

ЗМІСТ

ВСТУП	2
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	3
1.1 Доповнена реальність	3
1.2 Мобільні пристрої.....	17
1.3 Програми-аналоги.....	19
Висновки до розділу.....	23
РОЗДІЛ 2 ОПИС ЗАСОБІВ РОЗРОБКИ ТА РОБОТИ ДОДАТКУ	24
2.1 Постановка задачі дипломного проектування.....	24
2.2 Огляд засобів для реалізації Android додатка	24
2.3 Вхідні та вихідні дані.....	40
Висновки до розділу	41
РОЗДІЛ 3 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ	42
3.1 Опис математичної моделі	42
3.2 Підготовка проекту.....	46
3.3 Архітектура програмного забезпечення	51
3.4 Інтеграція ресурсів в додаток.....	56
3.5 Збірка проекту.....	58
3.6 Опис реалізованого програмного забезпечення.....	59
Висновки до розділу	69
РОЗДІЛ 4 СТАРТАП ПРОЕКТ	70
4.1 Опис ідеї проекту.....	70
4.2 Технологічний аудит ідеї проекту	71
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	71
4.4 Розроблення ринкової стратегії проекту	75
4.5 Розроблення маркетингової програми стартап-проекту.....	76
Висновки до розділу	78
ВИСНОВКИ	79
ЛІТЕРАТУРА	80

					ІАЛЦ.467100.003 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Сухина М. С.			Система візуалізації інтер'єру методом доповненої реальності	Літ.	Арк.
Перевір.		Антонюк А. І.					1
							84
Н. контр.		Жабін В.І.				НТУУ КПІ ФІОТ Група ІВ-71мп	
Затверд.		Стіренко С.Г					

ВСТУП

На даний час одним з найбільш перспективних напрямків ІТ-розробок є доповнена реальність (augmented reality, AR)[1]. Дана технологія - це новий спосіб представлення та отримання інформації.

Доповнена реальність здатна зробити сприйняття інформації людиною набагато простіше і наочніше. Необхідні запити будуть автоматично доставлятися користувачеві.

Доповнена реальність - це, перш за все, технологія, за допомогою якої реальні об'єкти набувають нових якостей і відкриваються користувачу з іншого боку. Принцип доповненої реальності полягає в накладенні віртуальних об'єктів на існуючі об'єкти в режимі реального часу. Взаємодія техніки з зображенням реального світу відрізняє доповнену реальність від віртуальної.

Основним завданням доповненої реальності є розширення можливостей користувачів, тобто їх взаємодія з оточенням, але вже на принципово новому рівні. За допомогою комп'ютерного пристрою на зображення реального середовища наносяться додаткові шари з набором об'єктів, несучих додаткову інформацію. Зараз технології дозволяють зчитувати і розпізнавати зображення навколишнього середовища при допомогі камер, а також доповнювати їх за допомогою неіснуючих або фантастичних об'єктів. Можна сказати, що доповнена реальність може розповісти все про необхідний нам об'єкт в реальному часі. Вже зараз існують різні технології, які і здійснюють це завдання. Наприклад: маркери роблять рекламу більш привабливою, системи розпізнавання руху роблять можливим управління інтерфейсами при безконтактній взаємодії. Таким чином, потрібна інформація стає доступною користувачеві в режимі реального часу, не вимагаючи зусиль для її пошуку в інших джерелах. Доповнена реальність - це новий метод отримання інформації та до інших різних даних, але вплив цієї технології, можливо, надасть незабутнє враження на людину, порівнянне з виникненням інтернету.

					ІАЛЦ.467100.003 ПЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Доповнена реальність

1.1.1 Передмова

Експерти намагаються визначити межі термінів і навіть виділяють стадії переходу від реального до віртуального світу, наприклад, «Mixed Reality in Architecture, Design and Construcrion» Xiangyu Wang і Marc Aurel Schnabel з Університету Сіднея, а також «The Engineering of Mixed Reality Systems »під керівництвом Emmanuel Dubois. У своїх дослідженнях автори сходяться в думках, що змішаною реальністю називається технологія, де віртуальний і реальний світ взаємодіють. Деякі фахівці йдуть далі і виділяють реальну віртуальність або Real Virtuality (RV) - коли людина починає існувати у віртуальному світі. Загальновизнаних стандартів поки не існує, але точно виділяються наступні стадії переходу до RV[2].

Реальний світ ми бачимо самі, без будь-яких додаткових гаджетів і технологій.

Віртуальна реальність (VR) повністю відсікає реальний світ, людина бачить картинку, намальоване, спроектоване оточення.

Доповнена реальність (AR) частково замінює реальний світ, на існуючий світ накладається віртуальне зображення. По суті це підказка чи голограма, намальована поверх реального світу. Важливо розуміти, що віртуальна картинка не дає відчуття реального розташування і взаємодії об'єктів з оточуючими світом. І саме в цьому ключова відмінність доповненої реальності від змішаної реальності.

Змішана реальність (MR) дозволяє бачити взаємодію реальних і віртуальних об'єктів. Людина вже може оцінити передній і задній план, як об'єкти розташовані щодо один одного і, найважливіше - з'являється точка дотику реальних і віртуальних об'єктів.

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

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»

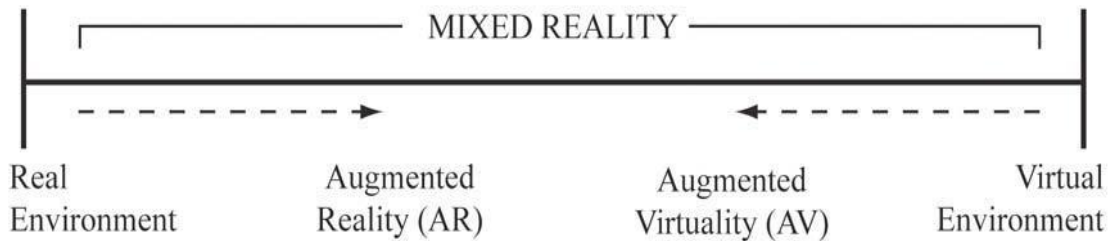


Рис. 1.1. Діапазон технологій змішаної реальності

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 цю 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. Сам термін «Доповнена реальність» (augmented reality, AR) був запропонований дослідником корпорації Boeing Томом Коделом в 1990 р.[3].

Взагалі існує кілька визначень доповненої реальності, однак найбільш оптимальним можна вважати визначення дослідника Рональда Азума (англ. Ronald Azuma) який в 1997 році визначив її як систему, яка [4]:

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

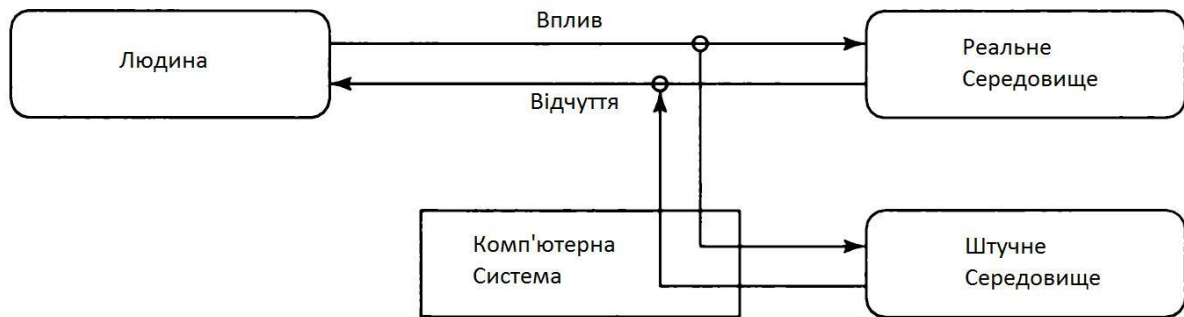


Рис. 1.1. Схема середовища доповненої реальності

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.

1.1.2 Класифікація технології доповненої реальності

Людина отримує уявлення про навколишній простір за допомогою різних 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 [5]. 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 [6].

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 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

					ІАЛЦ.467100.003 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

системи можуть аналізувати об'єкти, що знаходяться в полі зору людини і видавати довідкову інформацію про них. Також системи такого типу використовуються в медицині. Наприклад, система Gait Aid[7] 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 [8].

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, 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.

					ІАЛЦ.467100.003 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Подібні системи можуть показувати частини об'єктів інтересу, приховані іншими будівлями, додаткову інформацію про обрані об'єктах і т.д.

За ступенем мобільності [9] 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.

За реалізованою функціональною цінністю для споживача додатки доповненої реальності можна розділити на наступні п'ять напрямків:

Візуальний пошук. Передбачає підказки навігаційного характеру за запитом користувача. Причому в даному випадку мова не йде про вже традиційні дорожні навігатори, коли користувачеві потрібно переміститися з однієї географічної точки в іншу. В даному випадку мається на увазі розширений варіант запитів, пов'язаний з пошуком конкретного товару або послуги, об'єкта із затребуваними характеристиками, людей, відгуків і т.п.

Розпізнавання. Дана реалізація доповненої реальності передбачає надання контекстної інформації про об'єкт або людину в полі зору.

Людина 2.0 - даний тип додатків доповненої реальності передбачає надання покрокових інструкцій для реалізації конкретної задачі, наприклад: підбір рецепта, алгоритм приготування певної трави і контроль інгредієнтів, дозування і послідовності

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

Екран-дзеркало / лінза. Такий функціонал додатка доповненої реальності передбачає накладення віртуальних об'єктів на зображення оточення реального світу для кращого уявлення про просторові характеристики віртуального об'єкта.

Візуалізація продукції під контекстні завдання. Даний алгоритм більше підходить для промислових підприємств для вирішення інженерних або конструкторських завдань.

Можливості додатків з технологією доповненої реальності будуть найбільше затребувані в чотирьох областях:

Маркетингові комунікації.

Цінність пропозиції: доповнена реальність має великий потенціал, щоб розширити і підтримати зусилля бізнесу в просуванні своїх товарів і послуг. Маркетингові служби компанії можуть скористатися наявними можливостями доповненої реальності, щоб забезпечувати контекстні зв'язки між своєю пропозицією для споживачів, інтернет-ресурсами та точками продаж. Використання доповненої реальності разом з ресурсами друкованих ЗМІ може посилити емоційне включення споживачів в процес розглядання рекламних оголошень через використання 3D-зображень або медійно насиченого контенту і тим самим сприяти збільшенню продаж.

Потенційні споживачі: мережевий ритейл, торгові центри, готелі, ресторани, сфера послуг, транспорт, логістика, поштові послуги.

Продажі

Цінність пропозиції: доповнена реальність може поліпшити якість продажів як в традиційних каналах продажів, так і в електронній торгівлі. Додаткові шари даних можуть включати будь-яку корисну інформацію (опис продукту, порівняння з іншими альтернативами, відгуки і т.п.) при розгляданні товару через відеокамеру смартфона або іншого мобільного пристрою. Для електронної торгівлі доповнена реальність може стати інструментом докладного вивчення товару потенційним покупцем в контексті його умов.

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

Потенційні споживачі: електронна торгівля, магазини з продажу одягу, меблів, предметів інтер'єру, послуги по дизайну і обробці приміщень, салони краси, косметичні засоби, послуги пластичної медицини.

Післяпродажне обслуговування і навчання

Цінність пропозиції: технологія доповненої реальності може стати хорошим інструментом для пост продажного обслуговування або навчання споживача користуватися придбаним товаром. Додатки доповненої реальності, що легко завантажуються можуть стати інструкцією по збірці, користування або ремонту. Можливість включити додатковий контент в контекст традиційних текстових медіа-ресурсів підвищує їх привабливість не тільки з комерційної точки зору. Медійно насичена інформація за рахунок наочності подачі матеріалу полегшує і прискорює процес навчання, а також відповідає потребам сучасного молодого покоління.

Потенційні споживачі: медицина, освіта, ремонтні роботи, авіація, експлуатація автомобілів, армія, поліція, служби безпеки, митниця, будівництво, диспетчерські служби транспортних компаній, металургія та ін.

Логістика

Цінність пропозиції: технологія доповненої реальності в сукупності з геолокаційною інформацією відкриває нові можливості для управління логістикою і транспортними потоками. Користувачі за допомогою додатків з доповненою реальністю можуть отримувати оперативну інформацію про наявну транспортну інфраструктуру, графік руху громадського транспорту, завантаженості доріг, наявності черг в сервісних службах і т.д. Іншими словами, доповнена реальність може підняти на новий рівень планування і управління часом в умовах мегаполісів як для компаній, так і для приватних користувачів.

Потенційні споживачі: транспортні компанії, громадський транспорт, служби доставки, дорожня поліція, приватні користувачі.

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

1.1.3 Мобільні системи доповненої реальності

Мобільні системи доповненої реальності включають в себе мобільні додатки для смартфонів та планшетів[10]. 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 смартфонів мають додаткові вбудовані датчики такі як: акселерометри, магнітометри, GPS-приймачі та ін., які можуть поліпшити роботу AR програми.

У мобільній доповненій 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 віртуальна будівля, побудована на реальній площині, то для такої доповненої реальності необхідна додаткова інформація, така як кордони площини та її розміри. Отримання цієї додаткової інформації зазвичай досягається за допомогою спеціальних маркерів або за допомогою спеціальних функцій розпізнавання.

