальні датчики в поєднанні з камерою і програмним забезпеченням можуть бути використані для

					ІАЛЦ.467100.003 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

накладення даних САПР безпосередньо на реальні компоненти з точністю менше міліметра.

Також, доповнена реальність може бути використана у допомозі німим людям. Кінчики пальців, обличчя та губи можуть бути розпізнані, як маркери і комбінації їх рухів, з яких завчасно можна створити базу даних, будуть інтерпретуватись в слова. Таким чином можна створити додатки, які допоможуть людям, не знайомим із мовою німим, без проблем спілкуватись із ними. Також, можна розробити клавіатуру швидкого набору для німим.

Іншою галуззю використання доповненої реальності може стати реклама. Як вже говорилося у абзаці вище, інформація буде набагато більш гнучкою до оновлення. А це означає, що фінансові затрати на оренду площі та на виготовлення матеріалів можна буде значно скоротити. До того ж модна буде уникнути «забруднення» реального простору рекламою, що зараз займає кожний вільний метр.

І звичайно доповнена реальність може і буде широко використовуватись у розвагах і іграх. Інтерактивні журнали, ігри у реальному часі і просторі, все це вже можна зустріти серед існуючих програмних продуктів.

Доповнена реальність вже зараз застосовується в медицині, навігації, військовій справі, мистецтві, дизайні, біології, архітектурі, археології, офісній роботі та інших видах людської діяльності. І прогрес не стоїть на місці. З'являється нове покоління апаратних пристроїв, таких як смарт окуляри, безконтактні сенсорні контролери. З'являються нові прототипи інтерфейсів користувачів для доповненої реальності. Випускаються нові версії бібліотек для розробників.

З розвитком технологій доповнена реальність все глибше проникатиме в наше повсякденне життя. Зараз створюється і накопичується все більша і більша кількість даних і AR буде служити мостом для доступу до величезної кількості інформації. Причому Augmented Reality буде інтегрована настільки глибоко, що стане сприйматися інтуїтивно. І сам термін "Доповнена реальність", можливо, перестане використовуватися, а цифрові дані будуть

					ІАЛЦ.467100.003 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

вважатися частиною самої реальності. Головне, що доповнена реальність дозволяє перенести нас на новий рівень взаємодії з цифровим світом без необхідності відмовлятися від звичного оточення, і повністю занурюватися у віртуальність.

Таким чином, мета моєї дипломної роботи – дослідити існуючі методи роботи із доповненою реальністю і створити мобільний додаток для демонстрації AR технології на прикладі моделювання інтер'єру.

Для досягнення мети були посалені наступні завдання:

- Зробити поверхневий огляд існуючих методів роботи із AR;
- Зробити детальний огляд ARToolkit;
- Створити мобільний додаток.

					ІАЛЦ.467100.003 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Доповнена реальність

1.1.1 Передмова

Термін доповненої реальності (augmented reality, AR) імовірно був запропонований працюючим на корпорацію Boeing дослідником Томом Коделом в 1990 р

Важливо розуміти відмінності між доповненою реальністю і змішаною реальністю. У широкому сенсі доповнена реальність являє собою процес перегляду реального світу і віртуальних об'єктів одночасно, де віртуальна інформація накладається, вирівнюється і інтегрується в фізичному світі. У літературі з людино-машинної взаємодії доповнена реальність знаходиться в безперервному діапазоні інтерфейсів від «реальності» до віртуальної реальності «повного занурення»

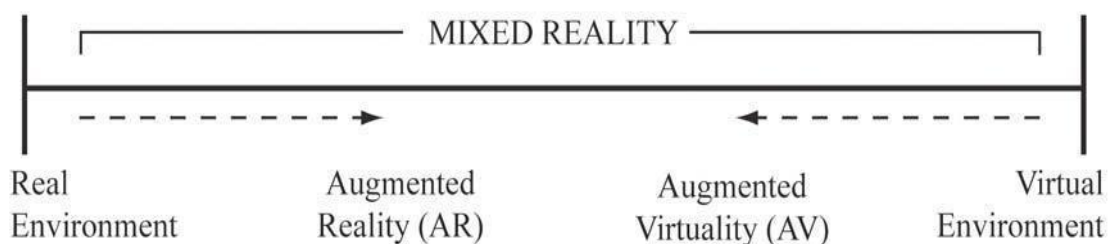


Рис. 1.1. Діапазон технологій змішаної реальності

Істотною відмінністю доповненої реальності від віртуальної є збереження фізичного світу як контексту, в якому представлені віртуальні об'єкти і з яким вони взаємодіють. Віртуальна реальність повністю абстрагується від фізичного світу, щоб помістити користувача повністю у віртуальний світ. Віртуальна реальність використовує спеціальні позиційні трекери з дисплеями (окуляри віртуальної реальності), які динамічно оновлюють видимий користувачем простір у віртуальному середовищі. Важливо розуміти, що доповнена реальність повністю змінює це парадигму, і

в підсумку віртуальні об'єкти розміщуються в реальному оточенні користувача.

Таким чином, доповнена реальність - це технології, що дозволяють доповнювати зображення реальних об'єктів різними об'єктами комп'ютерної графіки, а також поєднувати зображення, отримані від різних джерел комп'ютерного середовища: відеокамер, акселерометрів, компасів і т.д. Схема середовища доповненої реальності представлена на рис. 1.2. На відміну від «віртуальної реальності», яка передбачає повністю штучний синтезований світ (відеоряд), доповнена реальність припускає інтеграцію віртуальних об'єктів у природні відеосцени.

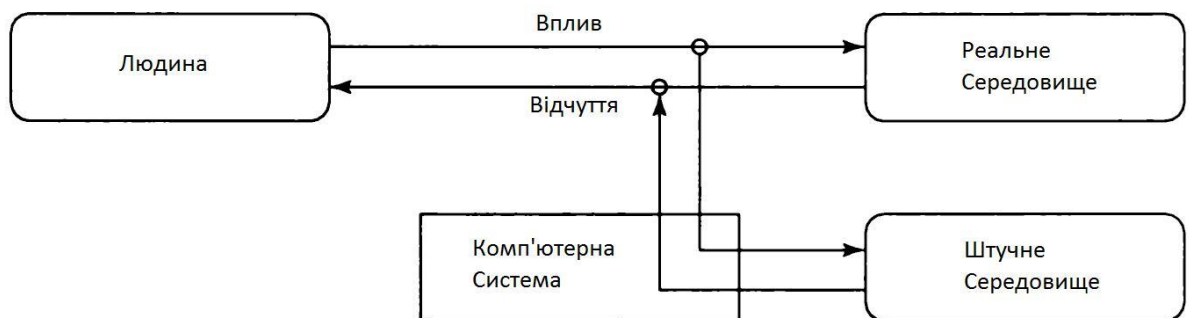


Рис. 1.2. Схема середовища доповненої реальності

Рональд Азума (Ronald Azuma)[1] виділив ряд ознак, якими має володіти доповнена реальність:

- 1) комбінування реального та віртуального світу;
- 2) інтерактивність;
- 3) тривимірне представлення об'єктів.

1.1.2 Класифікація систем доповненої реальності

Людина отримує уявлення про навколишній простір за допомогою великого набору органів сприйняття навколишньої інформації. Система доповненої реальності, будучи посередником між людиною і реальністю, повинна створювати сигнал для одного з таких органів. Таким чином, за типом подання інформації системи доповненої реальності бувають:

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

Аудіо. Такі системи орієнтовані на слухове сприйняття [2]. Найчастіше такі системи використовуються в навігації. Наприклад, вони видають спеціальні сигнали, коли людина досягає певного місця. Можливе використання стереоскопічного ефекту, що дозволяє людині йти в потрібному напрямку, орієнтуючись на джерело звуку. Прикладом такої системи є Hear & There [3].

Аудіовізуальні. Це комбінація двох попередніх типів, однак, аудіоінформація в них має лише допоміжний характер. Системи доповненої реальності завжди потребують інформації, одержуваної з навколишнього середовища. Саме на основі цих даних будуються віртуальні об'єкти. Кожна з таких систем володіє певним набором сенсорів - пристроїв, що дозволяють збирати інформацію з навколишнього середовища: звукові і електромагнітні коливання, прискорення і т.д. Для класифікації має сенс розділяти сенсори не по типам реєстрованих фізичних величин, а за їх призначенням, оскільки подібні за своєю природою сигнали можуть нести різну інформацію. За типом сенсорів можна виділити наступні системи:

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

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

Системи доповненої реальності можна розрізняти за ступенем взаємодії з користувачем. У деяких системах користувач грає пасивну роль, він лише спостерігає за реакцією системи на зміни в навколишньому середовищі. Інші ж системи вимагають активного втручання користувача - він може управляти

					ІАЛЦ.467100.003 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

як роботою самої системи, для досягнення результатів, так і змінювати віртуальні об'єкти. За цією ознакою системи діляться на:

Автономні. Вони не вимагають втручання користувача. Завдання таких систем зводиться до надання інформації про об'єкти. Наприклад, подібні системи можуть аналізувати об'єкти, що знаходяться в полі зору людини і видавати довідкову інформацію про них. Також системи такого типу використовуються в медицині. Наприклад, система Gait Aid[4] для людей з порушеннями опорно-рухового апарату. Вона шляхом використання віртуальних об'єктів надає мозку додаткову інформацію, яка допомагає координувати рухи.

Інтерактивні. Такі системи засновані на взаємодії з користувачем. На різні дії користувач отримує різну відповідь. У подібних системах необхідно мати пристрій введення інформації. В якості такого пристрою може застосовуватися сенсорний екран мобільного телефону, планшет або спеціальний маніпулятор. Вибір пристроїв введення залежить від специфіки системи. У разі простих дій з віртуальним об'єктом, достатньо простого вказівного пристрою. Якщо ж необхідна імітація будь-яких реальних процесів і виконання складних маніпуляцій з об'єктами використовуються спеціальні маніпулятори, які мають різну кількість ступенів свободи. Прикладом можуть служити пристрої PHANTOM [5].

Інтерактивність виражається в різному ступені. Бувають системи, що дозволяють користувачеві активно змінювати віртуальне середовище.

Зазвичай це системи-симулятори будь-яких реальних дій. Вони використовуються у разі, коли використання реальних об'єктів неможливо, наприклад, спеціалізовані медичні тренажери, що дозволяють початківцям лікарям відпрацьовувати необхідні навички.

Існують інші системи, де користувачеві не потрібно змінювати віртуальне середовище. Замість цього користувач обирає, які віртуальні об'єкти він хоче побачити. Користувач також має можливість маніпулювати віртуальними об'єктами, але не на рівні структури, а на рівні відображення,

					ІАЛЦ.467100.003 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

тобто застосовувати, наприклад афінні перетворення типу повороту, переміщення і т.д. До даної групи можна віднести різні архітектурні системи, що дозволяють побачити, як впишеться в реально існуючу обстановку нова споруда або його частина, а також навігаційні та геоінформаційні системи. Подібні системи можуть показувати частини об'єктів інтересу, приховані іншими будівлями, додаткову інформацію про обрані об'єктах і т.д.

За ступенем мобільності системи доповненої реальності можна класифікувати як:

Стационарні. Системи цього типу призначені для роботи в фіксованому місці; переміщення таких систем означає часткове або повне припинення їх працездатності.

Мобільні. Системи цього типу можуть без зусиль переміщатися; часто таке переміщення і лежить в основі виконуваної ними функції. Належність до того чи іншого типу визначається функціями системи. Так, симулятор хірургічного столу не повинен бути мобільним, оскільки його завдання - відтворити для людини спеціальні умови, максимально наближених до реальних. У той же час навігаційна система повинна бути якомога більш мобільною, щоб вона могла переміщатися разом з транспортним засобом або людиною, не створюючи додаткових витрат на її переміщення.

1.1.3 Мобільні системи доповненої реальності

Мобільні системи доповненої реальності включають в себе мобільні додатки для телефонів. Використання мобільних телефонів для доповненої реальності має як переваги так і недоліки. Більшість мобільних пристроїв в даний час обладнано камерами, що робить мобільний телефон однією з найбільш зручних платформ для реалізації систем доповненої реальності. Крім того, більшість сучасних телефонів мають додаткові вбудовані датчики такі як: акселерометри, магнітометри і GPS-приймачі, які можуть поліпшити роботу AR програми.

					ІАЛЦ.467100.003 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

У мобільній доповненої реальності (AR), користувачі дивляться на пряме зображення, отримане з відеокамери на їх мобільному пристрої і сцени, які вони бачать (тобто реальний світ) збагачуються (доповнюються) інтегрованими тривимірними віртуальними об'єктами (тобто об'єктами доповненої реальності). Ця технологія має величезний потенціал у таких областях, як реклама, навігація, розваги, культурно-виставкова сфера і т.д.

Якщо віртуальний об'єкт просто накладається на реальне зображення, а не інтегрується в нього, то для створення середовища доповненої реальності можуть бути використані додаткові сенсори, присутні в сучасних мобільних пристроях, такі як акселерометр, компас, GPS. Використовуючи інформацію про місцезнаходження, користувач може переміщатися по світу доповненої реальності. Якщо віртуальні об'єкти мають безпосередній зв'язок з реальним світом, більшу ніж просто глобальне положення, наприклад віртуальне будівля, побудована на реальному пустирі, то для такої доповненої реальності необхідна додаткова інформація, така як кордони пустиря і його розміри. Отримання цієї додаткової інформації зазвичай досягається за допомогою спеціальних маркерів або за допомогою спеціальних функцій розпізнавання.

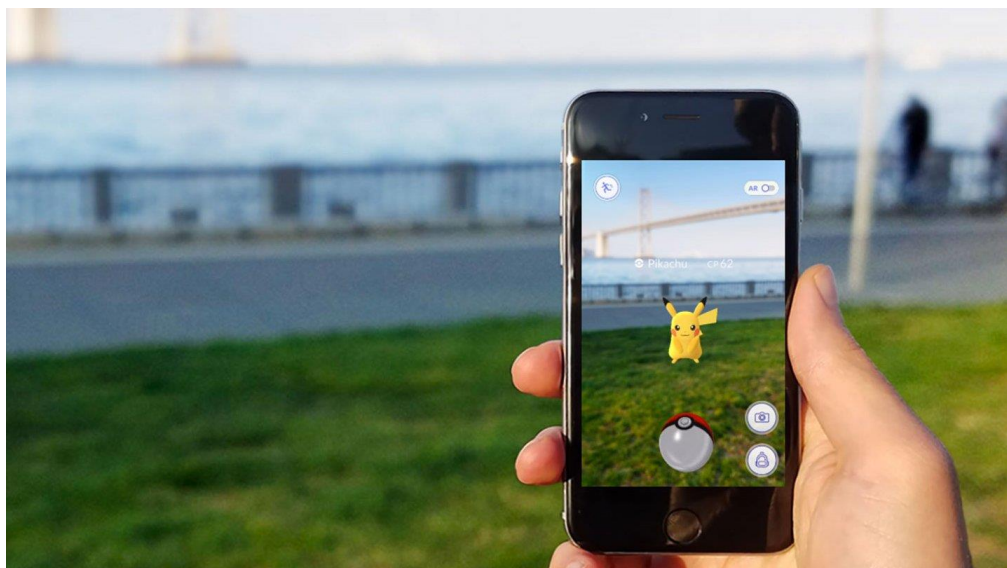


Рис. 1.3. AR на мобільному пристрої

Такими маркерами можуть слугувати: зображення, яке підготували завчасно, елементарні тривимірні фігури або об'єкти, які можуть розпізнаватися завдяки додатковим алгоритмам.

Добре відомо, що для якісних AR систем, щоб надати реалістичний результат потрібно дуже точно відстежувати реальну середу для подальшої інтеграції в неї віртуальних об'єктів. Найбільш поширений тип системи спостереження для мобільних систем - це стеження шляхом комбінування даних, що надходять з декількох датчиків. У вуличних системах в основному використовують GPS або інерційні методи відстеження за допомогою акселерометрів, гіроскопів, компасів та інших датчиків, поряд з методами комп'ютерного зору. Система GPS забезпечує простоту відстеження, незважаючи на малу точність. Для більш точної оцінки положення користувача і його орієнтація GPS використовується в поєднанні з різними інерційних датчиками. Таким чином, точки інтересу користувача звужуються, і це дозволяє спростити візуальне відстеження. У приміщенні GPS володіє поганими показниками, а отже не може бути використаний, тому використовуються тільки візуальні і інерційні методи. Поєднання цих методів має свої особливості: візуальне відстеження досягає найкращих результатів при низькій частоті руху, а інерційні датчики краще працюють при високій частоті руху. Під час повільного руху вони не дають добрих результатів через дрейф зсуву. Взаємодоповнюючий характер цих систем призводить до спільного їх використання в більшості гібридних систем.

Деякі системи покладаються тільки на комп'ютерний зір, але більшість з них розраховані на роботу в приміщеннях, де навколишнє середовище легко контролюється. Коли справа доходить до візуального відстеження на вулиці, з'являються зовнішні фактори, які значно ускладнюють завдання. Однією з найбільш "просунутих" мобільних систем є Google Goggles; ця система може: розпізнавати об'єкти простої форми, наприклад, штрих-коди

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

або книги; визначати місце розташування й напрямок руху, завдяки GPS і акселерометру, які допомагають системі визначити напрям погляду, щоб звузити точку інтересу.

1.1.4 Технології для роботи з AR

OpenCV - бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов. Може вільно використовуватися в академічних та комерційних цілях - поширюється в умовах ліцензії BSD.

Vuforia SDK - це програмне забезпечення для мобільних пристроїв, яке дозволяє створювати додатки доповненої реальності. Воно використовує технологію комп'ютерного зору для того, щоб розпізнавати і відстежувати плоскі зображення і прості 3D-об'єкти в режимі реального часу. Ця можливість реєстрації зображень дозволяє визначати розташування й орієнтації віртуальних об'єктів, таких як 3D-моделі, в реальному світі, коли вони розглядаються через камеру мобільного пристрою. Положення і орієнтація віртуального об'єкта відстежується в реальному часі, так що точки зору глядача на об'єкт співвідноситься з їх точкою зору на зображення, так, що здається, що віртуальний об'єкт є частиною реальної сцени світу.

Vuforia забезпечує API для C++, Java, Objective-C і .Net мов. Є розширення ігрового движка Unity. Таким чином, SDK підтримує як рідні для IOS і Android мови, так і одночасно дозволяє розробляти додатки доповненої реальності в Unity, які можна легко портувати на обидві платформи. Саме тому, додатки, розроблені з використанням Vuforia, сумісні з широким спектром мобільних пристроїв, включаючи iPhone, iPad, Android - телефонів і планшетів під управлінням ОС Android версії 2.2 або вище і ARmv6 або 7 процесор з FPU.

ARToolkit - це бібліотека комп'ютерного стеження для створення додатків з доповненою реальністю[6]. Для цього він використовує можливості

					ІАЛЦ.467100.003 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

відео спостереження, розрахунок реального стану та орієнтації камери по відношенню до квадратного фізичного маркеру в режимі реального часу. Коли реальний стан камери відомо, віртуальна камера може бути розташована в тій же точці і 3D модель накладається на реальний маркер. Так ARToolKit вирішує дві ключові проблеми в доповненої реальності: відстеження погляду і віртуальної взаємодії об'єктів.

ARToolKit (Artoolkit About, 2017) був спочатку розроблений Hirokazu Kato Нара інституту науки і технологій в 1999 році і був випущений в університеті Лабораторія НІТ, у Вашингтоні. В даний час він функціонує як проект з відкритим розміщенням на SourceForge з комерційною ліцензією доступна ARToolWorks. ARToolKit дуже широко використовується (бібліотека з більш ніж 160000 завантажень з 2004 року).

Metaio SDK - готова бібліотека для створення мобільних додатків доповненої реальності. Використовує OpenGL використовує SLAM методи для більш точної роботи

String - бібліотека для створення мобільних додатків орієнтована на iOS пристрою.

Таблиця 1.1. Порівняння технологій AR

Альтернативи	Кросплатформенність	Підтримка мов програмування	Тип ліцензування	Загальна оцінка(0-5)
OpenCV	iOS, PC, Android, Linux	C++,Python,Java, Ruby, Matlab	Boost Software License	4
Vuforia	iOS, Android, Unity	C++,Java, Objective-C, .NET	Free+Comercial SDK option	4
ARToolKit	iOS, Android, Unity	Java, Objective-C,	Free+Comercial SDK option	4
Metaio	iOS, Android, Unity, WEB	Java, Objective-C, .NET	Free+Comercial SDK option	3
String	iOS, Unity	Objective-C, .NET	Comercial SDK option	2

Огляд існуючих технологій для роботи з AR з урахуванням найбільш важливих критеріїв показав, що найоптимальнішим середовищем для розробки є ARToolKit. ARToolKit - безкоштовна бібліотека, постійно модернізується, дозволяє працювати як з нативними додатками, так і створювати кроссплатформенні програми за допомогою спеціального ігрового рушія.

1.2 Мобільні пристрої

У сучасному світі все більшого поширення набувають пристрої на базі мобільних платформ. Це зумовлено потребою сучасної людини бути завжди на зв'язку. Але якщо десять років тому мобільний телефон був всього лише засобом зв'язку - пересувною версією стаціонарного апарату то в наш час у поняття мобільний телефон вкладається надзвичайно широкий набір функцій. За допомогою смартфона можна робити високоякісні фотографії і відео, отримувати, зберігати, відтворювати і передавати значні обсяги даних, користуватися Інтернетом та грати в ігри, про якість та деталізації яких ще десять років тому не могли мріяти і власники настільних комп'ютерів. Смартфон володіє наступними особливостями:

- має операційну систему
- зручний великий дисплей (зазвичай сенсорний);
- багатоядерний процесор;
- графічна карта, що дозволяє обробку значних обсягів мультимедійної інформації без завантаження основного процесора;
- оперативна пам'ять, яка дозволяє оперувати великими обсягами даних;
- значний обсяг вбудованої пам'яті, а також можливість підключення зовнішніх SD карт;

Як і персональні комп'ютери, сучасні смартфони потребують операційну систему. На сьогоднішній день існує широкий спектр мобільних платформ, основні з яких наведено нижче:

					ІАЛЦ.467100.003 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Android. Операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux [7]. Підтримується альянсом Open Handset Alliance (ОНА). Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java. Android дозволяє створювати Java-додатки, що керують пристроєм через розроблені Google бібліотеки. Android Native Development Kit дозволяє перенести бібліотеки і компоненти додатків, написані на C та інших мовах.

Windows Phone. Мобільна операційна система, розроблена Microsoft, вийшла 11 жовтня 2010 року[8]. Операційна система є наступником Windows Mobile, хоча і несумісна з нею, з повністю новим інтерфейсом і вперше з інтеграцією сервісів Microsoft: ігрового Xbox Live і медіаплеєра Zune. На відміну від попередньої системи, Windows Phone більшою мірою орієнтований на ринок споживачів, ніж на корпоративну сферу.

BlackBerry OS. Компактна операційна система для мобільних пристроїв з основним набором застосунків[9]. BlackBerry OS працює на ряді пристроїв — смартфони й комунікатори, що випускаються компанією Research In Motion Limited (RIM).

IOS. Заснована на спеціальній версії Mac OS X[10]. Особливістю системи є закритий вихідний код. Користувацький інтерфейс iOS заснований на концепції прямої маніпуляції з використанням жестів Multi-Touch. Елементи інтерфейсу управління складаються з повзунків, перемикачів і кнопок. Він призначений для безпосереднього контакту користувача з екраном пристрою. Внутрішній акселерометр використовуються деякими програмами для реагування на струшування пристрою, яке є також загальною командою скасування, або обертати пристрій у трьох вимірах, що є загальною командою перемикання між книжковим та альбомним режимами.

Також крім цих операційних систем існує досить багато й інших, хоча вони менш популярні: Symbian, Palm OS, Linux та інші.

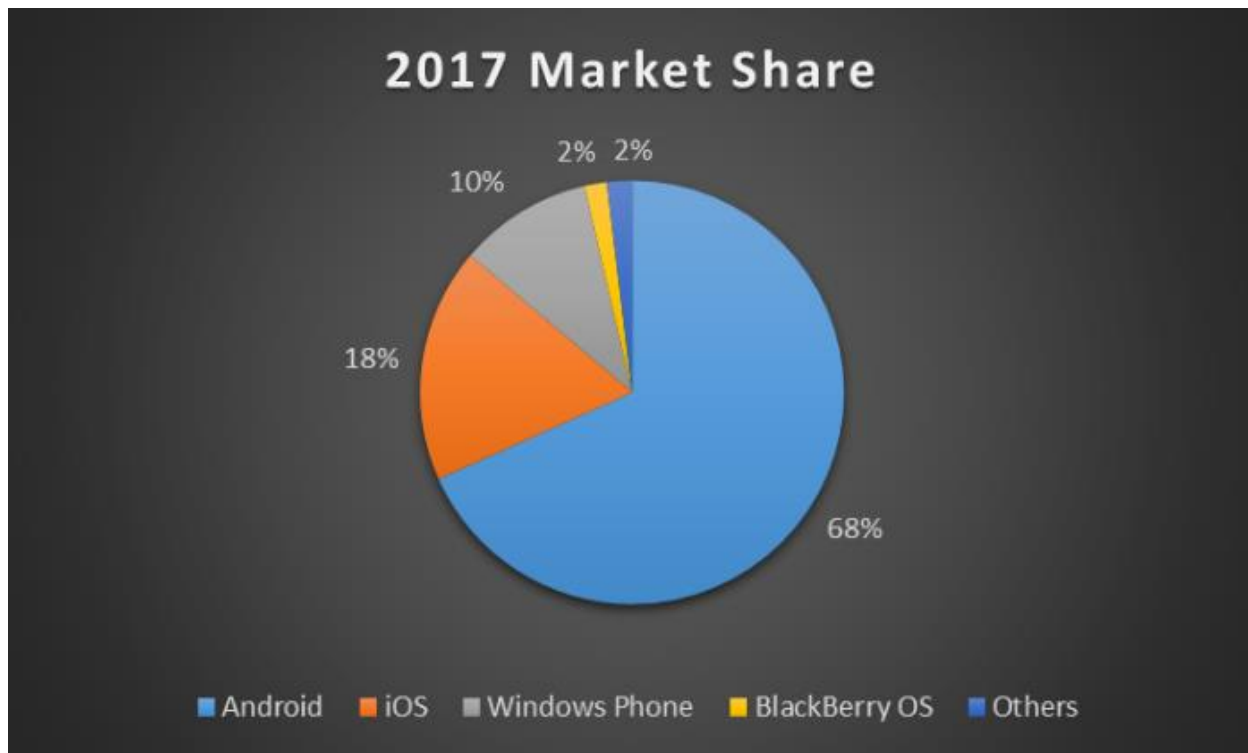


Рис. 1.4. Розподіл мобільних ОС станом на початок 2017

Огляд існуючих мобільних пристроїв та операційних систем показав, що для розробки AR додатків найоптимальнішою буде OS Android. Android - безкоштовна мобільна операційна система з відкритим кодом, постійно модернізується. Наразі Android є і найпопулярнішою операційною системою для мобільних пристроїв, її частка становить близько 70%, це означає, що додатки написані під Android матимуть більший попит, ніж додатки для інших операційних систем. Також для розробки додатків під Android є добре задокументоване Android SDK і безкоштовне середовище розробки - Android Studio.

1.3 Програми-аналоги

Оскільки ціловою платформою було обрано операційну систему Android, то розглядатися будуть програми аналоги саме під неї. На сьогодні в Play Market (офіційний магазин Android додатків від Google), можна знайти десятки додатків з використанням технології AR. Я розгляну найпопулярніші з них.

IKEA. Додаток від компанії IKEA було випущено ще в 2013 році, задовго до того, як технологія доповненої реальності набула широкої популярності. Додаток дозволяє подивитися, як будуть виглядати меблі у вашому інтер'єрі. В якості маркерного зображення виступає каталог меблів. Щоб продемонструвати технологію доповненої реальності необхідно вибрати з каталогу предмет інтер'єру, який необхідний і розмістити каталог в тому місці, де ви хотіли б бачити нові меблі. Після чого навести мобільний пристрій з запущеним додатком на маркер. В режимі реального часу додаток побудує 3d модель обраного вами предмету. Можна змінювати її розмір і розташування п. Додаток може бути корисним не тільки для покупців IKEA, які хочуть визначитися з вибором до покупки. Це також зручний спосіб почерпнути нові ідеї для облаштування будинку і відразу візуалізувати їх.

SmartReality. Додаток SmartReality[11] є захоплюючим об'єднанням смартфона, віртуальної і доповненої реальності. Призначений для професійних архітекторів, який дозволяє інтегрувати свої креслення в додаток, які потім будуть перетворені в SmartReality-сумісні файли. Після того, як файл буде завантажений і проаналізований, користувачі можуть використовувати свої ж креслення в якості маркерів і перевести їх в будь-який віртуальної реальності або доповненої реальності.

Програмне забезпечення сумісне з різними VR і AR лобовому встановлені дисплеї. У режимі VR користувачі можуть буквально пройти через свої плани. Цей тип взаємодії є надзвичайно корисним, щоб отримати відчуття масштабу і розуміння того, як люди всередині будівлі будуть рухатися.

В AR режимі SmartReality створює миттєву 3D модель на столі або іншій поверхні. Обидві функції програми представляють новий захоплюючий спосіб для архітекторів, щоб взаємодіяти зі своїми творіннями, перш ніж вони побудовані. Спираючись на креслення, програмне забезпечення дозволяє архітекторам визначити можливі помилки на етапі проектування і вчасно їх виправити.



Рис. 1.5. Інтерфейс додатку SmartReality

Augment. Додаток Augment використовує маркерну технологію, серед плюсів якої є можливість самостійно управляти контентом, а простіше кажучи, вертїти маркер разом з моделлю, як заманеться[12]. На сайті компанії пропонується завантажити мітки трьох розмірів, що покривають самі різні потреби, від мініатюрної, до зображення для листа ватману. Звичайно, з можна і не роздруковувати, користуючись зображенням з екрану, але в цьому випадку не гарантується дотримання задуманих розмірів моделей. Augment дозволяє розташовувати моделі товарів в масштабі 1:1 в квартирі, на прилавках та інших місцях, де вони можуть стати в нагоді. Клієнти інтернет-магазинів, що потурбуватися про наявність такого AR-функціоналу, можуть уникнути мас сумнівів і страхів, «поставивши» потрібний товар на своєму столі або навіть покрутивши його в руках, тим самим перевіривши його «в дії». Оскільки додаток Augment не призначений для якоїсь конкретної задачі, він дозволяє користувачам завантажувати і потім використовувати свої власні 3D моделі, що робить його ще універсальнішим, навідміну від конкурентів, в яких дана функція або відсутня або обмежена каталогами, які можна дозавантажити.

Порівняльна характеристика програм-аналогів наведена в таблиці 1.2.

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Таблиця 1.2. Порівняльна характеристика програм-аналогів

характеристика	IKEA	Augment	SmartReality
Наявність AR технології	+	+	+
Наявність VR технології	-	-	+
Необхідність маркера	+	-	+
Можливість розширити каталог моделей	+	+	+
Можливість переміщувати об'єкти вручну	+	+	-
Наявність під інші операційні системи	+	+	+

Проаналізувавши програми-аналоги, можна зробити висновок, що AR технологія є досить популярною в сфері інтер'єру та дизайну. Кожен з розглянутих додатків має якусь свою унікальну функцію і призначення. IKEA, це додаток від одноіменної меблевої компанії, за допомогою якого користувачі можуть переглядати нові колекції і «облаштовувати» власну оселю як їм заманеться, не докладаючи значних зусиль для цього. SmartReality це професійний інструмент для архітекторів, за допомогою якого можна реалізувати власний задум просто на папері і виявити власні помилки. Augment - це додаток який звичайно можна використовувати як розвагу, але водночас він може слугувати і помічником в вирішенні побутових проблем.