					ІАЛЦ.467100.003 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

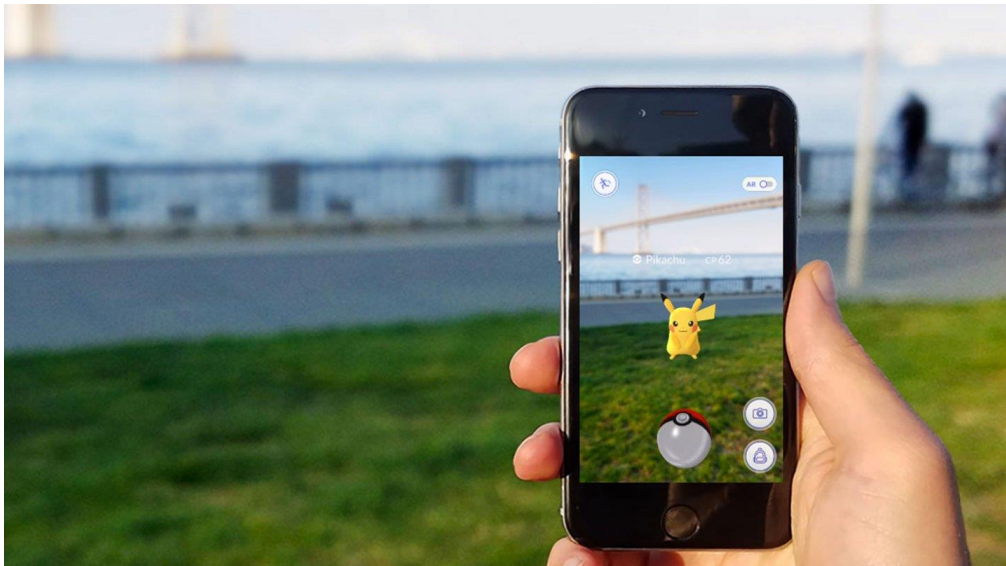


Рис. 1.2. AR на мобільному пристрої

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 реальне середовище 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 інерційними 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. Одна з найбільш "просунутих" мобільних систем є Google Goggles; ця система може: розпізнавати об'єкти простої форми, наприклад, штрих-коди 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

Протягом останніх трьох років Google працює над технологіями, які сприяють розвитку доповненої реальності на мобільних пристроях, за допомогою платформи Tango. Вона стала основою для створення ARCore[11]. Завдяки тому, що цей SDK не вимагає додаткового обладнання, він може застосовуватися на різних пристроях Android. З дати його першого релізу він був доступний лише на Pixel і Samsung S8, які працюють на Android 7.0 і пізніших версіях. На сьогодні даний SDK підтримується на десятках мобільних девайсах провідних компаній, серед яких Samsung, Huawei, LG, ASUS та ін..

ARCore працює на Java / OpenGL, Unity і Unreal і фокусується на таких напрямках:

Відстеження руху. Використовуючи камеру телефону для відстеження опорних точок в кімнаті (п.п. ці точки визначають місце, де буде розташований віртуальний об'єкт) і даних гіроскопа, ARCore визначає положення і

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

орієнтацію пристрою під час руху. При цьому віртуальні об'єкти залишаються саме там, де ви їх розташували.

Розпізнавання навколишнього середовища. Зазвичай об'єкти доповненої реальності розміщуються на підлозі або столі. ARCore може розпізнавати горизонтальні поверхні, використовуючи ті ж опорні точки, що і при відстежуванні руху.

Оцінка освітлення. ARCore визначає рівень освітленості навколишнього середовища і дає можливість розробникам висвітлювати віртуальні об'єкти відповідно до обстановки навколо. Завдяки цьому вони виглядають ще більш реалістично.

Поряд з ARCore Google розвиває і інші програми та сервіси, які допоможуть розробникам створювати відмінні рішення в області доповненої реальності. Зокрема запустили проекти Blocks і Tilt Brush, щоб кожен з легкістю міг створювати хороший 3D-контент для своїх AR-додатків. На конференції Google I / O Google оголосили, що працює над сервісом візуального позиціонування (Visual Positioning Service). Він дозволить вивести застосування технології доповненої реальності в світі за межі комп'ютера. Google також вважає, що ключова роль у розвитку доповненої реальності буде належати інтернету. Тому запускає прототипи браузерів для веб-розробників, щоб вони теж могли почати експериментувати з цією технологією. Спеціалізовані браузери дозволяють створювати сайти з підтримкою доповненої реальності для Android / ARCore і iOS / ARKit.

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим кодом[12]) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.

					ІАЛЦ.467100.003 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Бібліотека розроблена Intel і нині підтримується Willow Garage та Itseez. Сирцевий код бібліотеки написаний мовою C++ і поширюється під ліцензією BSD. Біндинги підготовлені для різних мов програмування, таких як Python, Java, Ruby, Matlab, Lua та інших. Може вільно використовуватися в академічних та комерційних цілях.

Vuforia - це платформа доповненої реальності і інструментарій розробника програмного забезпечення доповненої реальності [13] (Software Development Kit - SDK) для мобільних пристроїв, розроблені компанії Qualcomm. Vuforia використовує технології комп'ютерного зору, а також відстеження плоских зображень і простих об'ємних реальних об'єктів (наприклад, кубічних) в реальному часі. З версії 2.5 Vuforia розпізнає текст, а з 2.6 - має можливість розпізнавати циліндричні маркери.

Можливість реєстрації зображень дозволяє розробникам розташовувати і орієнтувати віртуальні об'єкти, такі, як 3D-моделі і медіаконтент, в зв'язці з реальними образами при перегляді через камери мобільних пристроїв. Віртуальний об'єкт орієнтується на реальному образі так, щоб точка зору спостерігача ставилася до них однаковим чином для досягнення головного ефекту - відчуття, що віртуальний об'єкт є частиною реального світу.

Vuforia надає інтерфейси програмування додатків на мовах C++, Java, Objective-C, і .Net через інтеграцію з ігровим движком Unity. Таким чином SDK підтримує розробку нативних AR-додатків для iOS і Android, в той же час припускаючи розробку в Unity, результати якої можуть бути легко перенесені на обидві платформи. Додатки доповненої реальності, створені на платформі Vuforia, сумісні з широким спектром пристроїв, включаючи iPhone, iPad, смартфони та планшети на Android з версії 2.2 і процесором, починаючи з архітектур ARMv6 або 7 з можливістю проведення обчислень з плаваючою комою.

ARToolkit - це бібліотека комп'ютерного стеження для створення додатків з доповненою реальністю [14]. Для цього він використовує можливості відео спостереження, розрахунок реального стану та орієнтації

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

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 доповненій 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

Таблиця 1.1.

Порівняння технологій AR

Альтернативи	Кросплатформенність	Підтримка мов програмування	Тип ліцензування
ARCore	Android, iOS, Unity	Java, Kotlin, Swift, Objective-C, .NET	Free+Comercial SDK option
OpenCV	iOS, PC, Android, Linux	C++,Python,Java, Ruby, Matlab	Boost Software License
Vuforia	iOS, Android, Unity	C++,Java, Objective-C, .NET	Free+Comercial SDK option
ARToolKit	iOS, Android, Unity	Java, Objective-C,	Free+Comercial SDK option
Metaio	iOS, Android, Unity,WEB	Java, Objective-C, .NET	Free+Comercial SDK option

Огляд існуючих технологій для роботи з AR з урахуванням найбільш важливих критеріїв показав, що найоптимальнішим середовищем для розробки є ARCore. ARCore – потужна, інноваційна, безкоштовна бібліотека, постійно модернізується і має великі перспективи.

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:

Android. Операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux [15]. 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

					ІАЛЦ.467100.003 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Development Kit дозволяє перенести бібліотеки і компоненти додатків, написані на C та інших мовах.

IOS. Заснована на спеціальній версії Mac OS X[16]. 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.

KaiOS - мобільна операційна система[17], заснована на базі операційної системи Firefox OS, розробка якої була припинена Mozilla в 2016 році. Операційна система з'явилася в 2017 році і розроблена KaiOS Technologies, компанією, що базується в Сан-Дієго, штат Каліфорнія, очолюваної Себастьяном Коудвіллем, і має відділення в інших країнах.

Основними можливостями KaiOS є підтримка LTE, GPS, Wi-Fi, додатків на HTML5, а також велика енергоефективність на пристроях без сенсорного екрану і менше споживання пам'яті. Крім того, є магазин додатків, що дозволяє користувачам завантажувати додатки. Деякі сервіси постачаються з операційною системою, такі як HTML5-додатки, включаючи Twitter, Facebook і YouTube. KaiOS може працювати з пристроями, в яких розмір оперативної пам'яті становить 256 МБ.

Windows Phone. Мобільна операційна система, розроблена Microsoft, вийшла 11 жовтня 2010 року[18]. 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.

					ІАЛЦ.467100.003 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

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



Рис. 1.3. Розподіл мобільних ОС станом на жовтень 2018

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 75%, це означає, що додатки написані під Android матимуть більший попит, ніж на 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[19] є захоплюючим поєднанням 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 об'єкт віртуальної реальності або доповненої реальності.

Програмне забезпечення сумісне з VR і AR технологіями. У режимі VR користувачі можуть буквально «пройти через свої креслення». Цей тип взаємодії є надзвичайно корисним, щоб отримати відчуття масштабу і розуміння того, як люди всередині будівлі будуть рухатися.

В AR режимі SmartReality створює моментальну 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.

					ІАЛЦ.467100.003 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.4. 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[20]. 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		21

Порівняльна характеристика програм-аналогів

Характеристика	IKEA	Augment	SmartReality
Наявність AR технології	+	+	+
Наявність VR технології	-	-	+
Використання маркера	+	-	+
Можливість розширити каталог моделей	+	+	+
Модифікація в режимі реального часу	+	+	-
Кросплатформенність	+	+	+

Проаналізувавши програми-аналоги, можна зробити висновок, що AR технологія є досить популярною в сфері інтерєру 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

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 ARCore 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.

Огляд існуючих додатків показав основні тенденції в розвитку технології доповненої реальності. Ця технологія в наш час дуже бурхливо розвивається і може знайти застосування в багатьох областях.

Аналіз додатків показав, що зараз вони мають, в основному, або розважальний, або вузькоспрямований характер. Основна увага в таких додатках приділяється їх можливостям при взаємодії з навколишнім середовищем і користувачем.

					ІАЛЦ.467100.003 ПЗ	Арк.
						23
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, основами 3D графіки, роботою з камерою на ОС Android, роботою з багатопоточністю в Java.

2.2 Огляд засобів для реалізації Android додатка

2.2.1 ARCore

Протягом останніх трьох років Google працює над технологіями, які сприяють розвитку доповненої реальності на мобільних пристроях, за допомогою платформи Tango. Вона стала основою для створення ARCore[21]. Завдяки тому, що цей SDK не вимагає додаткового обладнання, він може застосовуватися на різних пристроях Android. З дати його першого релізу він був доступний лише на Pixel і Samsung S8, які працюють на Android 7.0 і пізніших версіях. На сьогодні даний SDK підтримується на десятках мобільних девайсах провідних компаній, серед яких Samsung, Huawei, LG, ASUS та ін..

					ІАЛЦ.467100.003 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.1. Логотип ARCore

ARCore працює на Java / OpenGL, Unity і Unreal і фокусується на таких напрямках:

- Відстеження руху.
- Розпізнавання навколишнього середовища.
- Оцінка освітлення.

По суті, ARCore робить дві речі: відстежує позицію мобільного пристрою, коли він рухається, і формує власне розуміння реального світу.

Технологія відстеження руху ARCore використовує камеру телефону, щоб визначати «опорні точки», які називаються features та відслідковує, як ці точки рухаються з часом. Завдяки поєднанню руху цих точок і показань з інерційних датчиків телефону, ARCore визначає як позицію, так і орієнтацію телефону, коли він переміщується в просторі.

Крім визначення основних точок, ARCore може виявити плоскі поверхні, наприклад, столу або підлогу, а також може оцінити середнє освітлення в районі навколо нього. Ці можливості поєднуються, щоб дозволити ARCore створити своє власне розуміння навколишнього світу.

Уявлення ARCore про реальний світ дозволяє розміщувати об'єкти, анотації або іншу інформацію таким чином, щоб інтегруватися в реальний світ. Відстеження руху означає, що ви можете пересуватися та переглядати ці об'єкти з будь-якого кута, і навіть якщо ви обернетесь та вийдете з кімнати, коли повернетесь, об'єкт з'являться там, де ви його залишили.

					ІАЛЦ.467100.003 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні концепції[22]

Відстеження руху

Коли телефон переміщується в просторі, ARCore використовує процес, який називається *паралельна одометрія та відображення*, або COM (concurrent odometry and mapping), щоб зрозуміти, як положення телефону відноситься до навколишнього світу. ARCore виявляє візуально окремі features із зображень камери, які називаються «опорними точками», і потім використовує ці точки для обчислення зміни положення телефону в просторі. Візуальна інформація поєднується з даними з інерційних вимірювальних пристроїв (Inertial Measurement Units (IMU)) для оцінки позиції (положення та орієнтації) камери відносно простору з часом .

Корегуючи позицію віртуальної камери, яка рендерить ваш 3D-вміст враховуючи позицію камери пристрою, наданої ARCore, розробники можуть відображати віртуальний вміст з правильної точки зору. Відтворене віртуальне зображення може бути накладено поверх зображення, отриманого з камери пристрою, і воно виглядає так, ніби віртуальний вміст є частиною реального світу.

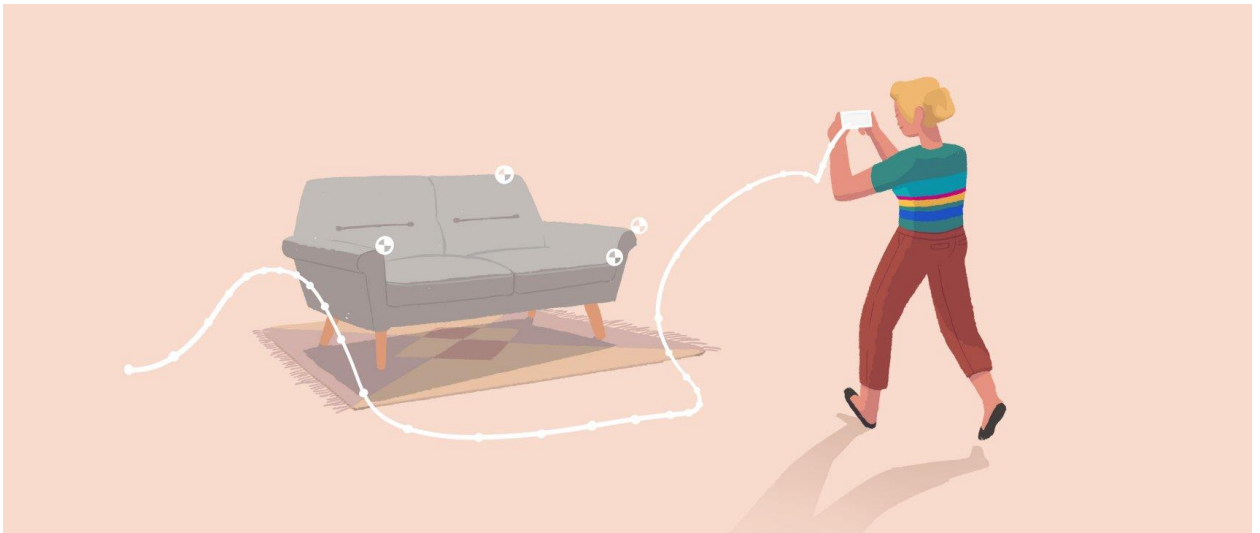


Рис. 2.2. Відстеження руху

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

26

Розуміння реального навколишнього середовища

ARCore постійно вдосконалює своє розуміння реального навколишнього середовища шляхом виявлення *опорних точок* (feature points) та *планів* (planes).

ARCore шукає кластери *опорних точок*, які, як видається, лежать на звичайних горизонтальних або вертикальних поверхнях, таких як стол, підлога або стіни, і робить ці поверхні доступними для AR-додатка як *плани*. ARCore також може визначити межу кожного плану та зробити цю інформацію доступною для AR-додатка. Згодом є можливість використовувати цю інформацію для розміщення віртуальних об'єктів, розташованих на рівних поверхнях.

Оскільки ARCore використовує опорні точки для виявлення площин, плоскі поверхні без текстури, такі як біла стіна, можуть не виявлятися належним чином.

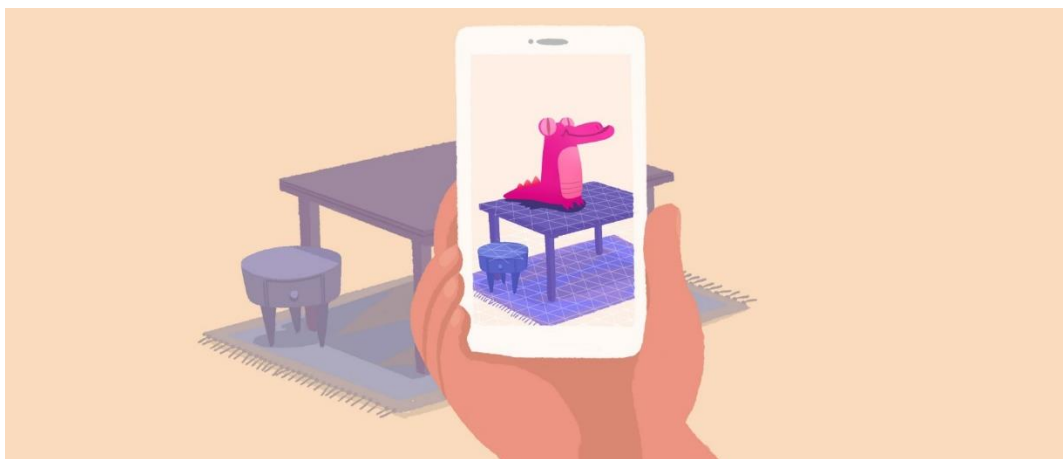


Рис. 2.3. Розуміння реального навколишнього середовища

Оцінка світла

ARCore може виявити інформацію про освітлення його оточення та забезпечити вам середню інтенсивність та корекцію кольору даного зображення камери. Ця інформація дозволяє освітлювати свої віртуальні об'єкти за тих самих умов, що й оточення навколо них, підвищуючи відчуття реалізму.



Рис. 2.4. Оцінка світла

Взаємодія з користувачем

ARCore використовує тестування на хіти (hit testing), щоб взяти координати (x, y), яка відповідає точці на екрані телефону (надана за допомогою кнопок або будь-якої іншої взаємодії, яку ви хочете, щоб підтримувала ваша програма), і проектує промінь у вигляд світу з камери, повертаючи будь-які *плани або опорні точки*, які промінь перетинає в світовому просторі. Це дозволяє користувачам вибирати або іншим чином взаємодіяти з об'єктами в середовищі.

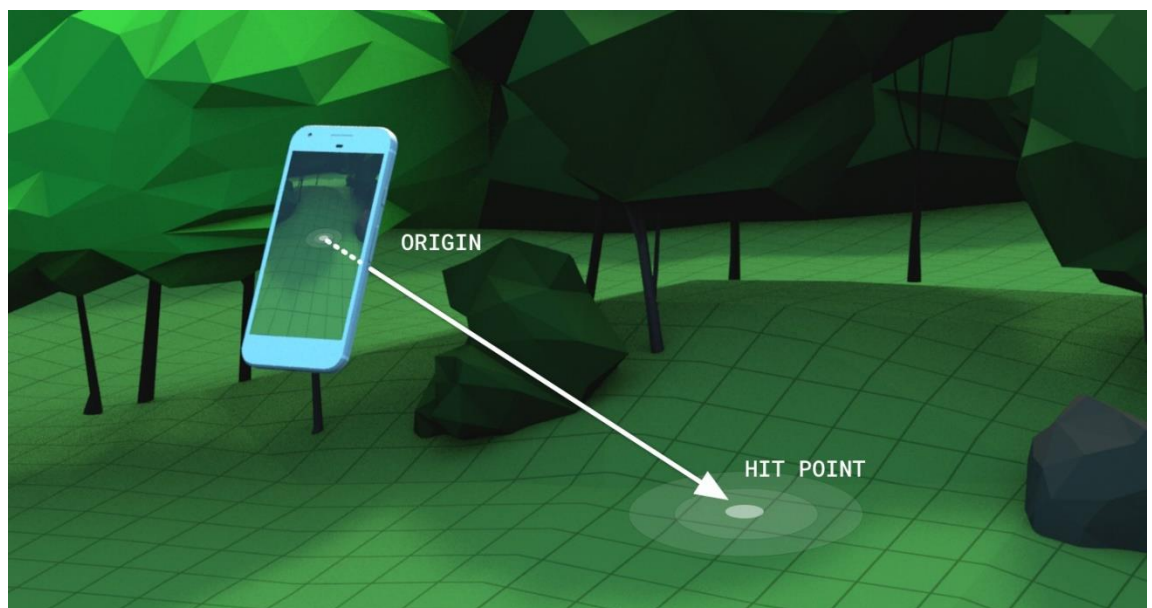


Рис. 2.5. Взаємодія з користувачем

Змн.	Арк.	№ докум.	Підпис	Дата

Орієнтовані точки

Орієнтовані точки (Oriented Points) дозволяють розмістити віртуальні об'єкти на кутових поверхнях. Коли ви виконуєте хіт-тест, який повертає опорну точку, ARCore буде дивитись на об'єкти, що знаходяться поблизу, і використовувати їх для спроби оцінити кут поверхні на даному об'єкті. ARCore потім поверне позу, яка враховує цей кут.

Оскільки ARCore використовує кластери опорних точок для виявлення кута поверхні, поверхню без текстури, такі як біла стіна, можуть бути неправильно виявлені.

Прив'язки(anchors) та відстеженості(Trackables)

Пози можуть змінюватися, оскільки ARCore покращує їх розуміння власної позиції та їх оточення. Якщо ви хочете розмістити віртуальний об'єкт, вам потрібно визначити *прив'язку(якір)*, щоб ARCore відслідковував позицію об'єкта з часом. Часто створюється якір на підставі пози, що повертається в ході тесту, як описано в взаємодії з користувачем.

Той факт, що пози можуть змінюватися, означає, що ARCore може оновлювати положення об'єктів навколишнього середовища, таких як плани та опорні точки з часом. Плани та точки - це особливий тип об'єкта, який називається відстежуваним (Trackables). Як підказує назва, це об'єкти, які ARCore відслідковує з часом. Ви можете прив'язати віртуальні об'єкти до певних треків, щоб переконатися, що взаємозв'язок між вашим віртуальним об'єктом і відстежуваним залишається стабільним навіть тоді, коли пристрій рухається навколо. Це означає, що якщо ви розмістите віртуальну фігурку Android на своєму столі, якщо ARCore пізніше налаштуватиме позицію плану, пов'язаного з робочим столом, фігурка Android все одно з'явиться на вершині столу.

Щоб зменшити витрати процесора, необхідно повторно використовувати якорі, коли це можливо, і від'єднати якорі, які вам більше не потрібні.

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

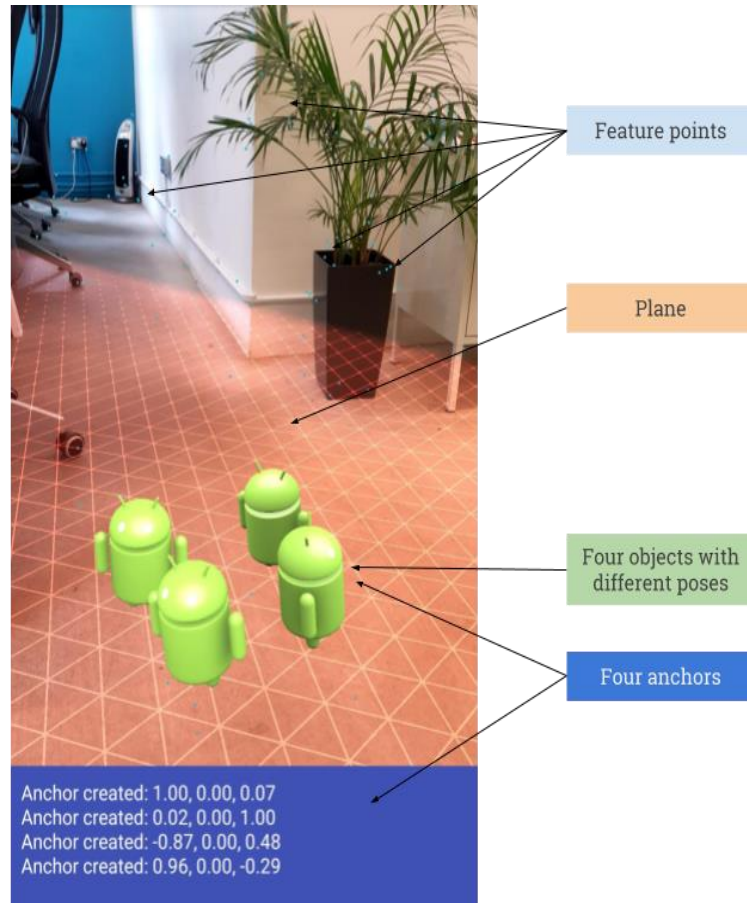


Рис. 2.6. Основні елементи ARCore

Розширені зображення

Розширені зображення дозволяють створювати додатки AR, які можуть реагувати на конкретні 2D-зображення, такі як упаковка продукту або фільми. Користувачі можуть викликати AR досвіди, коли вони вказують камеру свого телефону на певні зображення, наприклад, вони можуть вказувати камеру свого телефону на фільм-стендера і побачити зображення актора поверх розширеного зображення.

Зображення можна скопіювати офлайн, щоб створювати базу даних зображень, або окремі зображення можна додавати в реальному часі з пристрою. Після реєстрації ARCore буде виявляти ці зображення, границі зображень та повертати відповідну позу.

Спільний доступ

ARCore Cloud Anchors API дозволяє створювати спільні або багатокористувацькі додатки для пристроїв Android і iOS.

За допомогою Cloud Anchors один пристрій надсилає прив'язку та розташовує неподалік об'єкт, який вказує на хмару для хостингу. Ці якоря можна поділитися з іншими користувачами на пристроях Android або iOS у тій самій середовищі. Це дозволяє програмам надавати ті ж тривимірні об'єкти, які прикріплені до цих якорів, дозволяючи користувачам одночасно використовувати однакову технологію AR.

Підтримувані девайси

Для запуску і коректної роботи ARCore підходять лише сертифіковані девайси[23].

Сертифікація важлива, оскільки Google хоче, щоб користувачі мали хороший досвід роботи з AR-програмою на базі ARCore. Це пов'язано, перш за все, із чутливим відстеженням руху, яке здійснюється шляхом об'єднання зображення камери та даних датчика руху, щоб визначити, як пристрій користувача рухається в реальному світі.

Щоб засвідчити кожен пристрій, Google перевіряє якість камери, датчиків руху та архітектуру, щоб гарантувати, що він буде працювати, як очікувалося. Крім того, пристрій повинен мати досить потужний процесор, щоб забезпечити хорошу продуктивність та ефективні обчислення в реальному часі.

Сила Android залежить від великої різноманітності пристроїв, доступних в усьому світі. Google постійно працює з виробниками, щоб переконатися, що їхні апаратні засоби відповідають цим вимогам. У той же час, працює всередині, щоб переконатися, що ARCore добре інтегрується з кожною моделлю, яку ми сертифікуємо, щоб забезпечити хороший досвід для користувачів.

Станом на листопад 2018 ARCore підтримується понад сотнею девайсів виробників різних брендів, таких як: Acer, Asus, Google, Huawei, LG, Motorola, OnePlus, Samsung, Sony, Vivo, Xiaomi, HMD Global.

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

2.2.2 Scenform

Scenform - це 3D-фреймворк, представлений Google на IO'18[24]. API, що надає Scenform, допомагають швидко розвивати додатки ARCore, оскільки більшість основних концепцій, що потребуються, обробляються всередині фреймворку, наприклад, виявлення площин та оцінка освітленості здійснюється за допомогою Scenform.

Scenform робить простим рендеринг реалістичних 3D-сцен у програмах AR та інших програм без необхідності вивчати 3D-графіку або OpenGL. Він включає:

- API високого рівня
- Реалістичний рендеринг на основі фізики, який забезпечує Fiament
- Плагін Android Studio для імпорту, перегляду та створення 3D-активів

Основні елементи API, які використовуються в більшості додатків, є:

ArFragment - фрагмент, який можна додати до файла макета Android, як і будь-який інший фрагмент. Він автоматично перевіряє необхідну версію ARcore на пристрої, а також запитує дозволи на використання камери.

ArSceneView – отримує зображення камери та виділяє плани, коли вони виявляються ARCore. Ви можете використовувати *ArSceneView* безпосередньо у вашому додатку, однак у такому випадку потрібно вручну виконувати перевірки та дозволи.

ModelRenderable - дозволяє завантажувати 3D-моделі в загальному форматі sfb з вказаного шляху в файловій системі. Він повертає *Renderable*, який може бути відображений на екрані.

Node - це віртуальні об'єкти, які потрібно відтворити. Він містить всю інформацію, необхідну Scenform для відтворення об'єкта, яка надається API Pose (Позиція та орієнтація). Ви можете перетворити, анімувати та обертати вузол. Ви можете використовувати вбудований *TransformableNode* для перетворення об'єкта.

2.2.3 Java

Java — це об'єктно-орієнтована мова програмування. Синтаксис мови багато в чому походить від C та C++ [25]. 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 платформи-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.

2.2.4 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.
						33
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.