Висновки до розділу 1

В розділі 1 дано визначення поняття технології доповненої реальності та розглянута класифікація систем доповненої реальності. Також було розглянуто сучасні технології для роботи з AR та проведено їх детальний аналіз. В результаті бібліотека доповненої реальності ARToolkit виявилася найбільш оптимальною для вирішення технічного завдання дипломного проекту.

Також було розглянуто та порівняно мобільні операційні системи. Операційна система Android є на даний момент найпопулярнішою мобільною операційною системою, постійно оновлюється, має добре задокументовану документацію і безкоштовне середовище розробки Android Studio. Саме по цим причинам в якості базової платформи було обрано Android.

При розгляді програм-аналогів виявилось, що якісних додатків, які реалізують технологію доповненої реальності на сьогодні є досить багато. Проте в більшості із них все ж таки є свої недоліки. Основними з яких є недостатня реалістичність 3d моделей і сильна залежність від освітленості середовища, в якому відбувається використання додатку. Також до недоліків можна віднести шуми, які виникають під час розпізнавання маркерних зображень, через що часто 3d моделі мають властивість хаотичних зміщень, по відношенню до маркера.

					ІАЛЦ.467100.003 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ОПИС ЗАСОБІВ РОЗРОБКИ ТА РОБОТИ ДОДАТКУ

2.1 Постановка задачі дипломного проектування

Завданням на дипломне проектування є розробка мобільного додатку для візуалізації інтер'єру з використанням технології доповненої реальності(AR) під управлінням ОС Android. Клієнтський додаток повинен надавати користувачеві інформацію про доступні 3D моделі, мати зрозумілий та простий інтерфейс для здійснення маніпуляцій з 3D моделями, тобто переміщувати, повертати, масштабувати. При розробці додатку потрібно ознайомитись з існуючими методами роботи з технологією доповненої реальності, технологією JNI, роботою з камерою на ОС Android, роботою з багатопоточністю в Java.

2.2 Інструментальні засоби для реалізації

2.2.1 ARToolkit

ARToolKit - це бібліотека комп'ютерного відстеження з відкритим вихідним кодом для створення додатків з доповненою реальністю[13], які накладають віртуальні образи на реальний світ. В даний час він підтримується як проект з відкритим вихідним кодом, розміщений в GitHub. ARToolKit - дуже популярна бібліотека відстеження AR з більш ніж 160 000 завантажень в останній публічний реліз в 2016 році.

Щоб створити доповнену реальність, вона використовує можливості відстеження відео, які обчислюють реальну позицію і орієнтацію камери щодо квадратних фізичних маркерів або маркерів природних ознак в реальному часі. Як тільки реальна позиція камери відома, віртуальна камера може бути розташована в одній і тій же точці, а 3D-моделі комп'ютерної графіки намальовані точно накладаються на реальний маркер. Таким чином, ARToolKit вирішує дві ключові проблеми в розширеній реальності: відстеження точок огляду і взаємодія віртуальних об'єктів.

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

ARToolKit був спочатку розроблений Хироказу Като з Інституту науки і техніки Нарі в 1999 році і був випущений Вашингтонським університетом HIT Lab. У 2001 році був включений ARToolWorks, і версія 1.1 від версії ARToolKit з відкритим вихідним кодом була випущена через HIT Lab. ARToolKit був одним з перших AR SDK для мобільних пристроїв, який вперше був запущений на Symbian в 2005 році, потім IOS з iPhone 3G в 2008 році і, нарешті, Android вже в 2010 році з професійною версією ARToolWorks пізніше, в 2011 році.

ARToolKit був придбаний DAQRI і перевиданий з відкритим вихідним кодом починаючи з версії 5.2 13 травня 2015 года, включаючи всі функції, які раніше були доступні тільки в професійній ліцензованої версії. Серед цих функцій - мобільна підтримка і відстеження природних функцій.

Особливості:

1. Надійне відстеження, включаючи відстеження природних функцій
2. Сильна підтримка калібрування камери
3. Одночасне відстеження та підтримка стереозвуку
4. Підтримка декількох мов
5. Оптимізовано для мобільних пристроїв
6. Підтримка Full Unity3D і OpenSceneGraph
7. Одна з найскладніших частин розробки програми доповненої реальності - це точно розрахувати точку зору користувача в реальному часі, щоб віртуальні зображення були точно вирівняні з об'єктами реального світу. ARToolKit використовує методи комп'ютерного зору для розрахунку реального стану і орієнтації камери щодо квадратних фігур або плоских текстурованих поверхонь, що дозволяє програмісту накладати віртуальні об'єкти. ARToolKit в даний час підтримує класичний квадратний маркер, 2D-штрих-код, мультимаркер і природне відстеження функцій. Крім того, ARToolKit підтримує будь-яку комбінацію вищевказаного разом. Швидке і точне відстеження, що

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

надається ARToolKit, дозволило швидко розвинути тисячі нових і цікавих програм AR.

8. Документація містить повний опис бібліотеки ARToolKit, її установку і використання її функцій в додатках AR. Кілька простих прикладів додатків постачає ARToolKit, щоб програміст міг відразу почати роботу. ARToolKit включає бібліотеки відстеження і повні вихідні тексти для цих бібліотек, дозволяючи програмувати перенесення коду на різні платформи або налаштовувати його для власних додатків.
9. ARToolKit має набір інструментів, які працюють в операційних системах Windows, Mac OS X, Linux, IOS і Android. Функціональність кожної версії інструментарію однакова, але продуктивність може варіюватися в залежності від різних конфігурацій обладнання. ARToolKit можна легко перенести на інші нові і експериментальні платформи.
10. ARToolKit підтримує як відео, так і оптично-прозору доповнену реальність. Відео-перегляд AR - це те, де віртуальні зображення накладаються на живе відео реального світу. Альтернативою є оптична прозора доповнена реальність, де комп'ютерна графіка приходить безпосередньо на погляд реального світу. Оптична прозора реальна реальність зазвичай вимагає прозорого дисплея, встановленого на голові, і має більш складні вимоги до калібрування і реєстрації камери.



Рис. 2.1. Логотип ARToolkit

ARToolkit працює наступним чином:

1. Камера захоплює відео з камери і відправляє її на обробку.
2. Програмне забезпечення виконує пошук по кожному відеокадрі для будь-яких квадратних фігур (квадратні маркери).
3. Якщо знайдений квадратний маркер і вміст зображення, вбудоване в квадрат, шаблон, зіставляється та ідентифікується, програмне забезпечення використовує математику для обчислення щодо камери як положення чорного квадрата, так і орієнтації малюнка.
4. Як тільки стан і орієнтація камери відомі, модель комп'ютерної графіки малюється з використанням зсуву в розрахункове положення і з відповідною орієнтацією.
5. Ця модель малюється на передньому плані захопленого відео і відстежується проти рухів фоновому відео, змушуючи модель з'являтися на тлі.
6. Остаточний висновок відображається на дисплеї, тому, коли користувач переглядає переглядач, він бачить візуалізовану графічну модель над потоком реального світу;

На рис. 2.2 представлені ці кроки. ARToolKit здатний виконувати це спостереження в реальному часі, гарантуючи, що 3D моделі завжди показуються коректно.

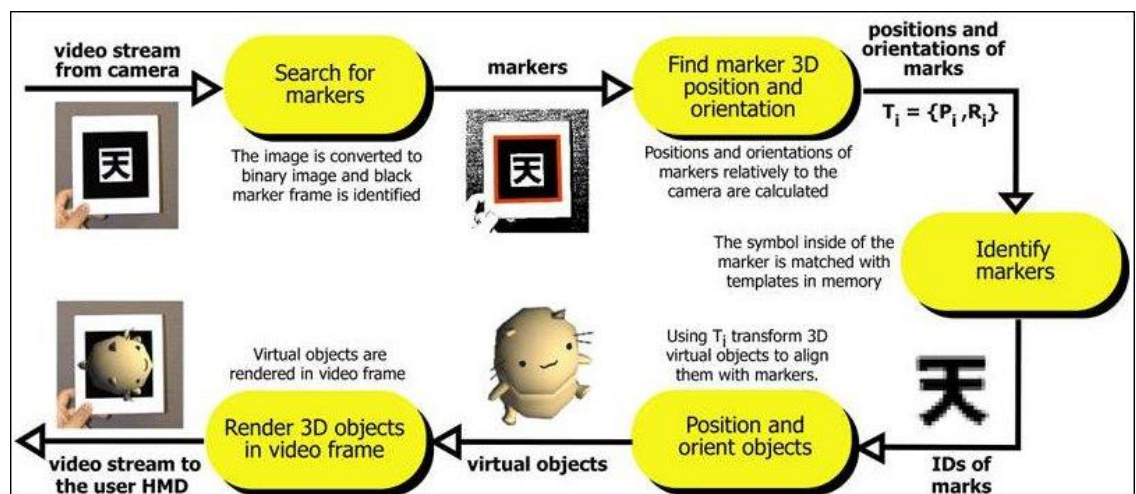


Рис. 2.2. Цикл роботи ARToolkit

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

25

2.2.2 Java

Java — це об'єктно-орієнтована мова програмування. Синтаксис мови багато в чому походить від C та C++ [14]. У офіційній реалізації, Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи. Oracle надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій java дозволяє викликати підпрограми, написані іншими мовами програмування.

Мова Java активно використовується для створення мобільних додатків під операційну систему Android. При цьому програми компілюється в нестандартних байтах-код, для використання їх віртуальна машина Dalvik (з Android починаючи 5,0 Lollipop віртуальної машини замінена на ART). Для такої компіляції використовується додатковий інструмент, а саме Android SDK (Software Development Kit), розроблений компанією Google.

2.2.3 Android SDK

Android SDK - універсальний засіб розробки мобільних додатків для операційної системи Android. Відмінною рисою від звичайних редакторів для написання кодів є наявність широких функціональних можливостей, що дозволяють запускати тестування і налагодження вихідних кодів, оцінювати роботу програми в режимі сумісності з різними версіями ОС Андроїд і спостерігати результат в реальному часі (опціонально). Підтримує велику

					ІАЛЦ.467100.003 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

кількість мобільних пристроїв, серед яких виділяють: мобільні телефони, планшетні комп'ютери, розумні окуляри (в тому числі Google Glass), сучасні автомобілі з бортовими комп'ютерами на ОС Андроїд, телевізори з розширеним функціоналом, особливі види наручних годинників і багато інших мобільні гаджети, габаритні технічні пристосування.

2.2.4 Android NDK

Android NDK (Android Native Development Kit) — необхідний набір інструментарію для розробки компонентів програмного забезпечення для платформи Android, який базується на C/C++ та інших мовах програмування. Містить в собі лімітований набір загальноживаних низькорівневих (нативних) бібліотек та API, написаних на C/C++ та інших мовах програмування, документацію і мінімальний набір прикладів для демонстрації базового функціоналу [15]. За допомогою NDK розробник застосунку для операційної системи Android може імплементувати окремі його частини, використовуючи такі мови, як C/C++, а не тільки Java. Це надає можливість використати деякі переваги, оскільки в окремих випадках код, написаний на C/C++, виконуватиметься швидше в порівнянні з кодом на Java. Android NDK може бути використаний для платформи Android 1.5 (API Level 3) і новіших версій.

Для зручності компіляції коду Android NDK містить спеціальний скрипт ndk-build, завданням якого є наступне:

- автоматичне визначення правил компіляції і файлу додатку для цільових бінарних файлів
- генерація бінарних файлів
- копіювання згенерованих бінарних файлів в потрібну директорію, визначену користувачем.

Враховуючи відмінності які існують між кодом написаним на C/C++ в порівнянні з кодом написаним на Java, розробниками Google рекомендовано використовувати Android NDK в наступних цілях:

					ІАЛЦ.467100.003 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

- пришвидшення розрахунків великих обчислень, таких як обробка сигналів, розрахунки для фізичних симуляцій, сортування та інше;
- використання функціоналу із сторонніх бібліотек написаних на C/C++, наприклад: OpenCV, OpenGL ES;
- програмування на низькому рівні, або у випадках коли Java не надає необхідного інструментарію.

2.2.5 Java Native Interface

Java Native Interface(JNI) - стандартний механізм для запуску коду, під керуванням віртуальної машини Java [16], який написаний на мовах C / C ++ або асемблері, і скомпонований у вигляді динамічних бібліотек, дозволяє не використовувати статичну зв'язування. Це дає можливість виклику функції C/C ++ з програми на Java, і навпаки.

Основна перевага перед аналогами (Native Interface Netscape Java Runtime інтерфейс або Microsoft, Raw і COM / Java Interface) є те що JNI спочатку розроблявся для забезпечення двійкової сумісності, для сумісності додатків, написаних на JNI, для будь-яких віртуальних машин Java на конкретній платформі (JNI не прив'язується до Dalvik машини, так як JNI був написаний Oracle для JVM ,який підходить для всіх віртуальних машин Java). Тому скомпільований код на C / C ++ буде виконуватися в не залежності від платформи. Більш ранні версії не дозволяли реалізовувати двійкову сумісність.

Двійкова сумісність або ж бінарна сумісність - вид сумісності програм, що дозволяє програмі працювати в різних середовищах без зміни її виконуваних файлів.

JNI таблиця, організована як таблиця віртуальних функцій в C ++. VM може працювати з декількома такими таблицями. Наприклад, одна буде для налагодження, друга для використання. Показчик на JNI інтерфейс дійсний тільки в поточному потоці. Це означає, що показчик не може гуляти з одного потоку в інший. Але нативні методи можуть бути викликані з різних потоків.

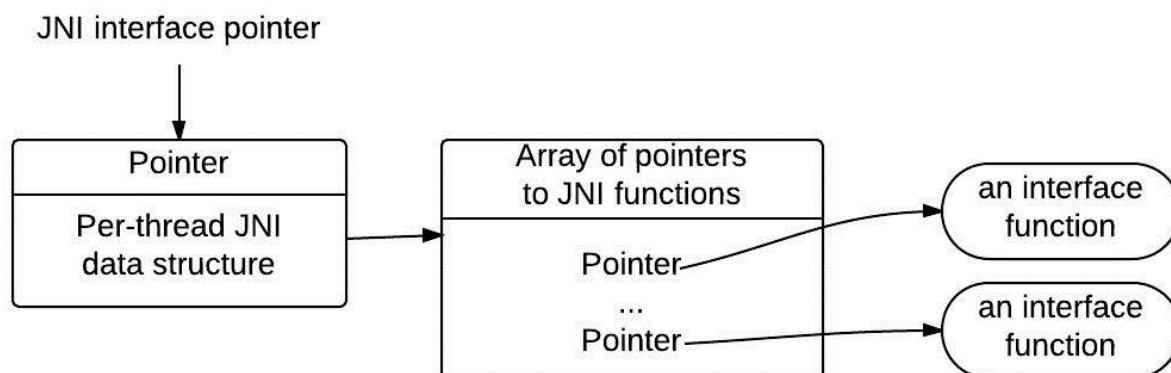


Рис. 2.3. Структура JNI

Примітивні типи (таблиця 2.3) копіюються між VM і нативним код, а об'єкти передаються по посиланню. VM зобов'язані відстежувати всі посилання які передаються в нативному коді. Весь передані посилання в нативному коді не може бути звільнена GC. Але нативний код в свою чергу повинен інформувати VM про те що йому більше не потрібні посилання на передані об'єкти.

Таблиця 3.3. Примітивні типи даних JNI

Java тип	Native тип	Опис
boolean	jboolean	unsigned 8 bits
byte	jbyte	signed 8 bits
char	jchar	unsigned 16 bits
short	jshort	signed 16 bits
int	jint	signed 32 bits
long	jlong	signed 64 bits
float	jfloat	32 bits
double	jdouble	64 bits
void	void	N/A

2.2.6 Android Studio

Android Studio — інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року на конференції Google I/O менеджером по продукції корпорації Google — Еллі Паверс (англ. Ellie Powers). 8 грудня 2014 року компанія Google випустила перший стабільний реліз Android Studio 1.0.

Android Studio прийшло на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі сирцевих текстів продукту IntelliJ IDEA Community Edition, що розвивається компанією JetBrains. Android Studio розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0.

Бінарні складання підготовлені для Linux (для тестування використаний Ubuntu), Mac OS X і Windows. Середовище надає засоби для розробки застосунків не тільки для смартфонів і планшетів, але і для носимих пристроїв на базі Android Wear, телевізорів (Android TV), окулярів Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для застосунків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android.[5] У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

					ІАЛЦ.467100.003 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Для прискорення розробки застосунків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компоновання, що надає зручний попередній перегляд різних станів інтерфейсу застосунку (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовище вбудовані функції завантаження типових прикладів коду з GitHub.

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.

2.2.7 Open GL

OpenGL (англ. Open Graphics Library — відкрита графічна бібліотека) — специфікація, що визначає незалежний від мови програмування крос-платформовий програмний інтерфейс (API) для написання застосунків, що використовують 2D та 3D комп'ютерну графіку [17]. Цей інтерфейс містить понад 250 функцій, які можуть використовуватися для малювання складних тривимірних сцен з простих примітивів. Широко застосовується індустрією комп'ютерних ігор і віртуальної реальності, у графічних інтерфейсах (Comriz, Clutter), при візуалізації наукових даних, в системах автоматизованого проектування тощо.

OpenGL орієнтується на такі два завдання: сховати складності адаптації різних 3D-прискорювачів, надаючи розробнику єдиний API і приховати відмінності в можливостях апаратних платформ, вимагаючи реалізації

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

відсутньої функціональності за допомогою програмної емуляції. Основним принципом роботи OpenGL є отримання наборів векторних графічних примітивів у вигляді точок, ліній та багатокутників з наступною математичною обробкою отриманих даних та побудовою растрової картини на екрані і/або в пам'яті. Векторні трансформації та растеризація виконуються графічним конвеєром (graphics pipeline), який власне являє собою дискретний автомат. Абсолютна більшість команд OpenGL потрапляють в одну з двох груп: або вони додають графічні примітиви на вхід в конвеєр, або конфігурують конвеєр на різне виконання трансформацій. OpenGL є низькорівневим процедурним API, що змушує програміста диктувати точну послідовність кроків, щоб побудувати результуючу растрову графіку (імперативний підхід). Це є основною відмінністю від дескрипторних підходів, коли вся сцена передається у вигляді структури даних (найчастіше дерева), яке обробляється і будується на екрані. З одного боку, імперативний підхід вимагає від програміста глибокого знання законів тривимірної графіки та математичних моделей, з іншого боку — дає свободу впровадження різних інновацій.

2.2.8 Формат .obj і .mtl

Тривимірна графіка (3D, 3 Dimensions) — розділ комп'ютерної графіки, сукупність прийомів та інструментів (як програмних, так і апаратних), призначених для зображення об'ємних об'єктів [18].

Існує досить багато форматів 3D об'єктів. Найпопулярнішими є такі: OBJ, 3DS, DXF, FBX, glTF, STL. В якості 3D моделей було обрано об'єкти формату OBJ.

OBJ (чи .OBJ) — формат файлу опису геометрії, вперше створений в Wavefront Technologies для їх анімаційного пакету Advanced Visualizer. Це відкритий файловий формат, і він був прийнятий іншими розробниками 3d редакторів, як стандартний.

Формат OBJ дуже простий, і задає тільки геометрію об'єкта, а якщо конкретно, то координати кожної вершини, її текстурні координати, нормалі, і грані, що задаються списками вершин многокутників. Вершини многокутників за замовчуванням задаються проти годинникової стрілки, роблячи явне задання нормалей необов'язковим.

Інформація про зовнішній вигляд об'єктів (матеріали) передаються в файлах-супутниках в форматі MTL (Material Library).

MTL це стандарт, встановлений компанією Wavefront Technologies. Вся інформація представлена в ASCII вигляді і абсолютно читабельна для людини. Стандарт MTL так само дуже популярний і підтримується більшістю пакетів для роботи з 3D-графікою.

2.3 Вхідні та вихідні дані

Вхідними даними буде зображення отримане з камери мобільного пристрою. Якщо дане зображення міститиме в собі зображення маркера, система проаналізує його і розрахує конфігурацію 3D моделі, що прив'язана саме до цього маркера.

Вихідними даними буде демонстрація 3D моделі, що пов'язана з маркерним зображенням, поверх вхідного зображення, у випадку успішного розпізнання маркера системою. Якщо ж система не розпізнає маркер то вихідними даними залишається зображення з камери.

Висновки до розділу 2

В розділі 2 описується постановка задачі дипломного проекту, яка заключається в розробці мобільного додатку для візуалізації інтер'єру з використанням технології доповненої реальності (AR) під управлінням ОС Android.

Більш детально описано інструментальні засоби для рішення завдання технічного проекту. Оскільки рендеринг 3d графіки потребує значних математичних обчислень, то для збільшення швидкодії було вирішено, що рендеринг необхідно проводити на мові програмування низького рівня. Щоб це реалізувати буде використано механізм JNI, який дозволяє виконувати частини коду C/C++ з Java класів і навпаки.

					ІАЛЦ.467100.003 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1 Опис математичної моделі

3.1.1 Геометричні перетворення в тривимірному просторі

У лінійній алгебрі, базис векторного простору розмірності n - це послідовність з n векторів $(\alpha_1, \alpha_2, \dots, \alpha_n)$ таких, що будь-який вектор простору може бути представлений єдиним чином у вигляді лінійної комбінації базисних векторів. При заданому базисі оператори представляються у вигляді квадратних матриць. Так як часто необхідно працювати з декількома базисами в одному і тому ж векторному просторі, необхідно мати правило перехлду координат векторів і операторів з базису в базис. Такий перехід здійснюється за допомогою матриці переходу.

Якщо вектори $\beta_1, \beta_2, \dots, \beta_n$ виражаються через вектори $\alpha_1, \alpha_2, \dots, \alpha_n$ як:

$$\beta_1 = a_{11} \alpha_1 + a_{12} \alpha_2 + \dots + a_{1n} \alpha_n$$

$$\beta_2 = a_{21} \alpha_1 + a_{22} \alpha_2 + \dots + a_{2n} \alpha_n$$

...

$$\beta_n = a_{n1} \alpha_1 + a_{n2} \alpha_2 + \dots + a_{nn} \alpha_n$$

то матрицею переходу від базиса $(\beta_1, \beta_2, \dots, \beta_n)$ до базиса $(\alpha_1, \alpha_2, \dots, \alpha_n)$ буде:

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

При множенні матриці, оберненої до матриці переходу, на стовпець, складений з коефіцієнтів розкладання вектора по базису $(\alpha_1, \alpha_2, \dots, \alpha_n)$ ми отримуємо той же вектор, виражений через базис $(\beta_1, \beta_2, \dots, \beta_n)$.

Матриці найбільш поширених перетворень в трьохвимірному просторі представлені в таблиці 3.1.

					ІАЛЦ.467100.003 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.3

<p>Масштабування</p> <p>де a, b і c - коефіцієнти масштабування відповідно по осях OX, OY і OZ:</p>	$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
<p>Поворот</p> <p>де φ - кут повороту зображення в двовимірному просторі</p>	<p>Відносно OX на кут φ</p> $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) & 0 \\ 0 & -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ <p>Відносно OY на кут μ</p> $\begin{bmatrix} \cos(\mu) & 0 & \sin(\mu) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\mu) & 0 & \cos(\mu) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ <p>Відносно OZ на кут Ω</p> $\begin{bmatrix} \cos(\Omega) & \sin(\Omega) & 0 & 0 \\ -\sin(\Omega) & \cos(\Omega) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
<p>Переміщення</p> <p>де a, b і c - зміщення відповідно по осях OX, OY і OZ.</p>	$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Застосовуючи дані матриці до об'єктів у тривимірному просторі можна виконати будь-яку трансформацію. Але якщо ми хочемо виконати декілька дій одночасно, наприклад повернути і перемістити, важливим є порядок застосування матриць. Загалом спочатку необхідно застосувати матрицю масштабування потім матрицю повороту і останньою застосовується матриця переміщення. Лише в такій послідовності трансформація виконається вірно.

3.1.2 Розрахунок параметрів матриці

Оскільки дисплей мобільного пристрою представляється площиною, а маніпуляції над 3d моделями необхідно виконувати в тривимірному просторі, постає проблема як, за допомогою даних, отриманих на площині розрахувати параметри для матриць переретворень.

При роботі з тривимірною графікою вісі OX, OY і OZ розташовані по ширині дисплею, по висоті дисплею і перпендикулярно до дисплею відповідно(рис. 3.1).

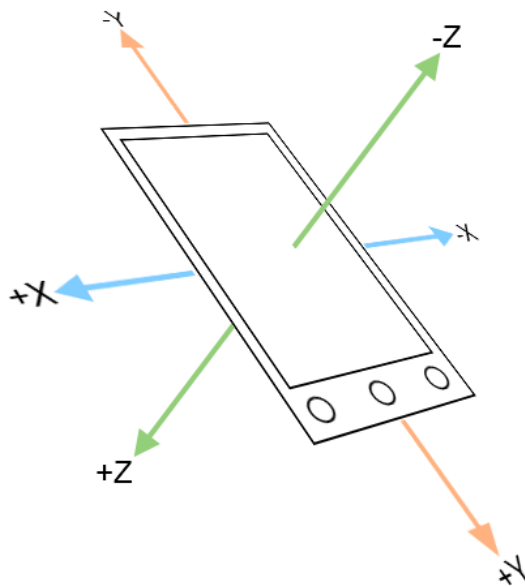


Рис. 3.1. Напрямок осей

Виконання повороту вздовж OX і OY. Нехай маємо вектор ON. Точка N має координатами $(M_1, M_2, 0)$. Вектор MN - одиничний вектор, що паралельний вісі OZ. З цього випливає, що вектор ON є проекцією вектора OM на площину OXY. Точка M має координати $(M_1, M_2, 1)$. Для знаходження кутів α і β необхідно знайти проекцію вектора OM на площини XOZ і YOZ відповідно. Після цього задача зводиться до розрахунку кутів між векторами на площині. Кут α можна виразити через визначення синуса з прямокутного трикутника M'_1OM_1 . Оскільки M'_1M_1 це паралельне перенесення вектора MN то модуль вектора M'_1M_1 обраховується як $|M'_1M_1| = \sqrt{M_1'^2}$. Модуль вектора M'_1O обраховується як $|M'_1O| = \sqrt{M_1'^2 + M_1^2}$. Отже $\sin(\alpha) = |M'_1M_1| / |M'_1O| \Rightarrow$

$\alpha = \arcsin(\sin(\alpha))$. Кут β можна виразитися також через синус з прямокутного трикутника M'_2OM_2 . Оскільки M'_2M_2 це паралельне перенесення вектора MN то модуль вектора M'_2M_2 обчислюється як $|M'_2M_2| = \sqrt{M_2'^2}$. Модуль вектора M'_2O обчислюється як $|M'_2O| = \sqrt{M_2'^2 + M_2^2}$. Отже $\sin(\beta) = |M'_2M_2| / |M'_2O| \Rightarrow \beta = \arcsin(\sin(\beta))$.

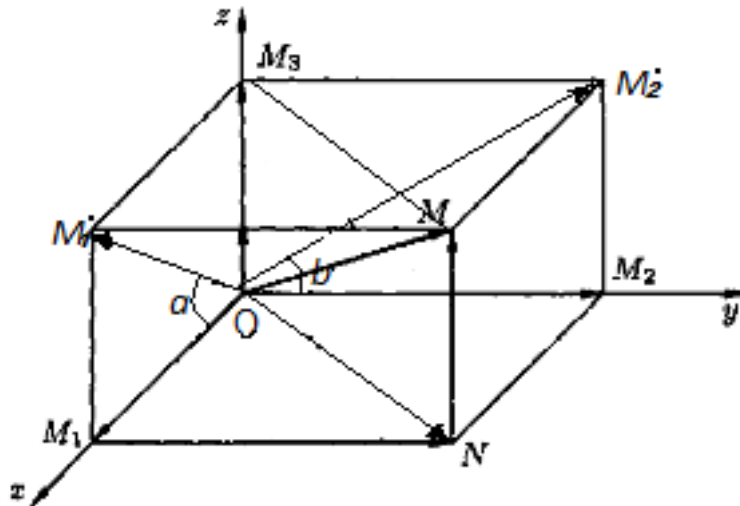


Рис. 3.2. Проекція вектора OM на вісі OX і OY

Знайдені кути α і β кути необхідно використати в матрицях перетворень типу повороту відносно OX і OY відповідно. В результаті чого 3D модель змінить орієнтацію в просторі на кут α відносно площини XOZ і на кут β відносно площини YOZ .

Виконання масштабування. Нехай маємо початковий вектор AB , де координати точок $A(x_a, y_a)$ і $B(x_b, y_b)$ і вектор $A'B'$ після масштабування, де координати точок $A'(x_{a1}, y_{a1})$ і $B'(x_{b1}, y_{b1})$. Щоб отримати коефіцієнт масштабування необхідно модуль різниці координат початкового вектора AB поділити на модуль різниці координат вектора $A'B'$: $s_x = |x_b - x_a| / |x_{b1} - x_{a1}|$, $s_y = |y_b - y_a| / |y_{b1} - y_{a1}|$, $s = \sqrt{s_x^2 + s_y^2}$.

3.2 Підготовка проекту

3.2.1 Інсталяція та налаштування Android Studio

Java SDK (JDK). Оскільки розробка додатків ведеться на Java, потрібно завантажити і встановити відповідне SDK, званий ще JDK(Java Development Kit). Завантажити його можна на офіційному сайті Oracle - <http://www.oracle.com>.

Середовище розробки + Android SDK. Зараз існує кілька середовищ розробки, я виберу рекомендовану Google - Android Studio. Завантажити її можна по адресу - <https://developer.android.com/studio/index.html>. Після завантаження необхідно запустити програму інсталяції. Далі вимагається вказати два шляхи. Перший шлях буде використаний для установки Android Studio. Другий - для установки Android SDK (рис. 3.3).

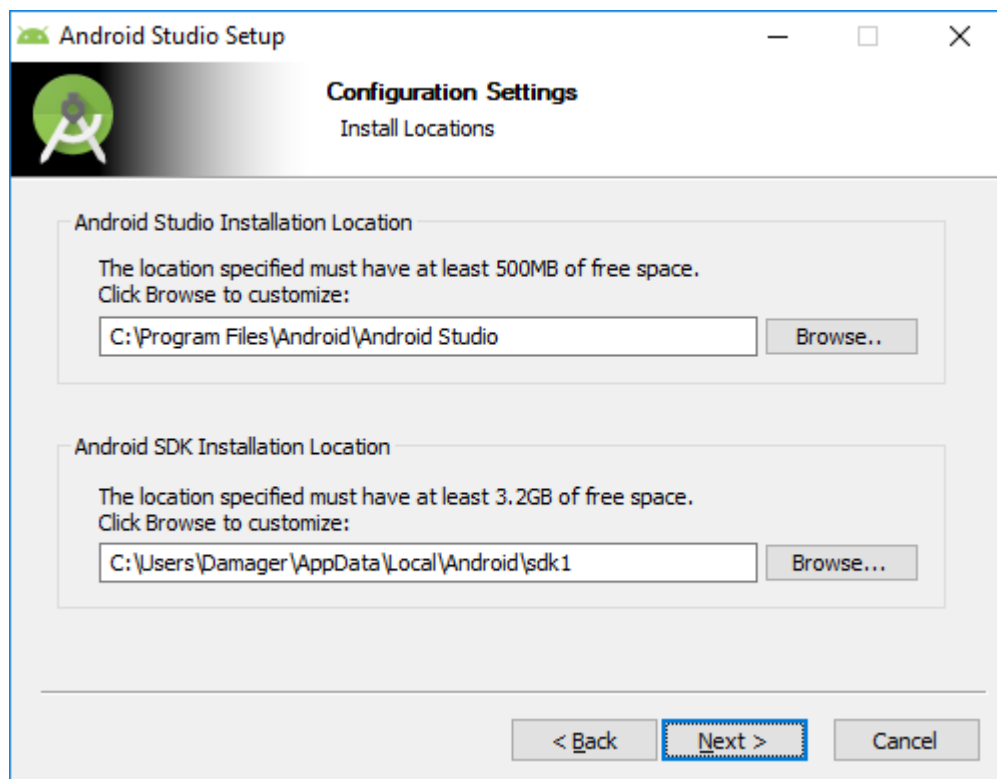


Рис. 3.3

Далі слідуючи підказкам інсталятора завершити встановлення. Під час установки необхідно буде з'єднання з інтернетом для завантаження Android SDK.