2.2.5 Android Studio

Android Studio — інтегроване середовище розробки[26] (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. 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.

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.

2.2.6 Google Sceneform Tools (Beta) plugin

Google Sceneform Tools (Beta) plugin[27] використовується щоб імпортувати 3D-моделі, перетворити їх у формат Sceneform та попередньо переглянути їх у Android Studio. Наразі плагін знаходиться на стадії бета-тестування, тому інколи можлива не коректна його робота.

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

2.2.7 Gradle

Gradle - система автоматичної збірки[28], побудована на принципах Apache Ant і Apache Maven, але надає DSL у Groovy мові замість традиційної XML-подібної форми представлення конфігурації проекту.

В відміню від Apache Maven, основанийого на концепції життєвого циклу проекту, і Apache Ant, в якому порядок виконання задач (цілей) визначається відносинами залежностей (depend-on), Gradle використовує спрямований ациклічний граф для визначення порядку виконання задач.

Градація була розроблена для розширених багатопроєктних збірок, і підтримує інкрементальні збірки, визначаючи, які компоненти дерева збирання не змінилися і які завдання, залежать від цих частин, не вимагають перезапуску.

Основні плагіни призначені для розробки та розгортання Java, Groovy та Scala додатків, але готуються плагіни і для інших програм програмування.

AndroidStudio використовує Gradle для збірки та налаштування проекту.

2.2.8 Git

Git — розподілена система керування версіями[29] файлів та спільної роботи. Проект створив Лінус Торвальдс для управління розробкою ядра Linux, а сьогодні підтримується Джуніо Хамано (англ. Junio C. Hamano). Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

Прикладами проєктів, що використовують Git, є ядро Linux, Android, LibreOffice, Cairo, GNU Core Utilities[en], Mesa 3D, Wine, багато проєктів з X.org, XMMS2[en], GStreamer, Debian DragonFly BSD, Perl, Eclipse, GNOME, KDE, Qt, Ruby on Rails, PostgreSQL, VideoLAN, PHP, One Laptop Per Child (OLPC), ABIC Koha, GNU LilyPond та ELinks і деякі дистрибутиви GNU/Linux.

					ІАЛЦ.467100.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Програма є вільною і випущена під ліцензією GNU GPL версії 2.

Система спроектована як набір програм, спеціально розроблених з врахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи управління версіями на базі Git або користувацькі інтерфейси. Наприклад, Cogito є саме таким прикладом фронтенда до репозиторіїв Git. А StGit використовує Git для управління колекцією латок.

Система має ряд користувацьких інтерфейсів: наприклад, gitk та git-gui розповсюджуються з самим Git.

Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH або HTTP сервером. TCP-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішим і надійним методом доступу. Метод доступу HTTP, незважаючи на ряд обмежень, дуже популярний в контрольованих мережах, тому що дозволяє використання існуючих конфігурацій мережевих фільтрів.

2.2.9 OpenGL

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 [30]. 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 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.

2.2.10 Формати .OBJ і FBX, .sfb, .sfa

Тривімірна графіка (3D, 3 Dimensions) — розділ комп'ютерної графіки, сукупність прийомів та інструментів (як програмних, так і апаратних), призначених для зображення об'ємних об'єктів [31].

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 та FBX.

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 багатокутників. Вершини

					ІАЛЦ.467100.003 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

FBX (Filmbox) - технологія та формат файлів, розроблена Kaydara, в даний час належить і продовжує розроблятися компанією Autodesk. Використовується для забезпечення сумісності різних програм тривірневої графіки.

FBX може зберігатися на диску у вигляді бінарних або ASCII даних, FBX SDK підтримує обидва формати. Обидва формати не задокументовані, однак формат ASCII має деревовидну структуру з чіткими позначеннями ідентифікаторів. В той час як бінарний формат не задокументований, є неофіційна специфікація, представлена Blender Foundation. Існує також високий рівень неофіційних специфікацій (в розробці), про те, як дані представлені в FBX (залежить від ASCII або двоїчного формату), також опубліковані Blender Foundation.

При використанні Google Sceneform Tools (Beta) plugin генеруються додаткові файли з форматами .sfb і .sfa. Файл * .sfa - це текстовий файл, який містить повний, зрозумілий для читання, опис параметрів імпорту об'єкта. Він посилається на моделі та текстури у вихідному об'єкті, а також визначає матеріали, надаючи параметри цим матеріалів для матеріалів Sceneform. Команда compileAsset- <ім'я_файлу> компілює файл * .sfa у файлі двійкового об'єкта Sceneform (.sfb). Цей файл * .sfb вбудовується в APK вашого додатка та завантажується під час виконання, щоб створити візуалізатор 3D моделі.

2.3 Вхідні та вихідні дані

Вхідними даними буде зображення отримане з камери мобільного пристрою. Якщо дане зображення міститиме в собі плоскі поверхні, система проаналізує його, розрахує опорні точки, визначить конфігурацію 3D моделі, яка буде рендериться поверх даної площини.

Вихідними даними буде демонстрація 3D моделі, яка буде рендериться поверх даної площини у випадку успішного розпізнання поверхні системою. Якщо ж система не розпізнає поверхню то вихідними даними залишається зображення з камери.

					ІАЛЦ.467100.003 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу

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 було розглянуто інструментальні засоби для рішення завдання технічного проекту. В ході більш детального огляду бібліотеки доповненої реальності ARCore було виявлено, що вона повністю задовольняє умовам, необхідним для рішення поставленої задачі. Оскільки розробка додатку буде вестися під операційну систему Android то мовою програмування було обрано мову Java, а середовище розробки – AndroidStudio. Було вирішено використати систему контролю версій git, для ефективного керування версіями проекту. Розглянуто основні формати тривимірної графіки .OBJ з .MTL і .FBX, для розуміння їх структури ефективної інтеграції активів у додаток.

					ІАЛЦ.467100.003 ПЗ	Арк.
						41
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 базису $(\beta_1, \beta_2, \dots, \beta_n)$ до базису $(\alpha_1, \alpha_2, \dots, \alpha_n)$ 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.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1

<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).

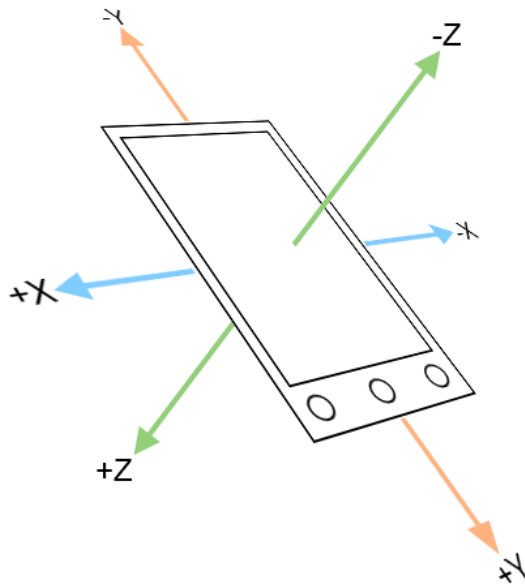


Рис. 3.1. Напрямок осей

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 обраховується як $|M'_2M_2| = \sqrt{M_2'^2}$. Модуль вектора M'_2O обраховується як $|M'_2O| = \sqrt{M_2'^2 + M_2^2}$. Отже $\sin(\beta) = |M'_2M_2| / |M'_2O| \Rightarrow \beta = \arcsin(\sin(\beta))$.

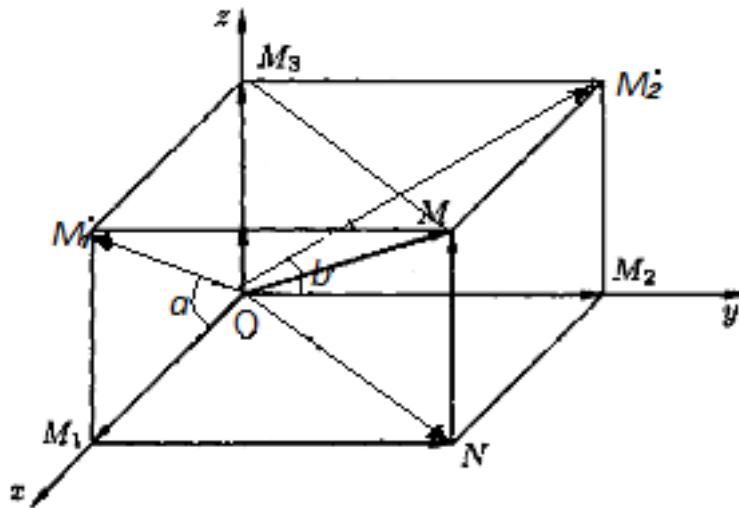


Рис. 3.2. Проекція вектора OM на вісі OX і OY

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).

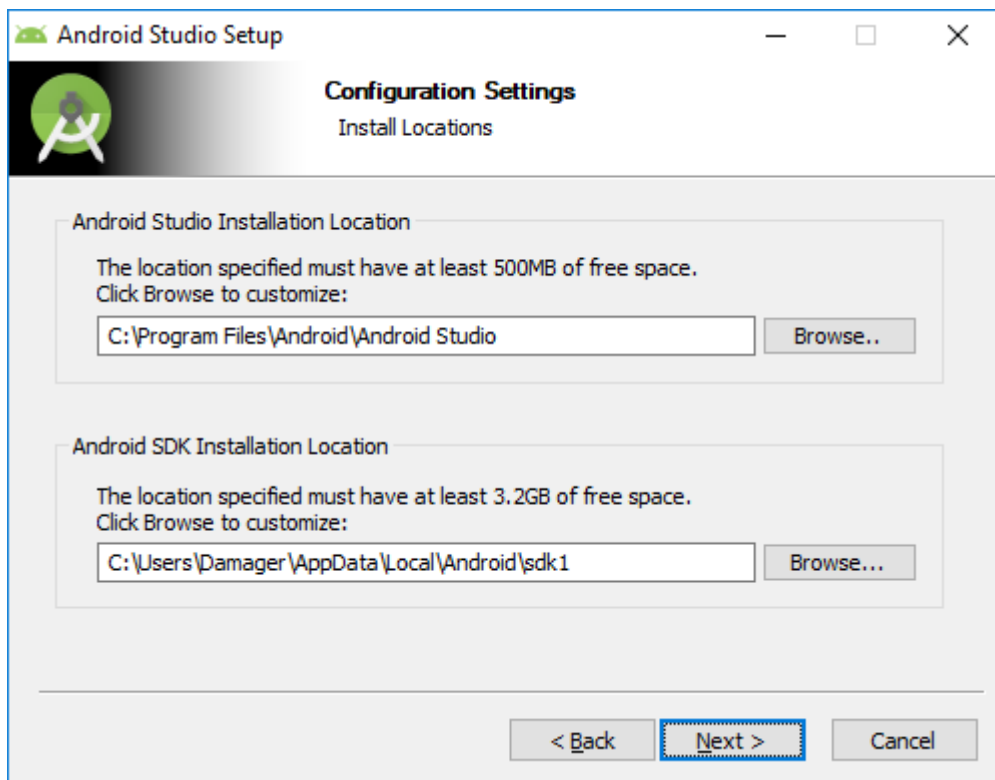


Рис. 3.3

Далі слідуючи підказкам інсталятора завершити встановлення. Під час установки необхідно буде з'єднання з інтернетом для завантаження Android SDK. На цьому налаштування середовища розробки можна вважати завершеним.

3.2.2 Створення проекту

Наступним кроком є створення проекту. Для створення проекту необхідно запустити Android Studio. У відкритому вікні вибрати «Start a new Android Studio project» (рис 3.4).

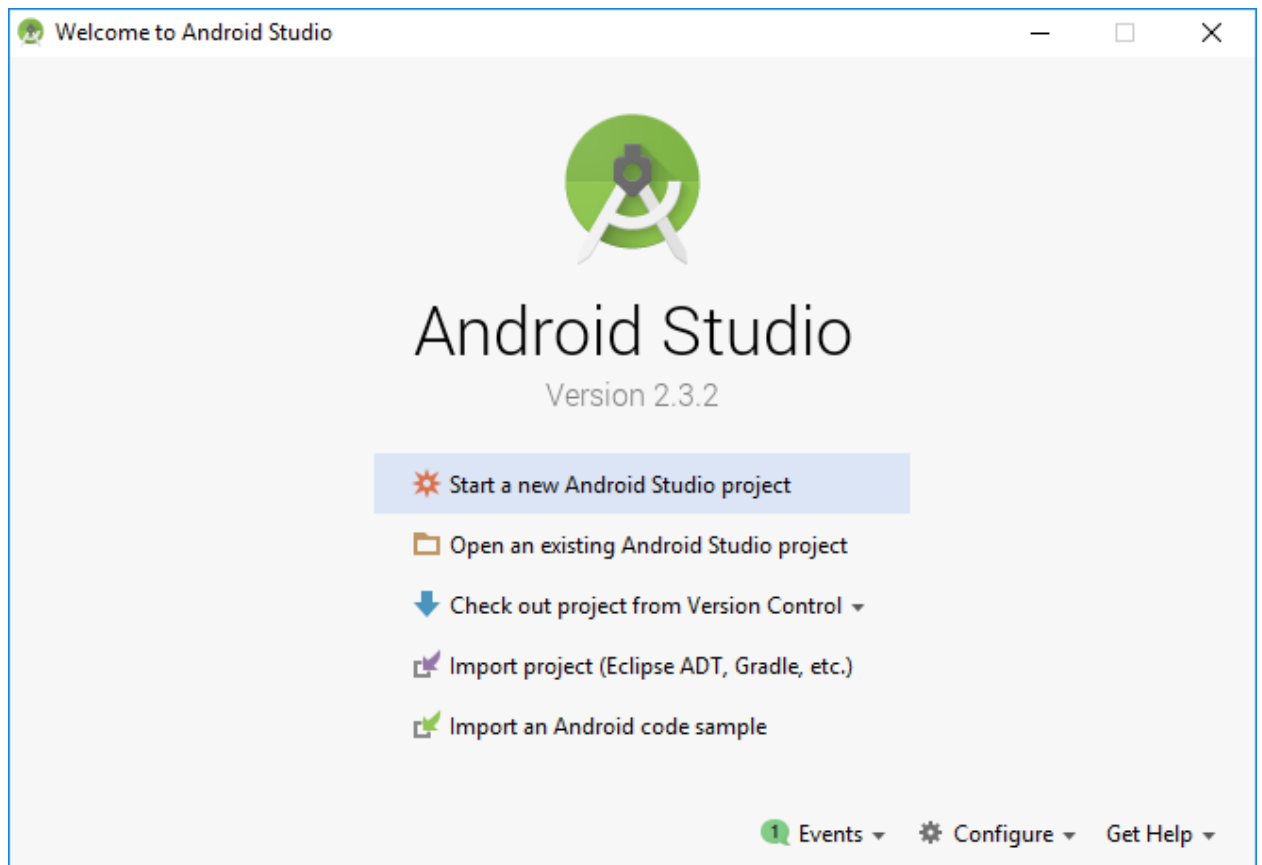


Рис. 3.4

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 натистути 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 24, Android 7.0 (Nougat)» (рис 3.5), це значить, на версіях Android нижче ніж 7.0 додаток працювати не буде. Хоча краще було б обрати меншу версію, і розшитити аудиторію користувачів проте ARCore

					ІАЛЦ.467100.003 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

працює лише на версії Android 7.0, що значно обмежує аудиторію потенційних користувачів. Проте перевагою є те, що для підтримки старіших версій, довелося б відмовитися від деякої функціональності введеної в нових версіях.