Для написання нативного коду C/C++ в Android необхідно завантажити Android NDK компонент. Для його завантаження необхідно відкрити Android SDK Manager і відмітити пункт “Android NDK” (рис. 3.4).

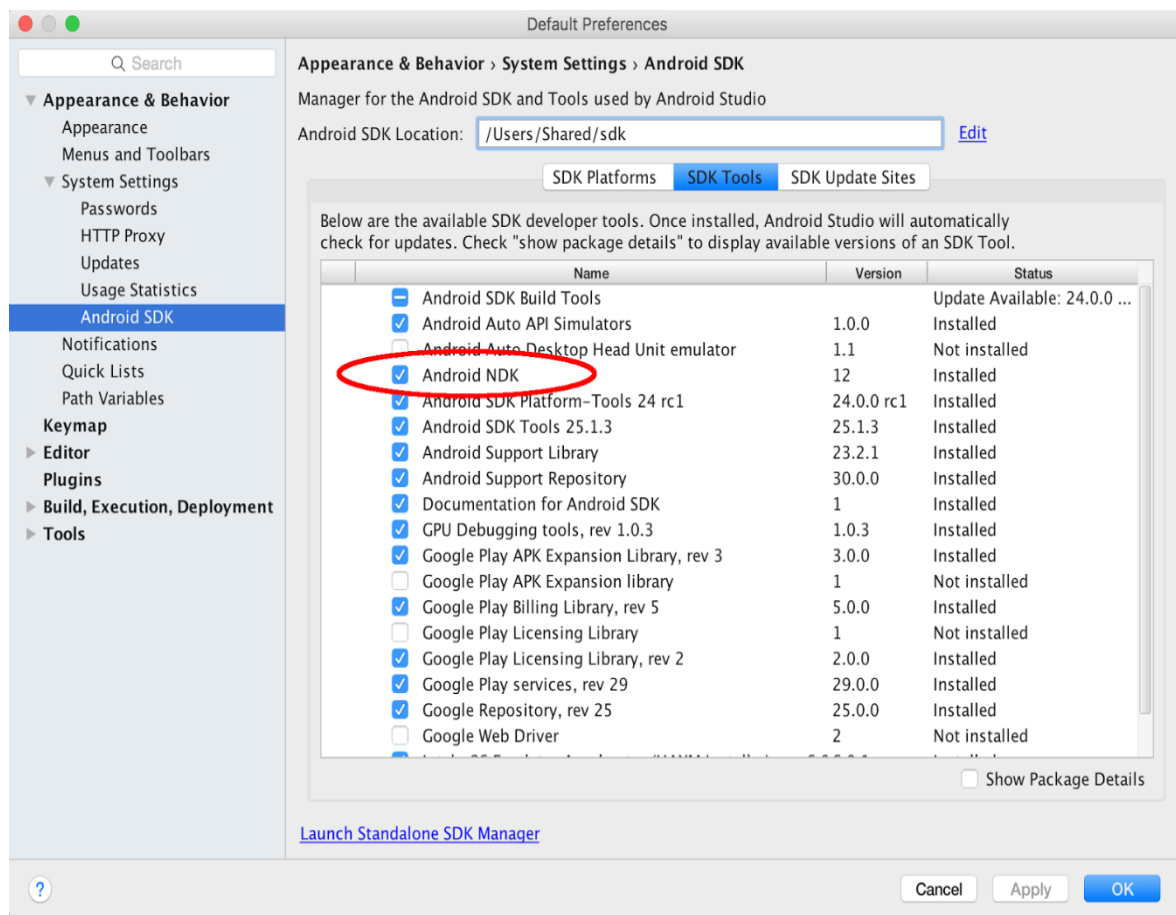


Рис. 3.4

Далі необхідно натиснути кнопку «ОК» для завантаження Android NDK компонентів. Для того щоб зміни вступили в силу, необхідно перезавантажити Android Studio. На цьому налаштування середовища розробки можна вважати завершеним.

3.2.2 Створення проекту та інтеграція JNI

Наступним кроком є створення проекту та інтеграція в нього JNI. Для створення проекту необхідно запустити Android Studio. У відкритому вікні вибрати «Start a new Android Studio project» (рис 3.5).

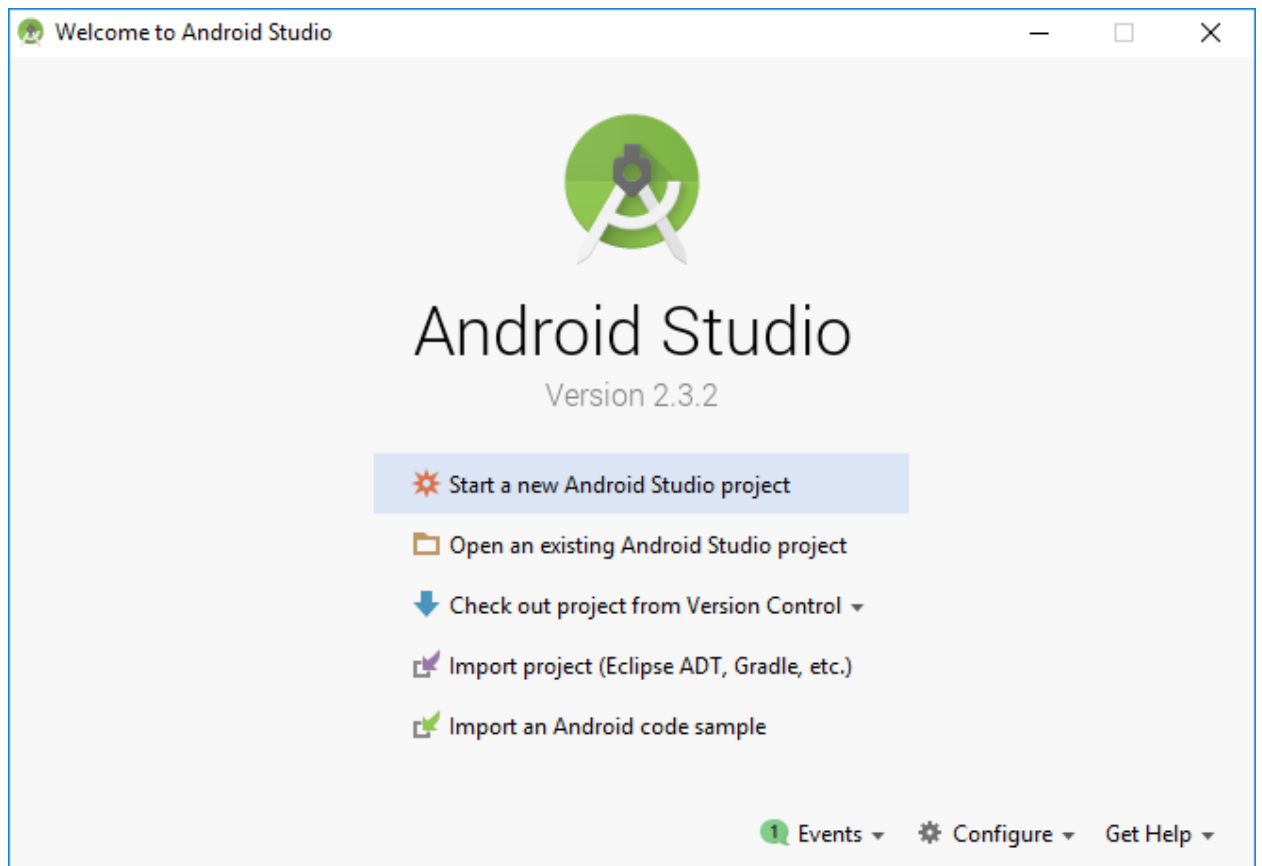


Рис. 3.5

Далі з'явиться вікно, в якому необхідно указати наступні дані: назву проекту, доменне ім'я компанії(використовується в якості кореневого пакета в Java) і вказати розташування проекту в файловій системі. Після цього натиснути кнопку «Next». В відкритому вікні необхідно вказати для яких типів пристроїв буде розроблятися додаток. Я виберу «Phone and Tabled», це значить, що додаток буде працювати лише на мобільних телефонах і планшетах. Ще необхідно вибрати мінімальну версію андроїд, для якої буде запускатися додаток. Я виберу «API 15, Android 4.0.3» (рис 3.6), це значить, на версіях Android нижче ніж 4.0.3 додаток працювати не буде. Хоча можна було б обрати і меншу версію, це буде недоцільно з огляду на те, що версії Android 4.0.3 і вище охоплюють понад 97% всього ринку Android пристроїв на сьогодні, а ще вагомою проблемою стало б те, що для підтримки старіших версій, довелося б відмовитися від деякої функціональності введеної в нових версіях

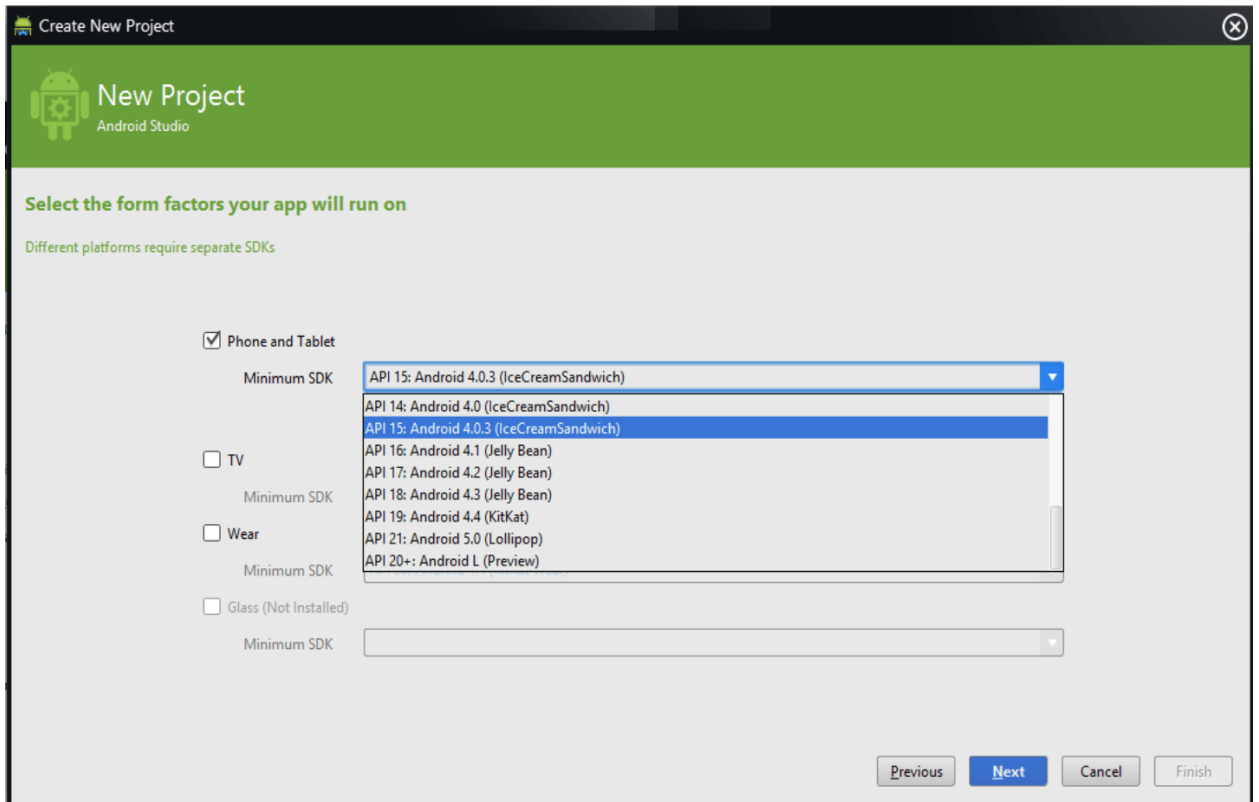


Рис. 3.6

Після цього необхідно вибрати шаблон головної активності(activity). Я обираю Empty Activity – це самий базовий шаблон, на якому відсутні будь які view-компоненти. Далі необхідно ввести ім'я головної активності і натиснути кнопку “Finish”. Після цього система збірки Gradle запустить автоматичну збірку проекту і по завершенню відкриється основне вікно Android Studio. На цьому створення можна вважати завершеним.

Далі необхідно додати підтримку нативного коду необхідно виконати наступні кроки:

1. У корені проекту потрібно створити папку JNI, в яку помістити файли нативного коду;
2. Створити файл Android.mk, який буде збирати проект;
3. Створити файл Application.mk, в якому описуються деталі збірки. Він не є обов'язковою умовою, але дозволяє гнучко налаштувати збірку;
4. Створити файл ndk-build, який буде запускати процес збірки (не є обов'язковим).

Загалом структура проекту повинна бути як на рис. 3.7.

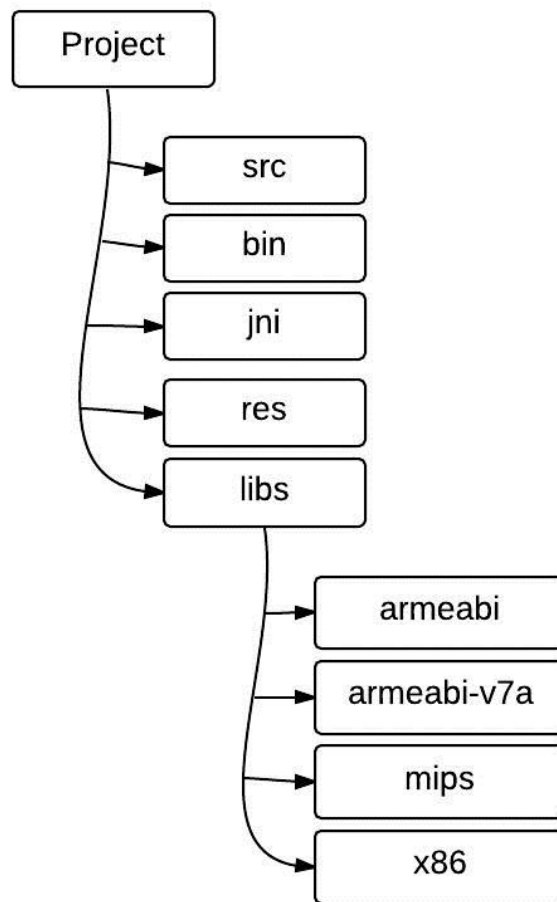


Рис. 3.7. Структура проекту

Весь нативний код має знаходитися в папці `jni`. Після складання проекту, в папці `libs` створиться чотири папки під кожен архітектур процесора, в якій буде лежати нативна бібліотека.

3.2.3 Налаштування ARToolkit

Бібліотеку Artoolkit для роботи з доповненою реальністю потрібно завантажити з офіційного сайту artoolkit.org. Далі у вкладці «Downloads» необхідно вибрати «ARToolkit for Android» і дочекатись завантаження. В завантаженому архіві є сама бібліотека, документація по ній, приклади проектів для Android, готові маркерні файли. Для інтеграції Artoolkit в додаток необхідно скопіювати файли бібліотек `libARWrapper.so` і `libc++_shared.so`, які знаходяться в завантаженій папці ARToolkit/libs, в кожен папку архітектур процесора.