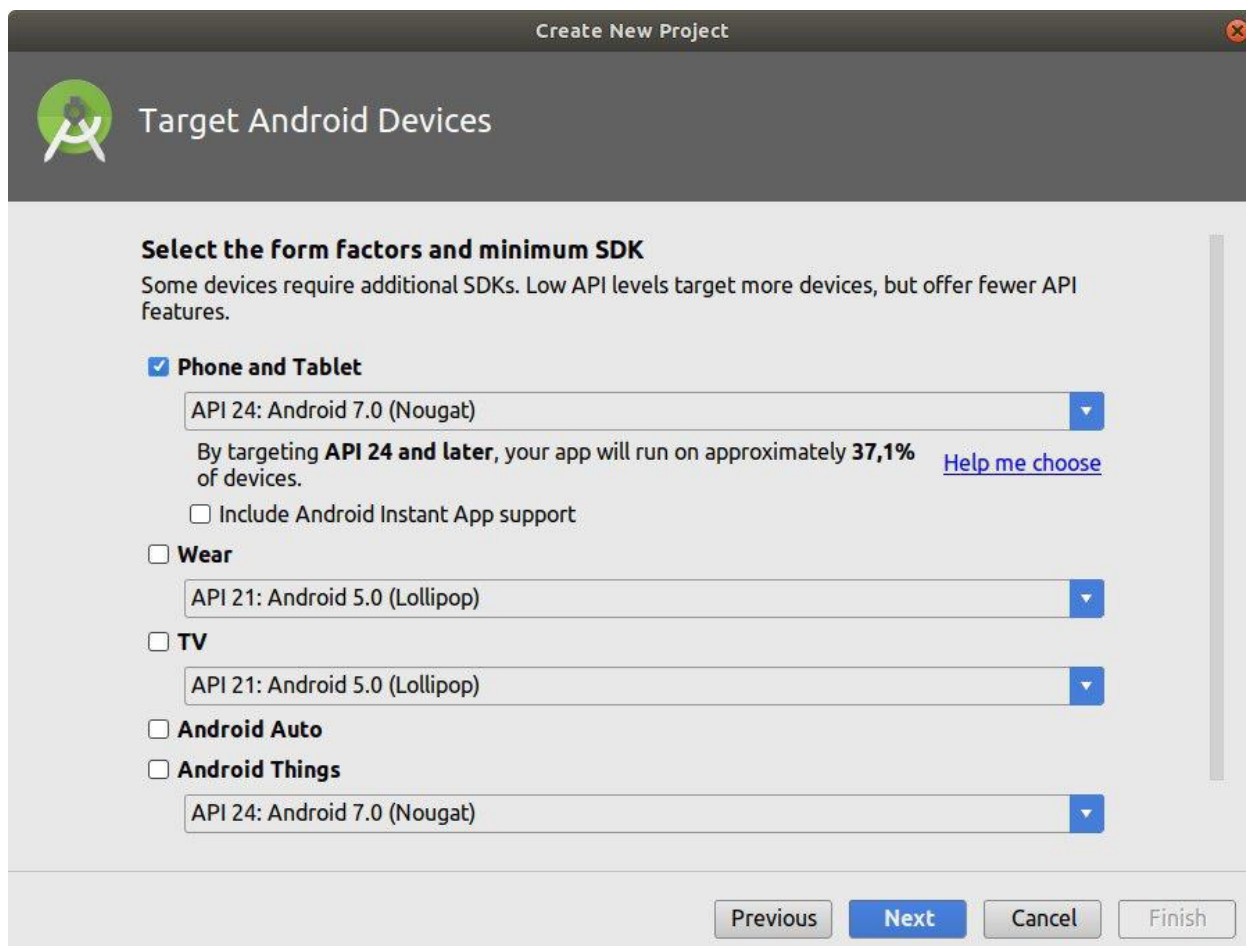


Рис. 3.5

Після цього необхідно вибрати шаблон головної активності(activity). Я обираю BlankActivity – це самий базовий шаблон, на якому присутній мінімальний набір view-компонентів: view-контейнер(ConstraintsLayout) для інших елементів, панель інструментів (Toolbar), та кнопка дії (FloatingActionButton). 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.

3.2.3 Налаштування ARCore

У файлі сценарія gradle потрібно включити Java 8 і додати залежність для Sceneform[24,33].

Sceneform використовує мовні конструкції з Java 8. Для проектів, які мають міні-API рівні менше 26, вам необхідно чітко додати підтримку для Java8.

У розділі *android {}* файла *app/build.gradle* необхідно додати:

```
compileOptions {  
    sourceCompatibility JavaVersion.VERSION_1_8  
    targetCompatibility JavaVersion.VERSION_1_8  
}
```

У файлі *app/build.gradle* необхідно додати залежності для API Sceneform та елементів Sceneform UX.

У розділі з *dependency {}* додається наступна залежність:

```
implementation "com.google.ar.sceneform.ux:sceneform-ux:1.4.0"
```

Далі не обхідно запустити синхронізацію в Gradle, щоб оновити конфігурацію і підтягнути бібліотеку з репозиторія.

3.2.4 Встановлення Google Sceneform Tools (Beta) плагіна

Для встановлення плагіна необхідно виконати наступні кроки. В Android Studio відкрити вікно **Plugins** :

- Windows: **File > Settings > Plugins > Browse Repositories**
- macOS: **Android Studio > Preferences > Plugins**

Потім натиснути **Browse repositories** і встановити *Google Sceneform Tools (Beta)*(рис.3.6). Після встановлення AndroidStudio потребує перезавантаження. Після перезавантаження плагін готовий до роботи. Для запуску необхідно відкрити вкладку праворуч на панелі інструментів Android Studio, в якій відкриється вікно рендеринга 3d об'єкта і текстовий файл коду 3d об'єкта.

Змн.	Арк.	№ докум.	Підпис	Дата

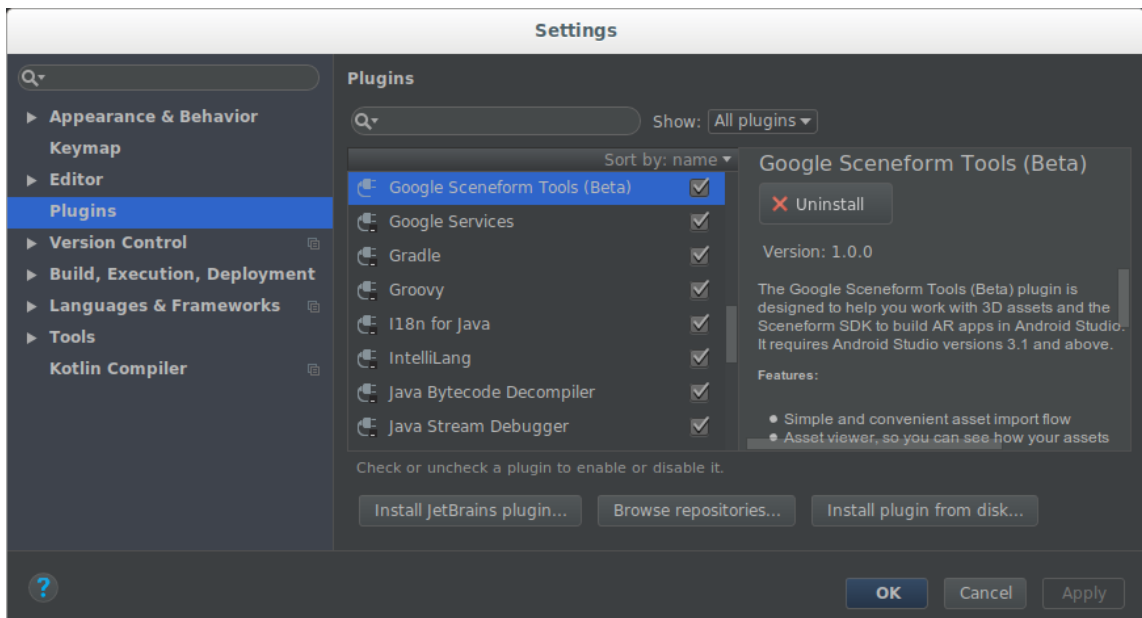


Рис. 3.6. Встановлення Google Sceneform Tools (Beta) плагіна

3.2.5 Налаштування AndroidManifest.xml

Файл маніфесту `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 маніфесту прописати наступне.

Відкрити `app/manifests/AndroidManifest.xml` і в секції `<manifest>` додати такі елементи:

```
<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"/>
<uses-feature android:name="android.hardware.camera.ar" android:required="true" />
<meta-data android:name="com.google.ar.core" android:value="required" />
```

Вони вказують, що програмою буде використана камера, а також використані функції AR, що вимагає наявності і запуску ARCore. Останній запис про метадані використовується Play Market, щоб відфільтрувати програми для користувачів на пристроях, які не підтримуються ARCore.

3.3 Архітектура програмного забезпечення

3.3.1 Життєвий цикл роботи ARCore додатка на Android

Життєвий цикл роботи ARCore додатка на Android[34] у загальному випадку складається з наступних етапів:

- Перевірка залежностей ARCore (ARCore SDK та дозволи на камеру) виконується перед запуском ARCore Session
- Відображення опорних точок (*features points*)
- Відображення виявлених планів (*planes*)
- Захоплення користувацьких жестів для виконання хіт тестів
- Створення анкерів(якорів)
- Прикріплення віртуальних моделей до якорів

Блок-схема(рис. 3.7) забезпечує високо-рівневий огляд того, що робить додаток:

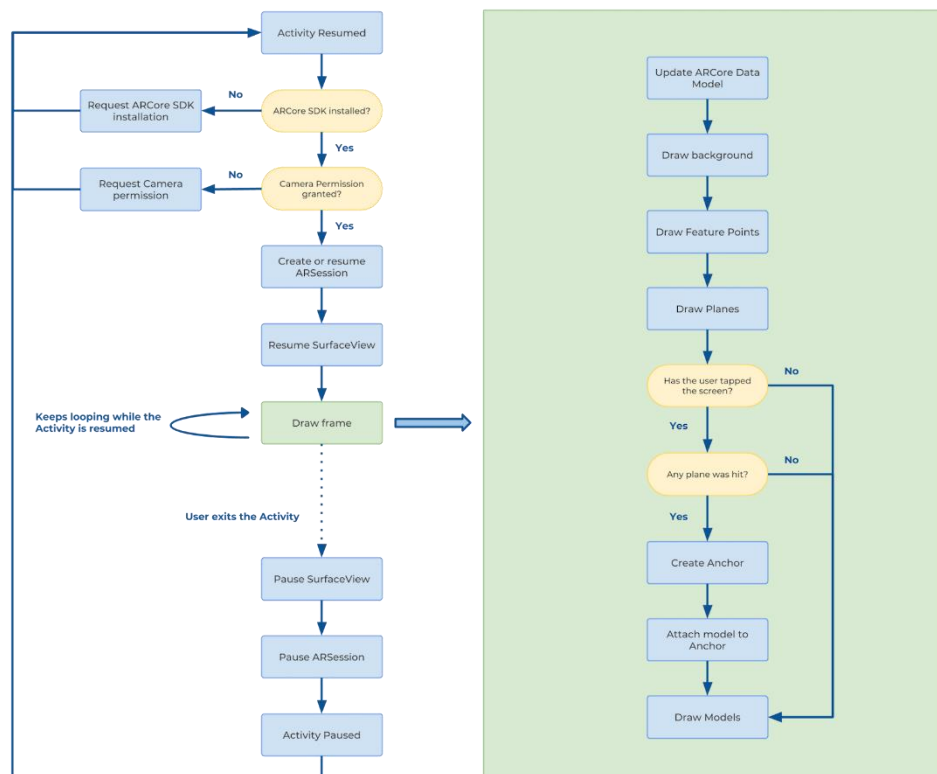


Рис. 3.7. Життєвий цикл роботи ARCore додатка на Android

3.3.2 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 [35], 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 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.

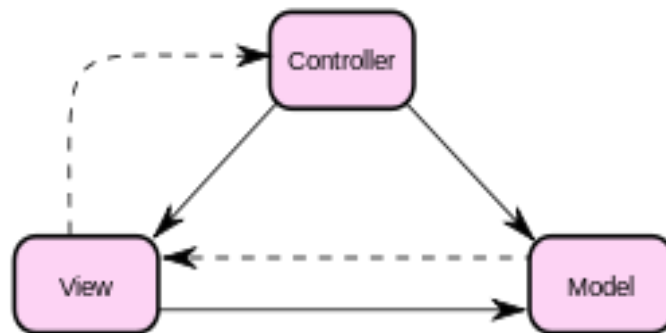


Рис. 3.8. Структура MVC

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.

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:

FurnitureNode.java - клас, що розширює TransformableNode, і зверігає інформацію про 3d об'єкт, позу об'єкта в тривимірному просторі і «карточку» у формі іншого Node класа;

FurnitureItem.java - POJO клас, визначає інформацію про 3d модель: ідентифікатор в системі, назва моделі, опис моделі, зображення моделі, шлях до файлів 3d об'єкта.

NodesHelper.java - клас для зберігання і керування списком 3D об'єктів. Клас реалізує шаблон програмування Singleton.

FurnitureItemsHelper.java - клас для зберігання і керування списком 3D моделей. Клас реалізує шаблон програмування Singleton

Діаграма класів моделі зображена на рис. 3.9.

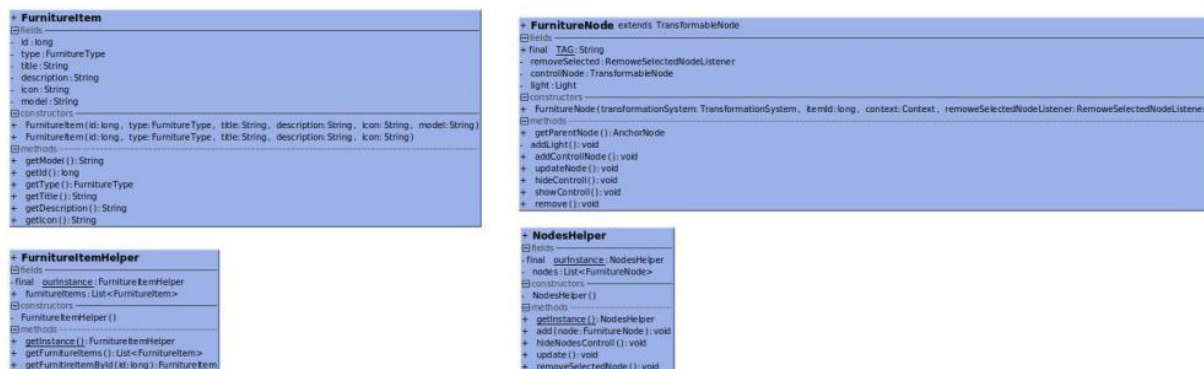


Рис. 3.9. Діаграма класів моделі

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:

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

SettingsActivity.java – клас, що наслідує клас Activity. Використовується для налаштування параметрів додатка.

MyArFragment.java – клас, що розширює Sceneform ArFragment. Перевизначає абстрактний метод на запит дозволів на використання функцій файлової системи та камери. Встановлює параметри сцени.

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



Рис. 3.10. Діаграма класів контролера

PhotoDialog.java - 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.

FurnitureListAdapter.java – клас-адаптер що зв’язує інформацію про 3D модель із елементом списку. Використовується при виводі всіх доступних 3D моделей в системі.

RemoveSelectedNodeListener.java – інтерфейс, який визначає метод видалення Node об’єкта зі сцени.

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:

MyFootprintSelectionVisualizer.java – клас, визначає Renderable об’єкт, для позначення активного Node на сцені.

PointerDrawable.java – клас, що розширяє Drawable клас, для відображення вказівника.

activity_main.xml - визначає представлення класу контролера MainActivity. Дане представлення доповнюється компоненти FloatingActionButton і секцією <include> для відображення файлу content_main.xml.

content_main.xml - відображає основні елементи інтерфейсу додатка. Вміщує AR fragment, список RecyclerView, елементи якого додаються динамічно, в Java коді. Також тут розміщені view-компоненти, які реалізують меню додатку.

fragment_photo.xml - визначає представлення класу контролера PhotoDialog. Розміщені view-компоненти, які реалізують основні функції при роботі зі скриншотом.

furniture_item_card.xml – визначає вигляд елементів списку 3d моделей.

controll_renderable.xml – визначає вигляд «карточки» 3d об'єкта, який використовується елементом Sceneform ViewRenderable.

footprint_renderable.xml – визначає представлення класу MyFootprintSelectionVisualizer.java.

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

3.4 Інтеграція ресурсів в додаток

Файли ресурсів в Android проекті можуть міститися в двох директоріях: /res і /assets(активи)[36]. 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.

					ІАЛЦ.467100.003 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

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 файлової системі за допомогою Java. Також активи доступні для використання тільки при необхідності, навідміну від даних res каталогу, які завантажуються при запуску додатку.

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. Власне для завантаження і налаштування 3d моделей використовується Google Sceneform Tools (Beta) плагін. Спочатку необхідно скопіювати файти 3d моделей і прилегли до них файли (текстури, матеріали та ін.) у каталог sampledata. Далі вибрати файл необхідної моделі і натиснути *Import Sceneform Asset*. Далі відкриється діалогове вікно (рис 3.11). В ньому необхідно ввести наступні дані. *Source Asset Path* – шлях до файлу об'єкта OBJ, FBX або glTF 3D моделі для імпорту[37]. *Material Path* – шлях, до файлу з матеріалами, які використовуються в 3d об'єкті; default вказує, що Sceneform використовує вбудований матеріал за замовчуванням. Можна замість цього вказати шлях до спеціального матеріалу * .mat файл[38]. *.sfa Output Path* – шлях, куди буде згенеровано .sfa файл. Можна використовувати за замовчуванням або вказувати інший шлях у папці sampledata. Це гарантує, що * .sfa не буде включено до вашого APK. *.sfb Output Path* - шлях, куди буде згенеровано .sfb файл. За замовчуванням використовується папка src / main / assets /, яка дозволяє використовувати довільні імена файлів активів. Якщо ім'я файлу (без розширення файлу) є дійсним ідентифікатором ресурсу (наприклад, R.raw.filename), можна замість цього скористатись каталогом src / main / res / raw /.

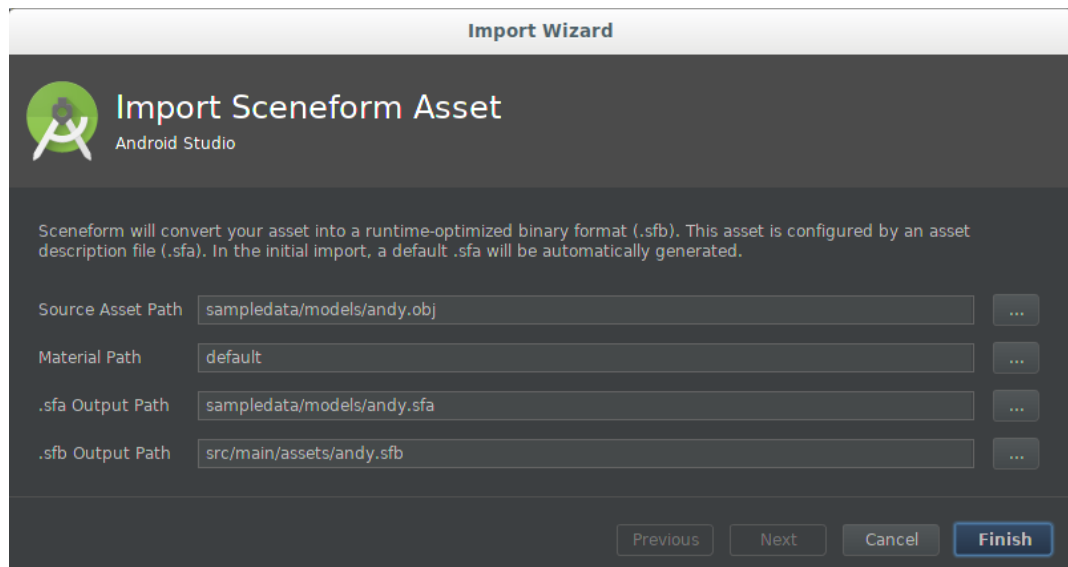


Рис. 3.11. Імпорт 3d моделей

Далі натиснути *Finish*, щоб завершити процес імпорту. Після цього відкриється вкладка плагіна, з попереднім зображенням 3d об'єкта у тривимірному просторі(файл .sfb) і текстовий файл .sfa, в якому можна відредагувати деякі параметри, такі як масштаб, поворот 3d моделі та ін.

3.5 Збірка проекту

3.5.1 Генерація APK

По замовчувню а AndroidStudio використовується автоматизована система збірки проектів Gradle. Щоб запустити збірку проекту необхідно

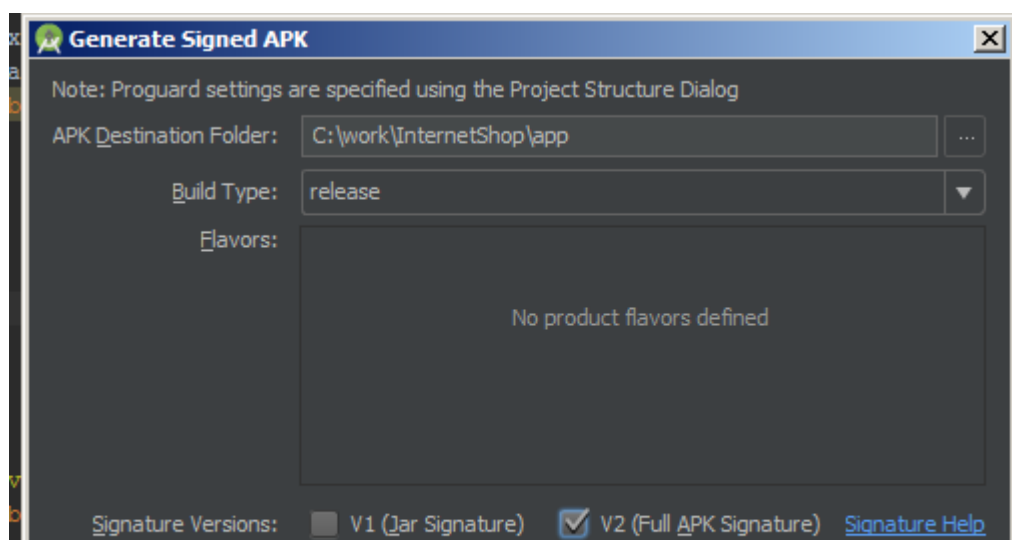


Рис. 3.12 Завершення створення APK-файлу

виконати команду *gradle-build*. Після цього згенерований APK файл[39](рис. 3.12) можна буде знайти в директорії */build* проекту.

3.6 Опис реалізованого програмного забезпечення

Після запуску додатку перевіряється відповідність до вимог програмної та апаратної частини девайсу. Якщо ж вимоги не задовільняються, програма завершить свою роботу з виводом даної інформації в Toast-повідомленні. Якщо девайс відповідає вимогам апаратної частини (наявність *android.hardware.camera.ar*) то система покаже діалог, для встановлення останньої версії ARCore (Рис. 3.13). Далі система показує діалог підтвердження доступу до камери і доступу до файлової системи[40] (для можливості зберегти знімок в пам'ять девайсу)(Рис 3.14.)

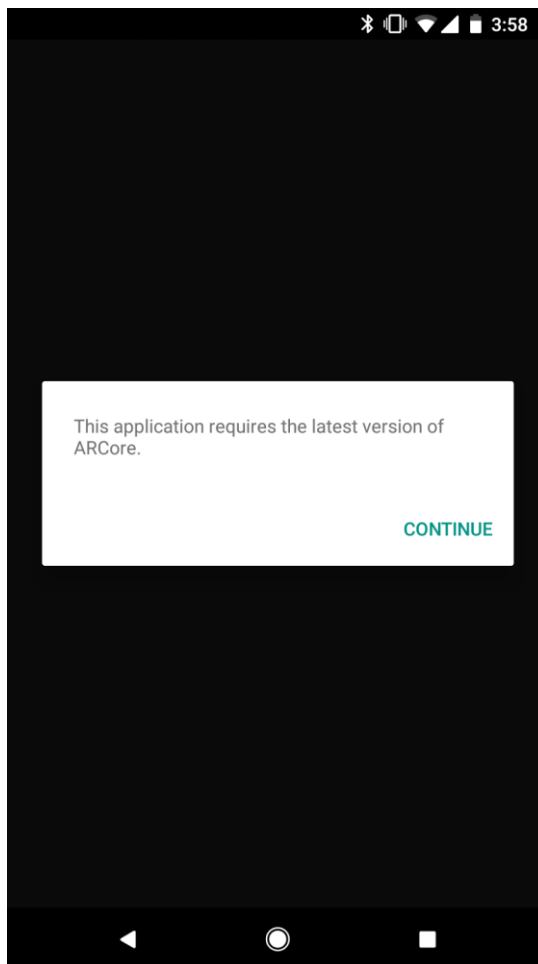


Рис. 3.13. Діалог встановлення ARCore

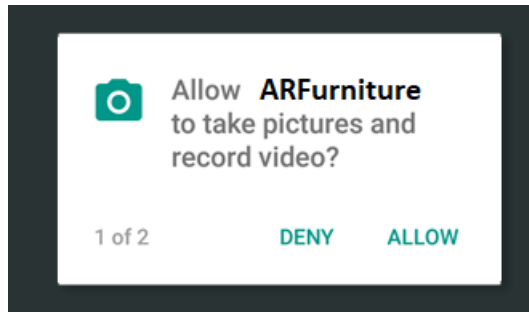


Рис. 3.14. Діалог запиту дозволів

Після запуску додатку з'являється головне вікно, на якому знаходиться ARSceneView елемент для відображення потоку даних з камери. Знизу



Рис. 3.15 Рис.Інтерфейс додатку:

- 1 – сцена
- 2 – список 3d об'єктів;
- 3 – кнопка «Screenshot»;
- 4 – панель налаштування видимості
- 5 – кнопка «Settings».

знаходиться список всіх доступних 3d об'єктів та кнопка для виконання фото сцени. Зверху розташована панель на якій знаходиться кнопка налаштувань та кнопки управління видимістю (рис. 3.15).

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

Після запуску додатку на сцені з'явиться анімоване зображення(рис. 3.16), яке вказує, що необхідно виконати калібрування сенсорів та камери повторивши відповідні рухи. Після успішного виконання калібрування дане зображення зникне з екрану і додаток стане готовим до роботи.



Рис. 3.16. Калібрування

Для демонстрації роботи додатку необхідно навести девайс на плоску поверхню (підлога, стіни, поверхня столу тощо), щоб камера захопила його. Як тільки система визначить плоску поверхню поверх неї відрисується текстурована поверхня зеленого відтінку (Рис 3.17).