3.2.4 Налаштування манифест файлу

Файл манифесту AndroidManifest.xml надає основну інформацію про додаток системі Android. Для доступу до спеціальних можливостей пристрою необхідно прописати відповідні дозволи. Для використання камери, доступу в мережу інтернет та можливості запису в файлову систему необхідно в файлі манифесту прописати наступне:

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

3.3 Архітектура програмного забезпечення

3.3.1 MVC

Існує декілька популярних рішень архітектури додатку під Android. Найпоширенішими є три: MVC, MVP і MVVM.

В якості архітектурного шаблону було вибрано паттерн MVC, так як він є одним з найпопулярніших шаблонів і добре вписується в концепцію розробки під OS Android.

Модель-представлення-контролер (англ. Model-view-controller, MVC) — архітектурний шаблон [19], який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів

					ІАЛЦ.467100.003 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

У рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами (рис 3.8). Модель (Model) відповідає за зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача і передає дані у модель.

Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.

Вигляд може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації.

Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

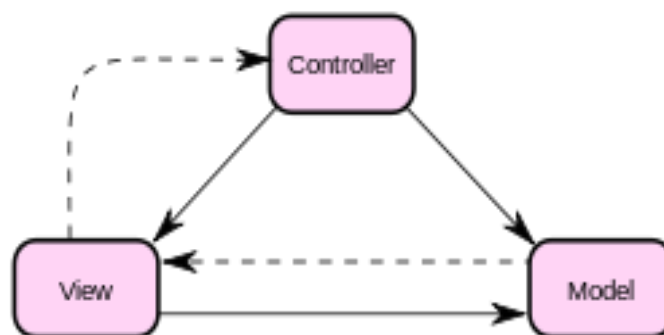


Рис. 3.8. Структура MVC

При програмуванні під Android паттерн MVC виражається наступними компонентами. Контролером є activity або fragment. Представленням є xml файли і класи що наслідують клас View. Моделлю виступає написана бізнес-логіка додатку.

3.3.2 Модель

Модель інкапсулює ядро даних і основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу даних. До категорії моделі відносяться наступні Java класи:

Object3D - клас моделі, що зберігає інформацію про 3D модель;

ObjectsHelper – клас для зберігання і керування списком 3D моделей. Реалізує шаблон програмування Singleton.

ModelsListAdapter – клас-адаптер що зв'яже інформацію про 3D модель із елементом списку. Використовується при виводі всіх доступних 3D моделей в системі.

Native - клас, що визначає інтерфейс функцій при роботі з нативним кодом. Особливістю є те, що в даному класі відбувається статичне підключення C/C++ бібліотек, а також всі функції оголошені із зарезервованим словом словом «native», що вказує на точку входу в JNI.

NativeRenderer - реалізує функції для рендеринга тривимірної графіки.

Діаграма класів моделі зображена на рис. 3.9.

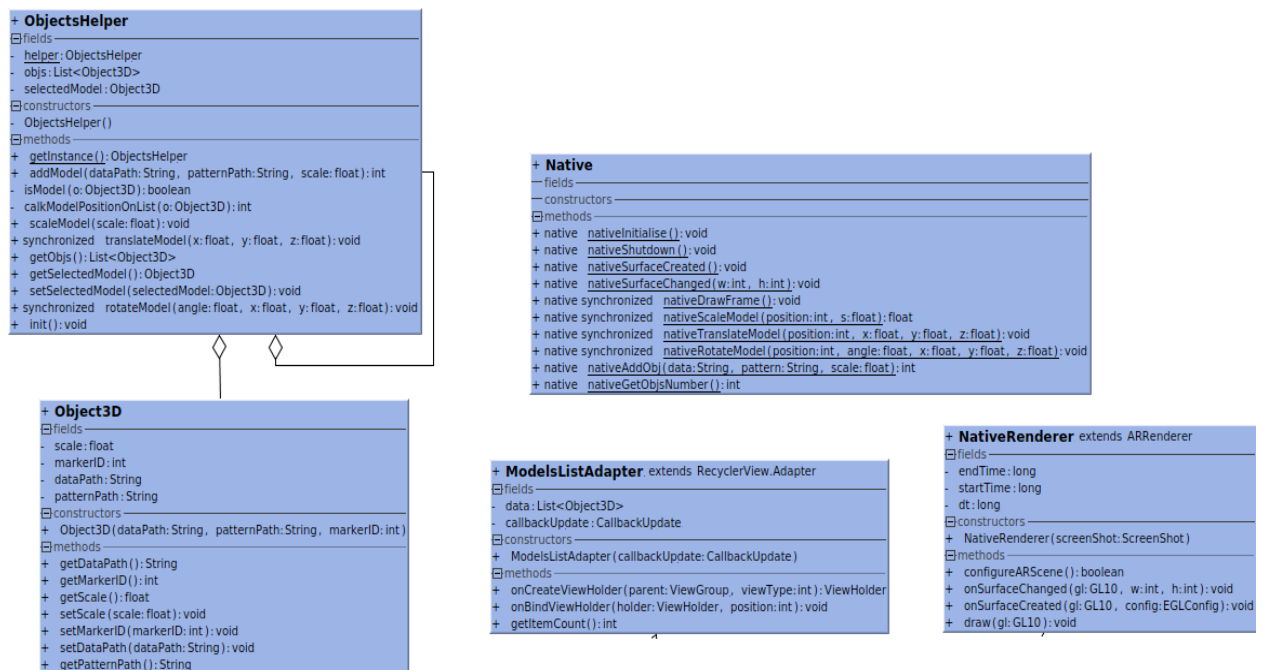


Рис. 3.9. Діаграма класів моделі

3.3.3 Контролер

У функції контролера входить відстеження визначених подій, що виникають в результаті дій користувача. Контролер дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас.

До категорії контролера відносяться наступні Java класи:

ARMainActivity – клас, що наслідує клас **Activity**. **Activity** - це компонент додатка, який видає екран з яким користувачі можуть взаємодіяти для виконання будь-яких дій. **ARMainActivity** зв'язує дані з відображенням.

SettingsActivity – клас, що наслідує клас **Activity**. Використовується для вибору типу камери і розширення камери.

ControllModelDialog - клас, що наслідує клас **DialogFragment**. Використовується для трансформації 3D моделей : масштабування, переміщення і поворот.

ModelsListDialog - клас, що наслідує клас **DialogFragment**. Використовується для відображення доступних 3D моделей у вигляді списку.

OnTouчController – клас, що використовується для розпізнавання жестів на екрані девайса. Реалізує функцію масштабування і повороту.

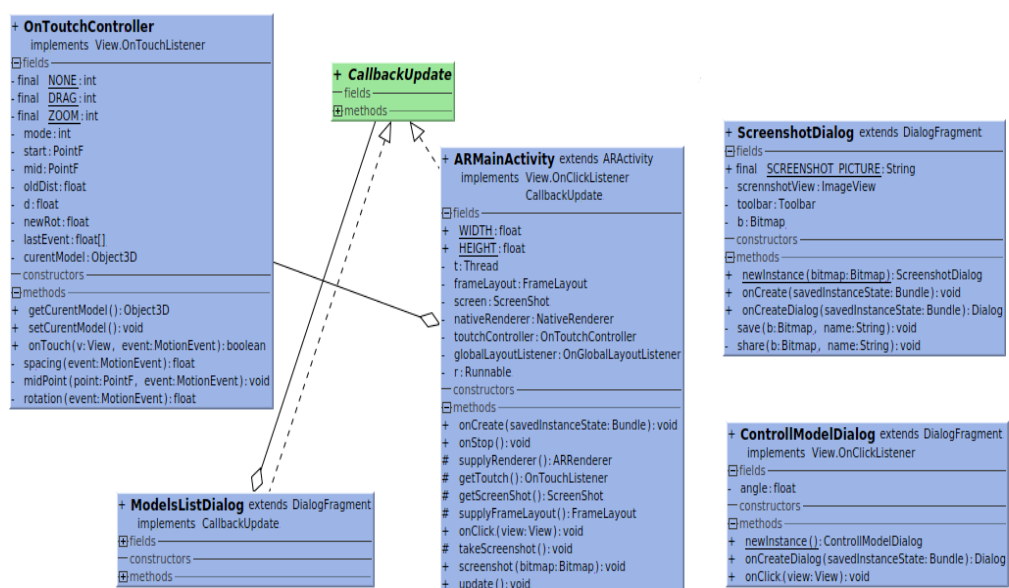


Рис. 3.10. Діаграма класів контролера

ScreenshotDialog - клас, що наслідує клас DialogFragment. Використовується для демонстрації виконаного знімку екрану. Реалізує функціонал збереження зображення в файловій системі і можливість поділитися зображенням у соціальних мережах.

3.3.4 Представлення

Операційна система Android дає можливість визначати представлення за допомогою xml розмітки. Тому зазвичай компонент представлення описується в xml файлах, що знаходяться в ресурсах проекту. Для кожної активності необхідно визначити представлення або xml форматі або за допомогою класів, що наслідують клас View з пакета android.widget або його нащадків. В даному проекті визначені наступні представлення:

main.xml - визначає представлення класу контролера ARMainActivity. Дане представлення доповнюється компоненти SurfaceView і GL SurfaceView для відображення зображення з камери і рендерингу 3D графіки відповідно, які додаються динамічно, в Java коді. Також на представленні розміщені view-компоненти, які реалізують меню додатку.

fragment_controll_model.xml - визначає представлення класу контролера ControllModelDialog. Розміщені view-компоненти, які реалізують основні функції трансформації моделей.

fragment_models_list.xml - визначає представлення класу контролера ModelsListDialog. Розміщений компонент RecyclerView, який дозволяє відображати список елементів.

fragment_screenshot.xml - визначає представлення класу контролера ScreenshotDialog. Розміщені view-компоненти, які реалізують основні функції при роботі зі скріншотом.

model_item.xml – визначає вид компонентів списку, який відображається в ModelsListDialog.

screenshot_dialog_menu.xml – файл ресурса типу menu, який реалізує відображення меню на Android ToolBar.

3.3.5 C/C++ функціонал

До категорії моделі також слід віднести C/C++ файл - Native.cpp. Він знаходяться в папці jni проекту. Даний файл визначає інтерфейс і реалізує цей інтерфейс в C/C++ коді з використанням JNI технології.

Загалом функцію JNI в C/C++ коді можна представити в наступному вигляді:

```
jtype Java_package_class_method(JNIEnv *env, jobject obj, args) {  
    //Тіло метода  
    return jtype;  
}
```

Написання JNI функцій має свої особливості, яких необхідно дотримуватися.

Функція прив'язана до Java класу, в якому вона буде викликатись. Назва функції формується наступним чином: спочатку пишеться ключовк слово «Java», слід відмітити, що регістр символів не має значення, далі ставиться нижнє підкреслювання після чого вказується java пакет і клас розділені нижнім підкресленням (в даному випадку регістр символів має значення).

Функція приймає два обов'язкових параметра: *env – вказівник на JNI інтерфейс; obj – посилання на об'єкт в якому описаний нативний метод. Інші args параметри є параметрами, які передаються з java коду, вони можуть бути відсутні. Типи параметрів визначає JNI інтерфейс, який має власні типи даних, що мають аналогії в Java. Примітивні типи даних JNI див в таблиці 2.3.

Native.cpp має наступний інтерфейс:

void nativeSurfaceCreated(JNIEnv* env, jobject object) - викликається при ініціалізації 3D моделей і створенні сцени.

void nativeDrawFrame(JNIEnv* env, jobject obj) - викликається для оновлення сцени.

void nativeShutdown(JNIEnv* env, jobject object) - викликається при завершенні рендеринку 3D моделей для звільнення ресурсів.

`jint nativeAddObj(JNIEnv* env, jobject object, jstring path, jstring pattern)`- використовується для додавання нових 3D моделей в систему. Параметри: `path`- шлях до .obj файла; `pattern` – назва файла маркера. Повертає значення ідентифікатора моделі.

`void nativeRotateModel(JNIEnv* env, jobject object, jint position, jfloat angle, jfloat x, jfloat y, jfloat z)` - використовується для поворота 3D моделі в просторі, яка знаходиться по індексу `position` в списку моделей, навколо вектора (x,y,z) на кут `angle`.

`void nativeTranslateModel(JNIEnv* env, jobject object, jint position, jfloat x, jfloat y, jfloat z)` - використовується для переміщення 3D моделі в просторі, яка знаходиться по індексу `position` в списку моделей, на відстань `x` по OX, на відстань `y` по OY, на відстань `z` по OZ.

`nativeScaleModel(JNIEnv* env, jobject object, jint position, jfloat s)` - використовується для масштабування 3D моделі в просторі, яка знаходиться по індексу `position` в списку моделей, на коефіцієнт `s`.

3.4 Інтеграція ресурсів в додаток

Файли ресурсів в Android проекті можуть міститися в двох директоріях: `res` і `assets`(активи). Обидва дуже схожі. Головна відмінність між ними полягає в тому, що в каталозі `res` кожному файлу призначається попередньо скомпільований ID, який генерується класом `R.java`, і отже може бути легко доступний через `R.id. [ID]`. Це корисно, щоб швидко і легко отримати доступ ресурсів.

Каталог активів більше схожа на файлову систему і забезпечує більшу свободу, щоб помістити будь-який файл, який би ви хотіли. Каталог активів дозволяє створювати дочірні каталоги, чого не дозволяє робити `res` каталог. Таким чином активи дозволяють вибудувати ієрархію файлів, яка необхідна. Ви можете отримати доступ до кожного з файлів в цій системі, як при доступі до будь-якого файлу в будь-якій файловій системі за допомогою Java. Також активи доступні для використання тільки при необхідності, на відміну від даних `res` каталогу, які завантажуються при запуску додатку.

					ІАЛЦ.467100.003 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4.1 Завантаження 3D моделей

Додавання 3D моделей до проекту буде відбуватися до каталогу активів. Так було вирішено по тій причині, що OBJ файли можуть містити суміжні MTL файли, а ті в свою чергу містити файли текстур, зображень і інші додаткові файли. Всі ці файли буде доцільніше використовувати у власному каталозі, що дозволяють активи навідміну від папки res, де можливість створити власний каталог відсутня.

3.4.2 Генерація ARToolkit маркера

ARToolkit підтримує декілька типів маркерів: квадратний маркер (square marker), QR штрих-код і мультимаркер. Я буду використовувати квадратний маркер. Квадратний маркер складається зі світлого, навколишнього, набивання, зазвичай білого кольору, всередині якого розміщена темна рамка, зазвичай чорного кольору і вбудованого висококонтрастного зображення, званого шаблоном (рис. 3.11). Шаблон - це те, що робить квадратний маркер унікальним. Квадратні маркери розпізнаються, відстежуються і використовуються для розрахунку положення в тривимірному просторі.



Рис. 3.11. Квадратний маркер Hiro

Квадратні маркери мають лише кілька обмежень. Вони повинні бути квадратними. У них повинна бути суцільна межа (зазвичай або повна чорна, або чисто біла). Із маркером на передньому плані, фон повинен бути контрастного кольору (як правило, темний або світлий колір або відтінок). За замовчуванням ширина кордону становить 25% від довжини краю маркера. Останнє обмеження полягає в тому, що область всередині кордону, яку ми називаємо шаблоном, повинна бути несиметричною по відношенню до обертання. Область всередині кордону може бути чорно-білою або кольоровою (ARToolKit забезпечує засіб для відстеження з більшою точністю, коли колір маркера забарвлений).

Розробка квадратного маркера. Новий маркер можна створити, відредагувавши файл зображення шаблону маркера, представлений в файлі ARToolKit SDK. Маркери можна масштабувати до будь-якого розміру і розміщувати в будь-якому місці цільової сцени. Утиліта ARToolKit використовується для створення файлу даних, який визначає розмір маркера, а також інші атрибути маркера.

Внутрішні 50% маркера інтерпретується як зображення шаблону ARToolKit, як показано на (рис. 3.12). Зображення шаблону може бути кольорове, біле на чорному або чорне на білому, і воно може сягати в області кордону. Частина зображення за межами внутрішніх 50% буде проігнорована ARToolKit, хоча і є необхідною, інакше ARToolKit може взагалі не розпізнати маркер, коли він знаходиться під дуже похилим кутом до камери.

Навчання ARToolKit для розпізнавання маркерів. Після розробки дизайну квадратного маркера ARToolKit повинен бути навчений розпізнавати його. Результатом навчання є файл з префіксом PATT., званий «файл-шаблон» маркера. Файли шаблонів дозволяють ARToolKit виявляти, розпізнавати, ідентифікувати і відстежувати нові маркери в захопленому відео потоці. Крім того, ARToolKit SDK може підтримувати розширення імен файлів «.patt» («.patt» в якості суфікса імені файлу).

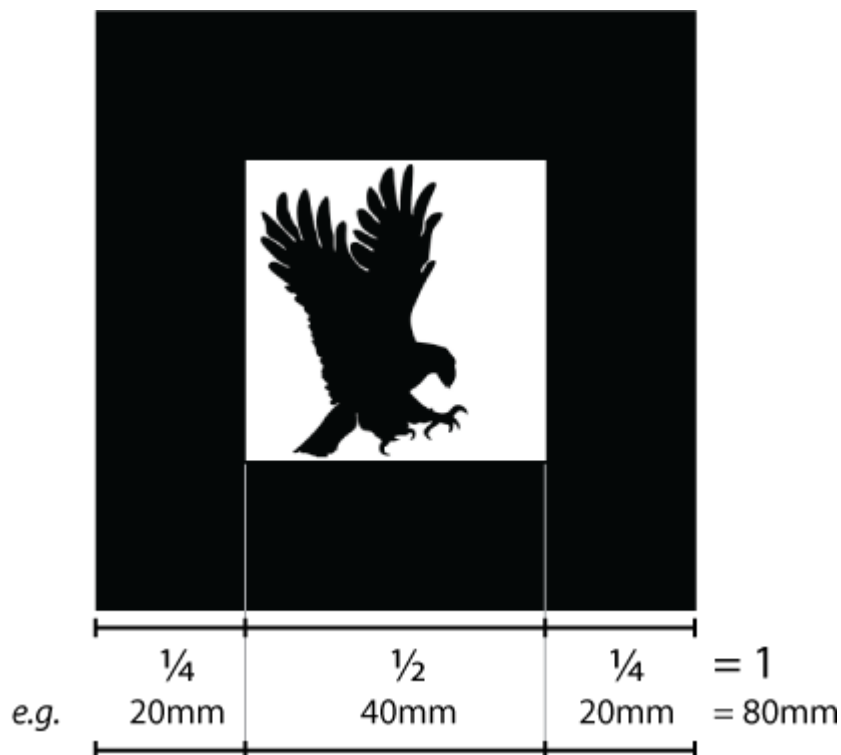


Рис. 3.12. Розмітка маркера

Навчання виконується з використанням утиліти `mk_patt`, яку можна в ARToolKit SDK. Або, альтернативно, навчання може бути виконано з використанням онлайн додатку на основі Adobe Flash: «Таротаро». Я буду використовувати Adobe Flash додаток (рис 3.13):

Для генерації необхідно виконати наступні кроки:

1. Створити оригінальні маркери і роздрукувати їх.
2. Відкрити ARToolKit Маркер Генератор.
3. Встановити сегменти і розміри маркера.
4. Навести веб-камеру на надрукований маркер.
5. Натиснути кнопку «Отримати шаблон», коли червона лінія включає маркери. І перейти в режим збереження.
6. Коли запускається «режим збереження», з'являється вікно попереднього перегляду.
7. Червоні квадрати показують всі виявлені маркери.
8. Зелений квадрат показує маркер в попередньому перегляді.
9. Вікно попереднього перегляду має 6 кнопок.

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

53

10. Кнопка «Пред. / Наступ. », щоб змінити маркер.
11. Кнопка «Видалити», щоб виключити поточний маркер попереднього перегляду з мети.
12. Кнопка «Зберегти все», щоб зберегти весь файл шаблону «* .zip».
13. Кнопка «Зберегти поточне», щоб зберегти файл шаблону «* .pat».
14. Кнопка «Скасування» або закрийте вікно, щоб завершити «режим збереження».



Рис. 3.13. Інтерфейс додатку «Таротаро»

Після завершення генерації зберений шаблон помістити в каталог assets.

3.5 Збірка проекту

3.5.1 Збірка нативної бібліотеки

Для збірки C/C++ бібліотеки необхідно запустити команду `ndk-build` з консолі Android Studio. Для коректного виконання в папці JNI проекту необхідно створити make-файли `Android.mk` і `Application.mk`, слід зауважити, що останній не є обов'язковим.

Android.mk це make файл для збірки нативного проекту. Android.mk дозволяє групувати ваш код в модулі. Модулі можуть бути як статичні бібліотеки (static library, тільки вони будуть скопійовані в ваш проект, в папку libs), колективні бібліотеки (shared library), автономний виконуваний файл (standalone executable).

В Application.mk файлі описується кілька змінних, які допоможуть зробити збірку більш гнучкою.

Після запуску збірки і успішного її завершення в папці libs проекту під кожен архітектуру процесора буде згенерований файл одноіменний з C/C++ класом, поверх якого відбувається збірка, з префіксом lib- та розширенням .so, в даному випадку згенерувався файл *libNative.so*.

3.5.2 Генерація APK

По замовчувню в AndroidStudio використовується автоматизована система збірки проектів Gradle. Щоб запустити збірку проекту необхідно виконати команду *gradle-build*. Після цього згенерований файл (рис. 3.14) можна буде знайти в директорії build проекту.

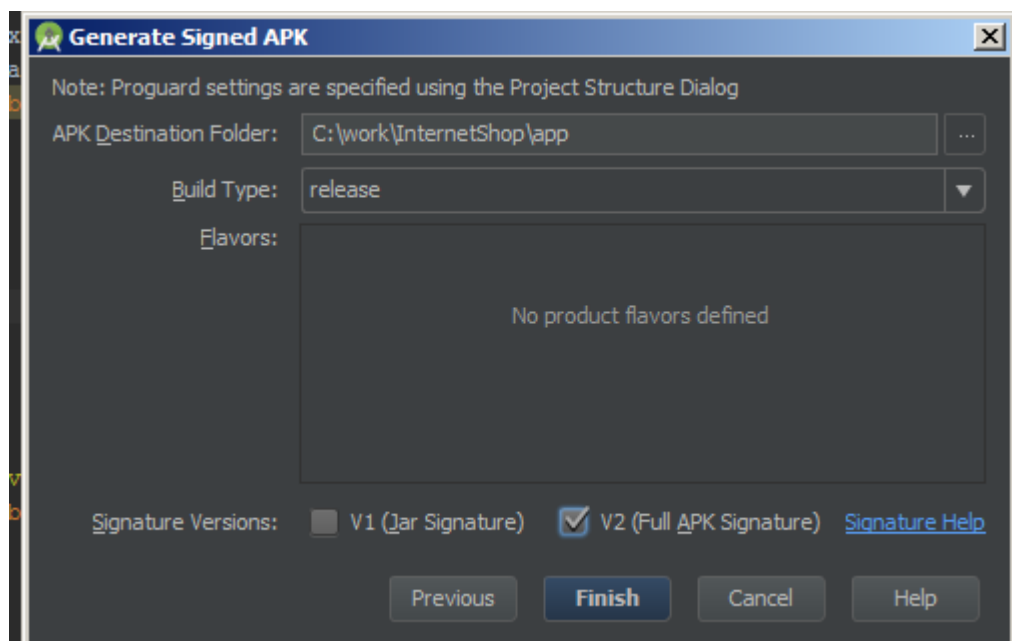


Рис. 3.14. Завершення створення APK-файлу

3.6 Опис реалізованого програмного забезпечення

Після запуску додатку з'являється головне вікно, на якому знаходиться View елемент для відображення потоку даних з камери. Праворуч знаходиться панель керування на якій розташовані 3 кнопки. Зверху розташована панель на якій знаходиться кнопка налаштувань (рис.3.15).



Рис. 3.15. Інтерфейс додатку:

- 1 – кнопка «Screenshot»;
- 2 – кнопка «List»;
- 3 – кнопка «Controll»;
- 4 – кнопка «Settings».

Оскільки для конфігурації 3D моделей і камери необхідний певний час, додаток при запуску відоразить повідомленн-Toast про готовність до роботи, вивівши нформацію про камеру та FPS.

Для демонстрації роботи додатку необхідно навести девайс на раніше заготовлене маркерне зображення, щоб камера захопи його. Після успішного розпізнавання і сканування зображення з камери, поверх маркера «будується» 3D модель (рис.3.16), що пов'язана з розпізнаним раніше маркером. На

успішність розпізнання додатком маркера можуть вплинути наступні фактори: погане освітлення, некоректно зображений маркер, погана якість камери тощо.

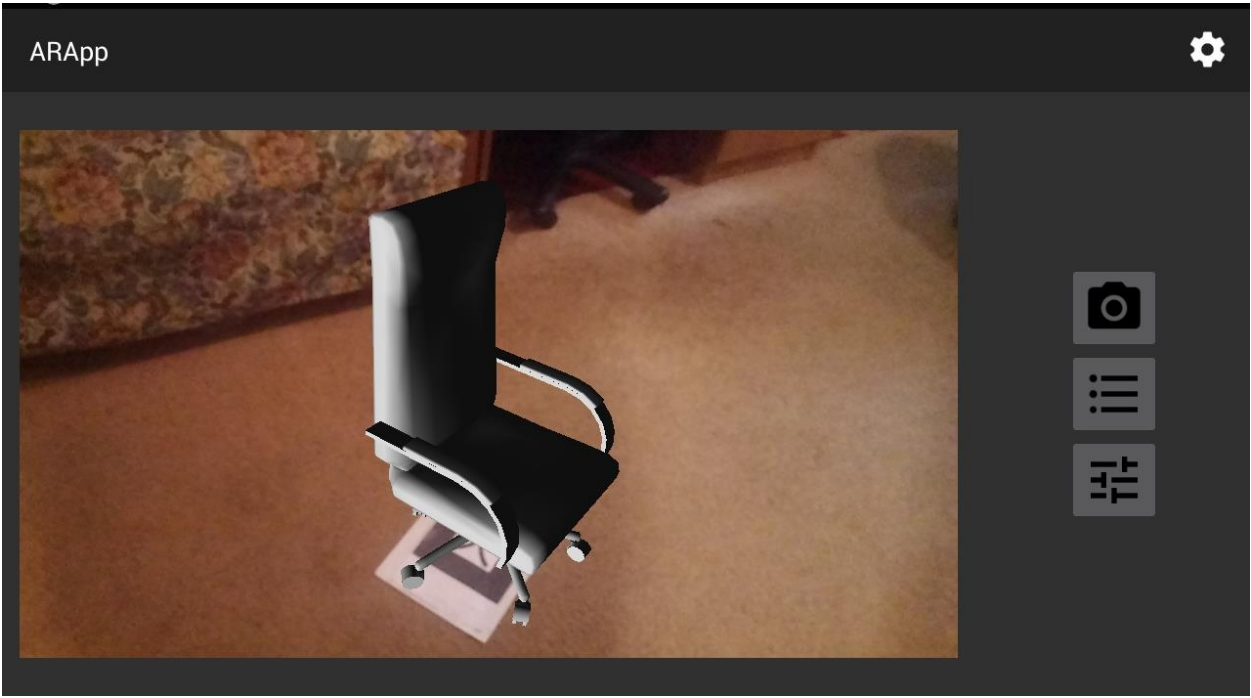


Рис. 3.16. 3D модель крісла поверх маркера

Для демонстрації доповненої реальності можна змінити кут огляду маркера або відстань до нього. В першому випадку 3D модель повернеться на відповідний кут, в другому – вона зміниться в розмірах

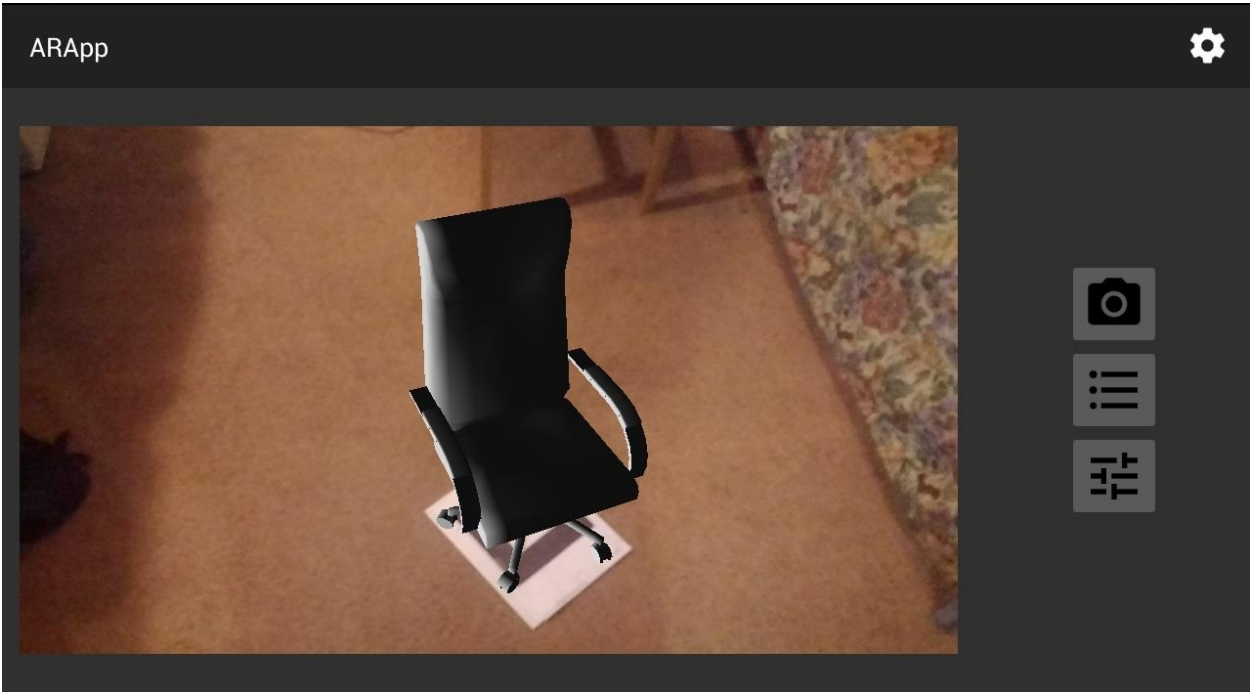


Рис. 3.17. 3D модель крісла з іншого ракурсу

Змн.	Арк.	№ докум.	Підпис	Дата

Якщо натиснути кнопку «ScreenShot» на якій зображена камера, на панелі інструментів, то додаток зробить знімок з камери з накладеним поверх нього 3D об'єктом (рис. 3.18). Фінальне зображення буде показане в відповідному діалоговому вікні.

Отримане зображення можна зберегти в пам'ять пристрою в форматі .png натиснувши кнопку «Save».