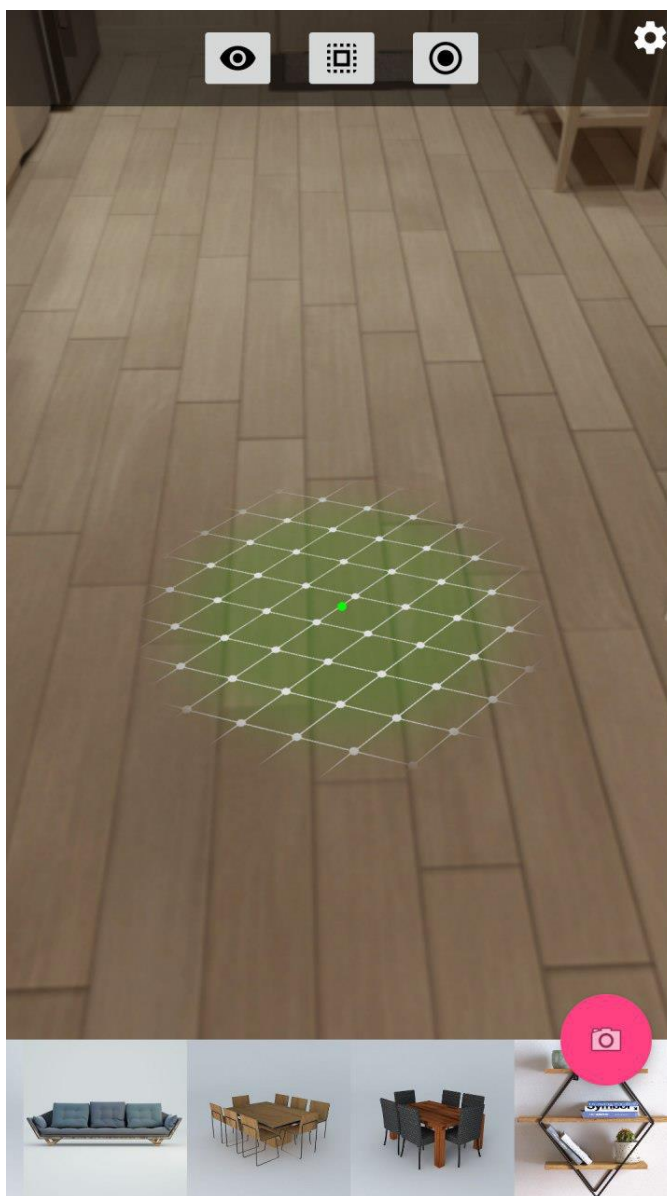


Рис. 3.17. Розпізнана поверхня

Для визначення положення, куди буде поміщено 3d об'єкт, використовується вказівник. Він розташований посередині екрану, і представляє собою зелену точку (рис 3.17). Він може знаходитися в 2 станах: активний і пасивний. За активного стану від має форму зелені точки і дозволяє поміщувати 3d об'єкт на сцену, у випадку пасивного, це значить, що 3d об'єкт

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

62

помістити на сцену не можна, так як його основа буде не на розпізнаній поверхні. Пасивний режим відображається чорним хрестиком.

Після успішного розпізнавання площини можна вибрати необхідну модель з каталогу об'єктів, що розташовані знизу екрану, після чого «будується» 3D модель на позиції, куди вказує вказівник (рис. 3.18).

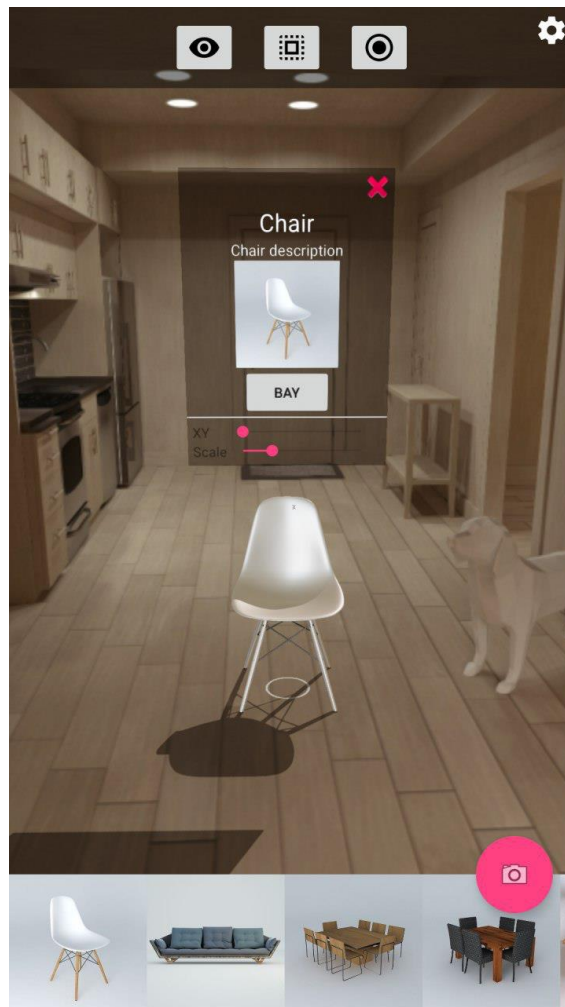


Рис. 3.18. 3d модель стільця на сцені

Для демонстрації доповненої реальності необхідно змінити кут огляду поверхні або відстань до неї, перемістивши смартфон в просторі, як би відбувалась зйомка реального об'єкта. В першому випадку 3D модель повернеться на відповідний кут, в другому – вона зміниться в розмірах(рис. 3.19). Система також враховує освітленість навколишнього середовища, і відкидає тіні від об'єктів.



Рис. 3.19. 3D об'єкт після модифікацій

Разом з 3d об'єктом на сцені з'являється його «карточка» в тримимірному просторі на якій відображаються короткі відомості про модель (назва, короткий опис і фото), кнопка «Купити», з якою можна також взаємодіяти, як із звичайною кнопкою, після натиску на яку, відкриється сторінка браузера з адресою товару, панель керування з можливістю змінювати масштаб моделі і її поворот в ХУ площині, а також кнопка видалення 3d об'єкта зі сцени. Взаємодія з об'єктами на сцені також можлива через відповідні жести на сенсорному екрані.

У випадку декількох об'єктів на сцені можна вибрати активний об'єкт(він буде підсвічений білим колом знизу) і «карточка» буде видима лише у активного об'єкта (Рис. 3.20).

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

64



Рис. 3.20. Сцена з декількома об'єктами

Активний об'єкт також можна переміщувати по поверхні до якої він належить.(Рис. 3.21)

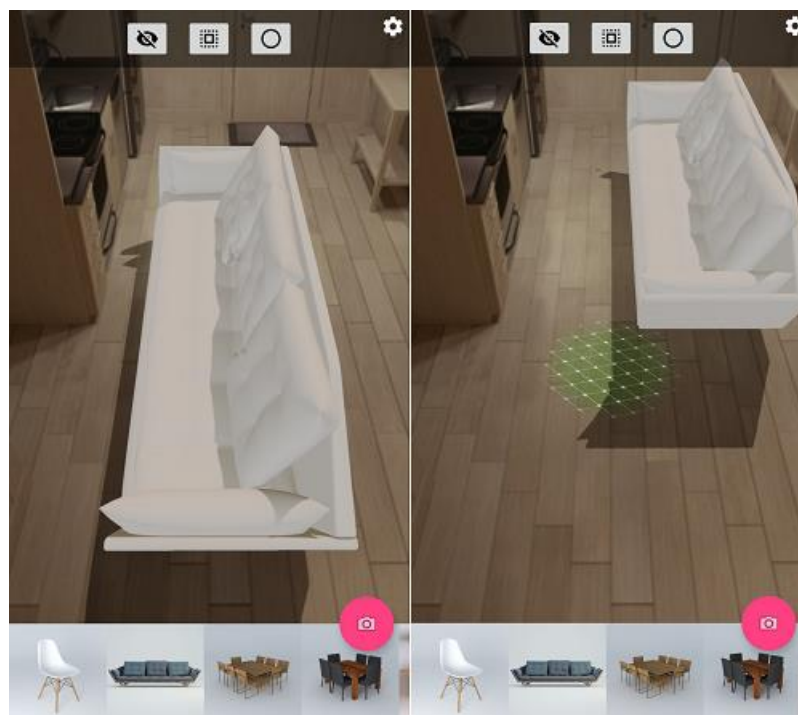


Рис. 3.21. Переміщення об'єктів на сцені

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

65

Панель зверху екрану(рис. 3.22) налаштовує видимість окремих елементів не сцені і елементів керування. Кнопка «Hide Cards» відповідає за видимість «карточок» елементів. Кнопка «Hide Plane» відповідає за видимість поверхні (ромбовидної решітки). Кнопка «Hide Pointer» відповідає за видимість вказівника. Всі ці конфігурації допомагають сховати всі штучні елементи, щоб підвищити реалістичність сцени.



Рис. 3.22. Керування видимістю

Якщо натиснути кнопку «ScreenShot» на якій зображена камера, на панелі знизу, то додаток зробить знімок з камери з накладеними поверх нього 3D об'єктами (рис. 3.13). Фінальне зображення буде показане в відповідному діалоговому вікні.

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



Рис. 3.23. Знімок екрану

Також отриманим зображенням можна поділитись з друзями в соціальних мережах натиснувши кнопку «Share» (рис. 3.24).

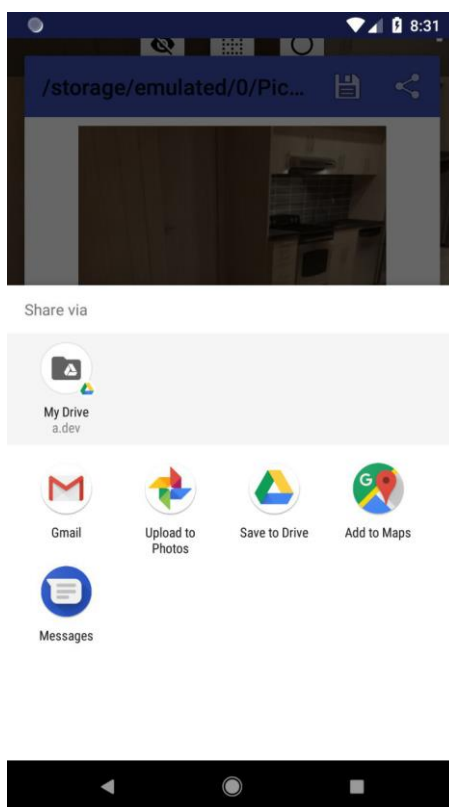


Рис. 3.24

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

67

При натисненні кнопки «Settings» в меню додатку, відкривається нове вікно налаштувань додатку(рис. 3.25). В якості налаштувань реалізовано вибірку камери а також розширення камери.

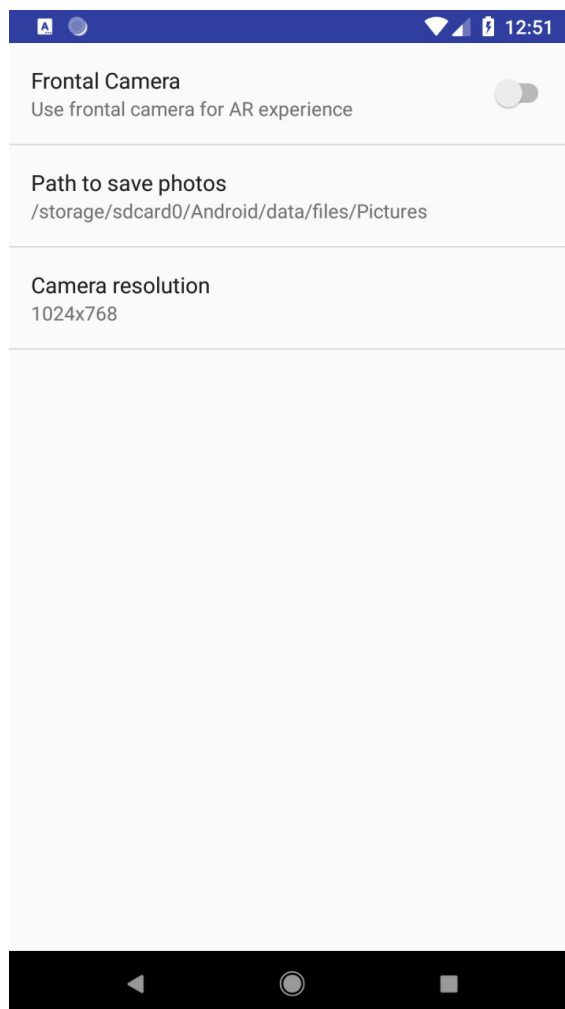


Рис. 3.25. Вікно налаштувань

Висновки до розділу

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.

Описано процес розробки архітектури додатку, інтеграції ресурсів і генерації вихідного APK файлу.

В ході тестування розробленої програми виявилось, що вона має наступні характеристики:

- На точність розпізнавання поверхонь (*planes*) сильно впливає освітленість середовища, в якому відбувається тестування: при недостатньому освітленні додаток має низький рівень розпізнавання поверхонь.
- На точність розпізнавання поверхонь (*planes*) також сильно впливає якість відеокамери встановленої на мобільному пристрої.
- При рендерингу декількох 3D моделей одночасно відбуваються короточасні зависання дисплею, що пов'язано з обмеженими обчислювальними потужностями мобільних пристроїв.
- Поверхні (*planes*), що не мають вираженої текстурованої поверхні, розпізнаються системою, дещо гірше, ніж ті в яких явно виражена текстура. Це пов'язано з тим, що системі складніше визначати опорні точки (*features points*), і відповідно відслідковувати їх переміщення.