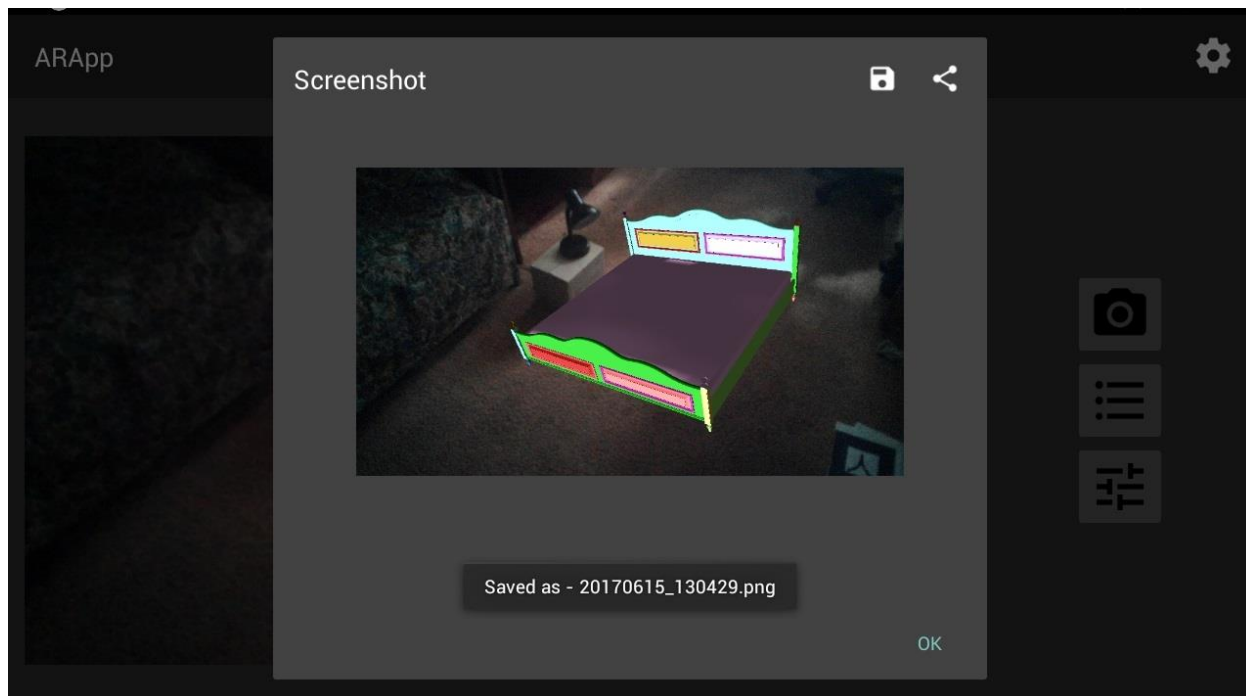


Рис. 3.18. Знімок екрану

Також отриманим зображенням можна поділитись з друзями в соціальних мережах натиснувши кнопку «Share» (рис. 3.19).

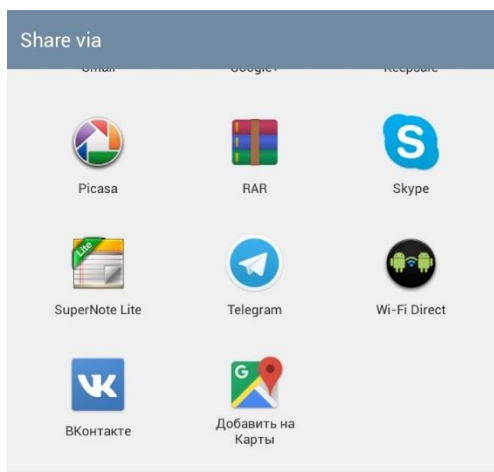


Рис. 3.19

Якщо натиснути кнопку «List» на панелі керування, відкриється діалогове вікно зі списком всіх доступних 3D моделей в системі (рис. 3.20). Поряд з назвою 3D моделі відображається пов'язаний з нею маркер Поточна модель відмічена відповідним знаком. Саме з цією моделлю юзер може виконувати всі маніпуляції трансформації. Також білим кольором підсвічуються всі видимі моделі на даний момент.

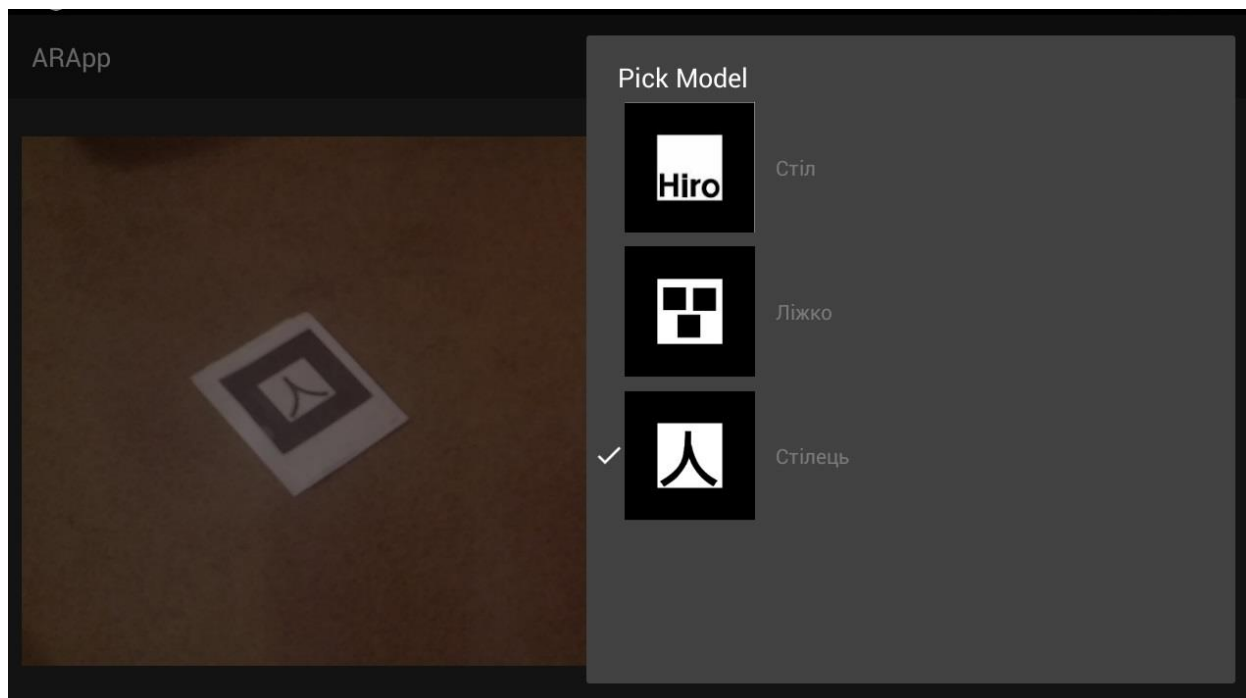


Рис. 3.20. Список доступних 3D моделей

Якщо натиснути кнопку «Controll» на панелі керування, відкриється діалогове вікно, в якому передбачена можливість трансформації поточної 3D моделі (рис. 3.21). Є можливість повертати, крутити, масштабувати та перереміщати об'єкт. Всі ці функції також реалізовані за допомогою жестів на сенсорному дисплеї пристрою. Простим проведенням пальцем по екрану реалізується функція повороту моделі у відповідному напрямку. При дотику до екрана двома пальцями одночасно і звуженням або розширенням їх реалізується функція масштабування. При дотику до екрана трьома пальцями одночасно і обертанням їх за або проти часової стрілки реалізується функція повороту моделі.

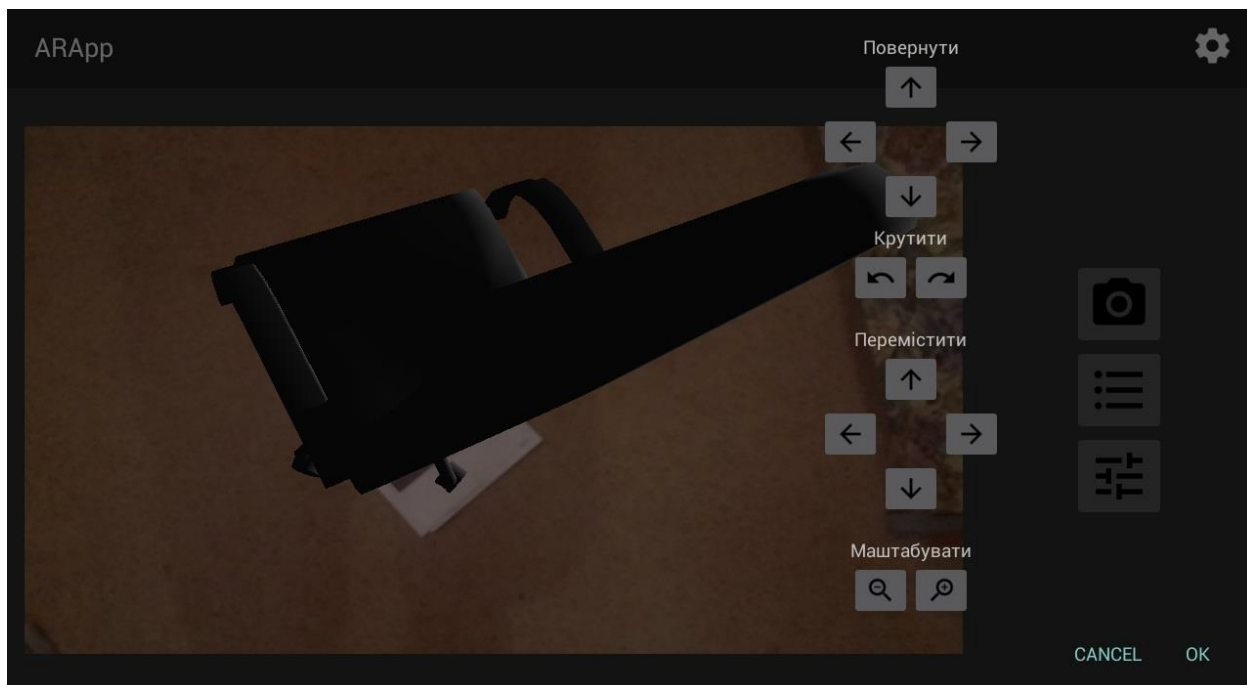


Рис. 3.21. Панель керування 3D моделлю

При натисненні кнопки «Settings» в меню додатку, відкривається нове вікно налаштувань додатку(рис. 3.22). В якості налаштувань реалізовано вибірку камери а також розширення камери.

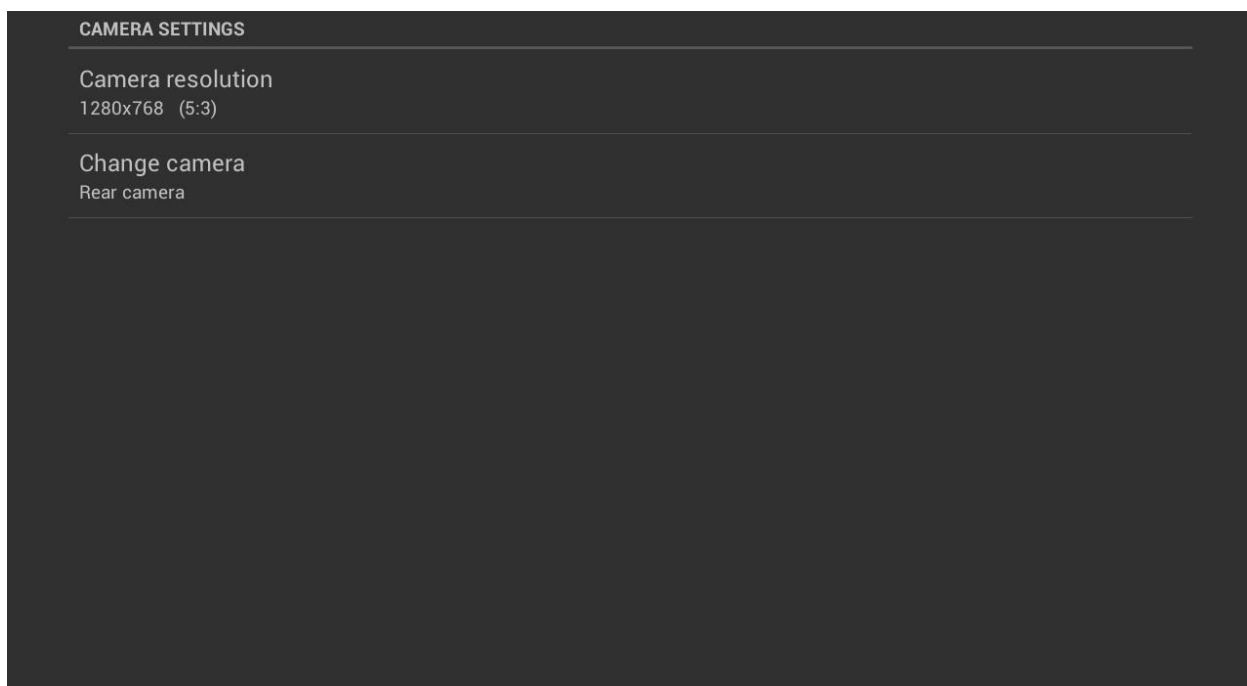


Рис. 3.22. Вікно налаштувань

Висновки до розділу 3

Було проведено огляд перетворень об'єктів у тривимірному просторі за допомогою матриць перетворень. Також було розроблено схему обчислення параметрів матриць масштабування та повороту, де вхідними даними є жести користувача на сенсорному дисплеї.

Хоча проект розроблявся на мові програмування Java, для підвищення швидкодії був насаний C++ файл, який відповідає за маніпуляції з 3D моделями. Зв'язування Java коду і C++ відбувається за допомогою JNI інтерфейсу.

Описано процес розробки і генерації маркера і інтеграції його в android проект.

					ІАЛЦ.467100.003 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У рамках виконання дипломної роботи був розроблений мобільний додаток для платформи Android для моделювання інтерєру з використанням технології доповненої реальності. Було досягнуто наступне:

- Був проаналізований та досліджений стан технологій створення мобільних додатків доповненої реальності. Були обрані платформа Android, мови програмування Java та C/C++, середовище розробки Android Studio.
- Були розглянуті популярні бібліотеки для роботи з доповненою реальністю. Проаналізовані їхні плюси й мінуси і вибрано технологію ARToolkit для розробки додатку.
- Реалізовані «user-friendly» дизайн, ефективний функціонал та надійність мережевого з'єднання мобільного додатку.
- Готовий додаток був протестований як на стандартних емуляторах Android пристроїв, взятих з Android SDK, так і на емуляторі Genymotion, а також на реальному пристрої на платформі Android OS.

Таким чином, були досягнуті всі поставлені задачі в повному обсязі. Планується інтеграція додатку для інших мобільних платформ. В подальшому додаток буде впроваджено на ресурс загального доступу користувачів Android-пристроїв Google Play.

					ІАЛЦ.467100.003 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

ЛІТЕРАТУРА

1. Ronald T. Azuma A Survey of Augmented Reality // In Presence: Teleoperators and Virtual Environments. – 1997. – No 4. – P. 355–385.
2. Miika Tikander. Development and evaluation of augmented reality audio systems: Abstract of dissertation for the degree of Doctor of Science in Technology. – Helsinki, 2009.
3. Joseph Rozier. Hear&There: An Augmented Reality System of Linked Audio / Joseph Rozier, Karrie Karahalios, Judith Donath // Online Proceedings of the ICAD, 2009
4. Karrie Karahalios. GaitAid Virtual Walker for Movement disorder patients Augmented Reality, 2007
5. PHANTOM Premium [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <http://www.sensable.com/haptic-phantom-premium-6dof.htm>.
6. Artollkit About [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://archive.artoolkit.org/>.
7. Android [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Android>.
8. Windows Phone [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Windows_Phone.
9. Blackberry [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://us.blackberry.com/>.
10. iOS [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/IOS>.
11. How Reality Technology is Being Used in Design [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <http://www.realitytechnologies.com/design>.
12. Реальность на ладони: обзор приложения Augment [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <http://arnext.ru/reviews/realnost-na-ladoni-obzor-prilozheniya-augment-3650>.
13. ARToolKit [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/ARToolKit>.

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

14. Java [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Java>.
15. Android NDK [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/ndk/index.html>.
16. Введение в Android NDK [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: <https://habrahabr.ru/post/203014/>.
17. Welcome to the OpenGL Wiki! [Електронний ресурс] – Режим доступу до ресурсу: <https://www.khronos.org/opengl/wiki/>.
18. OBJ [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Obj>.
19. Model-View-Controller [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Model-View-Controller>.