РОЗДІЛ 4

СТАРТАП ПРОЕКТ

4.1 Опис ідеї проекту

Таблиця 4.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Мобільний додаток для моделювання інтер'єру	Застосування у торговій мережі з продажу меблів	Попередній перегляд потенційної покупки у реальних умовах
	Застосування для планування майбутнього інтер'єру приміщення	Попередній перегляд майбутньої обстановки, без необхідності реальних змін

Таблиця 4.2

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W	N	S
		Мій проект	IKEA	Smart Reality	Augment			
1	Вартість експлуатації	Безкоштовно	50 тис. грн/міс	25 тис. грн/міс	5 тис. грн/міс	-	-	+
2	Вартість обслуговування	0,5 тис. грн/міс	10 тис. грн/міс	5 тис. грн/міс	5 тис. грн/міс	-	-	+
3	Наявність веб -інтерфейсу	-	+	-	+	-	+	-
4	Наявність програмного коду у відкритому доступі	+	-	-	-	-	-	+
5	Використання безмаркерної технології	+	-	-	-	-	-	+
6	Підтримка різних мобільних платформ	-	+	+	+	+	-	-
7	Підтримка VR	-	-	+	-	-	+	-
8	Модифікація в режимі реального часу	+	+	+	+	-	+	-
9	Широкий каталог	-	+	-	+	-	+	-
10	Можливість розширити каталог	+	+	+	+	-	+	-
11	Необхідність спеціального програмного забезпечення	+	-	-	-	-	+	-
12	Необхідність спеціального апаратного забезпечення	+	-	-	-	+	-	-
13	Необхідність додаткових матеріалів для роботи	-	+	+	+	-	-	+

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3

Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	SDK доповненої реальності	ARCore	Наявна	Доступна
2		Vuforia	Наявна	Недоступна
3		ARKit	Наявна	Недоступна
4		Artoolkit	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: ARCore				
1	Мова програмної реалізації	Java, Kotlin	Наявна	Доступна
2		C/C++	Наявна	Доступна
3		Python, JavaScript	Наявна	Недоступна
Обрана технологія реалізації ідеї проекту: ARCore				
1	Метод поширення	Google Play	Наявна	Доступна
2		App Store	Наявна	Недоступна
3		Amazon	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: Java, Kotlin				

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3-4
2	Загальний обсяг продаж, грн/ум.од	100
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	6-7

Таблиця 4.5

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Застосування у торговій мережі з продажу меблів	Малий бізнес Середній бізнес	Наявність вимог до високої доступності та відмовостійкості	Широкий каталог товарів, доступність
2	Планування інтер'єру приміщення	Звичайні користувачі	Відсутність практично будь-яких вимог	Доступність

Таблиця 4.6

Фактори загрози

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Крадіжка інтелектуальної власності	Крадіжка ідеї або реалізації	Забезпечення якіснішого захисту інформації
2	Відсутність ринку	Відсутність шляху збуту товару	Пошук ринку збуту на зарубіжному ринку Залучення експертів та менторів
3	Неякісна реалізація	Продукти конкурентів мають кращу реалізацію	Оптимізація, доопрацювання продукту, залучення спеціалістів
4	Недостача капіталовкладень	Витрачені усі кошти до моменту виходу на ринок	Пошук нових інвестицій

Таблиця 4.7

Фактори можливостей

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Отримання інвестицій	Отримання додаткового капіталу	Більш якісна реалізація товару
2	Успішна маркетингова кампанія	В результаті проведеної маркетингової кампанії з'явилась висока зацікавленість клієнтів	Підтримка стабільної роботи системи та розширення системи; Використання подібної маркетингової стратегії надалі;
3	Поглинання конкурентами	Пропозиція купівлі проекту або розроблених технологій одним із конкурентів	Розвиток розроблених технологій; Оцінка вартості розроблених технологій;

Таблиця 4.8

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Олігополія	Незначна кількість конкурентів; Схожість технологій;	Інформування ринку щодо появи нової системи візуалізації інтер'єру
Галузевий	Загроза появи нових конкурентів; Висока потреба у товарі	Інформування ринку щодо якості використовуваної новаторської технології
Внутрішньогалузева	Діяльність в одній галузі економіки Надання сервісів одного типу	Зменшення вартості товару Примноження каналів збуту

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Товарно-видова	Надання різних сервісів одного типу	Маркетингова політика
Нецінова	Використання цін для покращення економічних умов збуту	Зменшення вартості платформи Використання нових каналів розподілу
Марочна	Пропозиція схожої платформи Спільна цільова аудиторія	Інформування ринку щодо появи нової платформи управління хмарною інфраструктурою

Таблиця 4.9

Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Товари-замінники
Складові аналізу	IKEA, SmartReality, Augment	Розмір капіталовкладень; Забезпечення гнучких цін; Доступ до каналів розподілу, Витрати на масштабах	відсутні	Копіювання функціоналу, Монополізація дистриб'юторів, Демпінгування
Висновки	В даній галузі не висока конкуренція ринку	Можливості входу на ринок забезпечить мінімізація цін, швидкість та простота надавання послуги споживачам і співпраця із головними гравцями ринку	відсутні	Пропонування вигідних умов дистриб'юторам, забезпечення захисту інтелектуальної власності; гнучкість цінової політики

Таблиця 4.10

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Інноваційна технологія	Розроблений продукт використовує інноваційну технологію порівняно з конкурентами
2.	Цінова політика	Додаток має бізнес модель Freemium, що дозволяє всім користувачам отримувати базовий набір послуг.

Таблиця 4.11

Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів- конкурентів у порівнянні ARFurniture						
			-3	-2	-1	0	+1	+2	+3
1.	Унікальність сервісу	18							+
2.	Цінова політика	13					+		

Таблиця 4.12

SWOT- аналіз стартап-проекту

Сильні сторони: Якість та продуктивність Низькі ціни Інноваційність	Слабкі сторони: Нестача капіталовкладень
Можливості: Інвестиції Висока зацікавленість цільової аудиторії	Загрози: Крадіжка інтелектуальної власності Неякісна реалізація Відсутність ринку

Таблиця 4.13

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка власних технологій доповненої реальності	Малоймовірне	24 місяці
2.	Маркетингова кампанія для приваблювання користувачів	Малоймовірне	2 місяці
3.	Пропонування безкоштовних тарифів	Ймовірне	1 місяць
4.	Пошук бізнесів інших галузей для співпраці	Дуже ймовірне	6 місяців
Обрана альтернатива: Пошук бізнесів іншої галузі для співпраці			

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

74

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.14

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Адміністратори інфраструктур малого бізнесу	Висока	65%	Середня	Низькі бар'єри входу
2.	ІТ-підрозділи середнього бізнесу	Мала	28%	Середня	Високі бар'єри входу
3.	Звичайні користувачі	Середня	45%	Середня	Високі бар'єри входу
Які цільові групи обрано: адміністратори інфраструктур малого бізнесу					

Таблиця 4.15

Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Надання платформи малому та середньому бізнесу	Вибірковий розподіл	Здатність протистояти прямим конкурентам; Низькі витрати; Ефективна співпраця;	Стратегія диференціації

Таблиця 4.16

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Шукати нових споживачів та забирати існуючих у конкурентів	Кросплатформенність; Динамічне розширення каталогу моделей; Наявність WEB інтерфейсу	Стратегія наслідування лідера

Таблиця 4.17

Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Відповідність затвердженим характеристикам; Висока якість реалізації; Простий інтерфейс; Гнучка цінова політика; Оперативна підтримка	Стратегія диференціації	Формування регулярного попиту; Виявлення нових груп споживачів; Нові напрями застосування існуючої послуги;	Інноваційність технології; Низькі ціни; Простота використання;

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Попередній перегляд товару в реальній обстановці	Дозволяє оглянути потрібний товар у реальних умовах експлуатації	Інноваційність технологій що використовуються Простота використання Цінова перевага
2	Попередній перегляд можливих конфігурацій	Дозволяє переглянути можливі варіанти інтер'єру	Інноваційність технологій що використовуються Простота використання Цінова перевага

Таблиця 4.19

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмний продукт що надає можливість візуалізації інтер'єру		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Кількість		1 шт.
	Якість: стандарти якості постачання програмних продуктів		
	Пакування: apk файл		
	Марка: ARFurniture		
III. Товар із підкріпленням	Програмний продукт		
	Програмний продукт, технічна підтримка та підписка на оновлення		
За рахунок чого потенційний товар буде захищено від копіювання: застосування механізму оптимізації і обфускації Java коду			

Таблиця 4.20

Визначення меж встановлення ціни

№ п/ п	Рівень цін на товари замінники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	До 1 тис. грн	До 1 тис. грн	До 10 000 грн./міс.	0,5 -1 тис. грн

Таблиця 4.21

Формування системи збуту

№ п/ п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Закупівля здійснюється через довірені джерела	Інформування користувачів Доступ користування сервісом	Канал одного рівня	Селективна з використанням комбінованого каналу збуту

Таблиця 4.22

Концепція маркетингових комунікацій

№ п/ п	Специфіка поведінки цільових клієнтів	Канали комунікації , якими користують- ся цільові клієнти	Ключові позиції, обрані для позиціону-вання	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Автоматиза- ція бізнес- процесів Вимоги до високодосу- пності та відмовостій- кості	Прямі офіційні	Послідовність в реалізації обраної позиції Доступність та об'єктивність інформації про фірму і товар Унікальність послуги	Інформування користувачів про появу нового продукту Інформування користувачів про властивості та переваги продукту Інформування користувачів про нові способи використання відомого продукту Виправити у користувачів неправильні представлення про продукт	Раціоналістич- на стратегія реклами

Змн.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

77

Висновки до розділу

В результаті проведеної розробки стартап-проекту було отримано список очікувань користувачів системи візуалізації інтер'єру, були проведені технологічний аудит та аналіз ринкових можливостей запуску проекту, розроблені ринкова стратегія та маркетингова програма. В цілому, існує можливість ринкової комерціалізації проекту, наявний попит на такого роду системи. Враховуючи не високу конкуренцію та інноваційність використовуваної технології проект є конкурентноспроможним. Нажаль, через високу трудомісткість, більш перспективною є реалізація ідеї великими ІТ компаніями України.

					ІАЛЦ.467100.003 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

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

Був проаналізований та досліджений стан технологій створення мобільних додатків доповненої реальності. Були розглянуті популярні бібліотеки для роботи з доповненою реальністю. Проаналізовані їхні плюси й мінуси і вибрано технологію ARCore для розробки додатку. Були обрані платформа Android, мови програмування Java, середовище розробки Android Studio. В якості архітектури мобільного додатку було обрано архітектурний паттерн програмування MVC.

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. Для коректного використання розробленого додатку необхідне виконання наступних умов: наявність сертифікованого девайсу, необхідна відеокамера хорошої якості та сенсори висової чутливості на мобільному пристрої; в середовищі використання важливим елементом є освітленість.

Перевагою даного підходу реалізації доповненої реальності порівняно з маркерною технологією є те, що для демаонстрації не потрібно розробляти зображення-маркери і зв'язувати їх з 3d моделями, також суттєвою перевагою є те, що технологія ARCore використовує не лише методи комп'ютерного бачення а й застосовує сенсори, для трекінга переміщення, що дозволяє рендерити 3d об'єкти навіть тоді коли розпізнана площина знаходиться поза полем зору камери.

Було проведено розробку стартап проекту. Враховуючи не високу конкуренцію та інноваційність використовуваної технології проект є конкурентноспроможним. Нажаль, через високу трудомісткість, більш перспективною є реалізація ідеї великими ІТ компаніями.

					ІАЛЦ.467100.003 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

ЛІТЕРАТУРА

1. М. О. Макогон,, ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ И ЕЕ ПРИМЕНЕНИЕ [Электронный ресурс] / М. О. Макогон,, А. С. Долгих. – 2014. – Режим доступа до ресурсу: https://buk.irk.ru/library/sbornik_14/makogon.pdf.
2. Что такое смешанная реальность (MR) и почему ее нельзя путать с VR и AR [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://www.crn.ru/news/detail.php?ID=129171>.
3. If You're Not Seeing Data, You're Not Seeing, Brian X. Chen [Электронный ресурс]. – 2009. – Режим доступа до ресурсу: <http://www.webcitation.org/61DG5huig>.
4. Ronald T. Azuma A Survey of Augmented Reality // In Presence: Teleoperators and Virtual Environments. – 1997. – No 4. – P. 355–385.
5. Miika Tikander. Development and evaluation of augmented reality audio systems: Abstract of dissertation for the degree of Doctor of Science in Technology. – Helsinki, 2010.
6. Joseph Rozier. Hear&There: An Augmented Reality System of Linked Audio / Joseph Rozier, Karrie Karahalios, Judith Donath // Online Proceedings of the ICAD, 2009
7. Karrie Karahalios. GaitAid Virtual Walker for Movement disorder patients Augmented Reality, 2007
8. PHANTOM Premium [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <http://www.sensable.com/haptic-phantom-premium-6dof.htm>.
9. Класифікація та перспективні напрямки використання технології доповненої реальності. // ZPRÁVY VĚDECKÉ IDEJE - 2018. – 2018. – №6. – С. 27–32.
10. Савчук А. Ю. Дослідження інструментальних засобів розробки ігрових програм з елементами доповненої реальності. : дис. канд. техн. наук : 8.050102 / Савчук Артем Юрійович – Київ, 2015. – 74 с.
11. ARCore. Overview [Электронный ресурс] – Режим доступа до ресурсу: <https://developers.google.com/ar/>.
12. OpenCV [Электронный ресурс] – Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/OpenCV>.

- 13.Vuforia [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Vuforia>.
- 14.Artollkit About [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://archive.artoolkit.org/>.
- 15.Android [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Android>.
- 16.IOS [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/IOS>.
- 17.KaiOS [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/KaiOS>.
- 18.Windows Phone [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Windows_Phone.
- 19.How Reality Technology is Being Used in Design [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <http://www.realitytechnologies.com/design>.
- 20.Реальность на ладони: обзор приложения Augment [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <http://arnext.ru/reviews/realnost-na-ladoni-obzor-prilozheniya-augment-3650>.
- 21.ARCore Overview [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://developers.google.com/ar/discover/>.
- 22.Fundamental Concepts [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://developers.google.com/ar/discover/concepts>.
- 23.Supported Devices [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://developers.google.com/ar/discover/supported-devices>.
- 24.Sceneform Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/ar/develop/java/sceneform/>.
- 25.Java [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Java>.
- 26.Android Studio [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://developer.android.com/studio/>.

- 27.Import and Preview 3D Assets [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/ar/develop/java/sceneform/import-assets>.
- 28.Gradle [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Gradle>.
- 29.Git [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Git>.
- 30.Welcome to the OpenGL Wiki [Електронний ресурс] – Режим доступу до ресурсу: <https://www.khronos.org/opengl/wiki/>.
- 31.OBJ [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Obj>.
- 32.FBX [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/FBX>.
- 33.Enable ARCore [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/ar/develop/java/enable-arcore>.
- 34.Getting started with Google ARCore on Android [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.novoda.com/getting-started-with-google-arcore-on-android/>.
- 35.Model-View-Controller [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Model-View-Controller>.
- 36.Asset and res/raw Android difference [Електронний ресурс]. – 2608. – Режим доступу до ресурсу: <https://www.programering.com/a/MTM0QzMwATg.html>.
- 37.Sceneform asset file reference [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/ar/develop/java/sceneform/sfa>
38. Custom Material Reference [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://developers.google.com/ar/develop/sceneform/custom-material>.
- 39.APK [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/APK>.
- 40.Request App Permissions [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/training/permissions/requesting>.