ВСТУП

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

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

Ця технологія має безпосередній вектор у майбутнє. Її застосування можна легко представити у навігації. Наприклад, використовуючи смартфон

(обладнаний камерою), можна буде без проблем знайти необхідний шлях, навіть якщо у вас є проблеми з орієнтацією на карті. Такий результат може бути досягнений завдяки синхронній роботі систем глобального позиціонування, завантаженим, на смартфон, картам, і алгоритмам, що зможуть прив’язати данні з карти на зображення, що отримується з камери смартфону. Як результат, ми зможемо побачити виділений шлях (наприклад, зеленою лінією), що буде проектуватися у реальність і вказуватиме необхідну користувачеві дорогу.

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

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

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

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

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

З розвитком технологій доповнена реальність все глибше проникатиме в наше повсякденне життя. Зараз створюється і накопичується все більша і більша кількість даних і AR буде служити мостом для доступа до величезної кількості інформації. Причому Augmented Reality буде інтегрована настільки глибоко, що стане сприйматися інтуїтивно. І сам термін "Доповнена реальність", можливо, перестане використовуватися, а цифрові дані будуть вважатися частиною самої реальності. Головне, що доповнена реальність дозволяє перенести нас на новий рівень взаємодії з цифровим світом без необхідності відмовлятися від звичного оточення, і повністю занурюватися у віртуальність.

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

Для досягнення мети були посалені наступні завдання:

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

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

* 1. Доповнена реальність

1.1.1 Передмова

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

Важливо розуміти відмінності між доповненою реальністю і змішаною реальністю. У широкому сенсі доповнена реальність являє собою процес перегляду реального світу і віртуальних об'єктів одночасно, де віртуальна інформація накладається, вирівнюється і інтегрується в фізичному світі. У

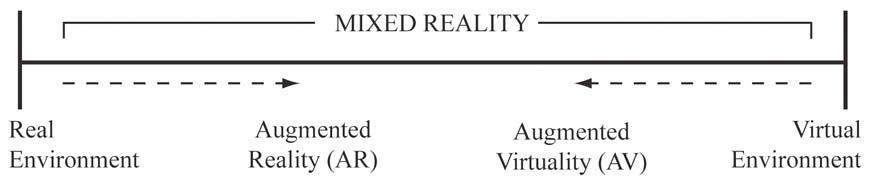
літературі з людино-машинної взаємодії доповнена реальність знаходиться в безперервному діапазоні інтерфейсів від «реальності» до віртуальної реальності «повного занурення»

Рис. 1.1. Діапазон технологій змішаної реальності

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

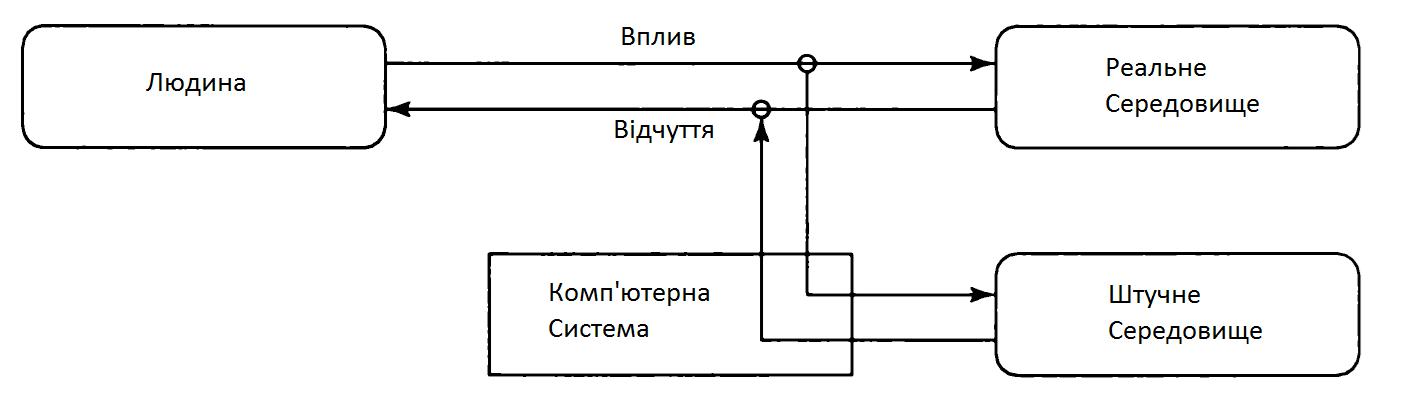
Таким чином, доповнена реальність - це технології, що дозволяють доповнювати зображення реальних об'єктів різними об'єктами комп'ютерної графіки, а також поєднувати зображення, отримані від різних джерел комп'ютерного середовища: відеокамер, акселерометрів, компасів і т.д. Схема середовища доповненої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

Рис 1.2. Схема середовища доповненої реальності

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

1. комбінування реального та віртуального світу;
2. інтерактивність;
3. тривимірне представлення об'єктів.
   * 1. Класифікація систем доповненої реальності

Людина отримує уявлення про навколишній простір за допомогою великого набору органів сприйняття навколишньої інформації. Система доповненої реальності, будучи посередником між людиною і реальністю,

повинна створювати сигнал для одного з таких органів. Таким чином, за типом подання інформації системи доповненої реальності бувають:

*Візуальні*.В їх основі лежить зорове сприйняття людини.Завдання такихсистем - 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.Найчастіше такісистеми 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 .

*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 системпозиціонування 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,якапереміщається 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 такихсистем зводиться до надання інформації про об'єкти. Наприклад, подібні системи можуть аналізувати об'єкти, що знаходяться в полі зору людини і видавати довідкову інформацію про них. Також системи такого типу використовуються в медицині. Наприклад, система Gait Aid 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 різнідії 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.

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

За ступенем мобільності системи доповненої реальності можна класифікувати як:

*Стаціонарні*.Системи цього типу призначені для роботи в фіксованомумісці; переміщення таких систем означає часткове або повне припинення їх працездатності.

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

* + 1. AR прилади

Основними пристроями, використовуваним в системах доповненої реальності є дисплеї, пристрої введення, пристрої відстеження та комп'ютер.

*Дисплеї.* Існує три основних типи дисплеїв, використовуваних в роботі з доповненою реальністю: head mounted displays (HMD), ручні дисплеї і просторові.

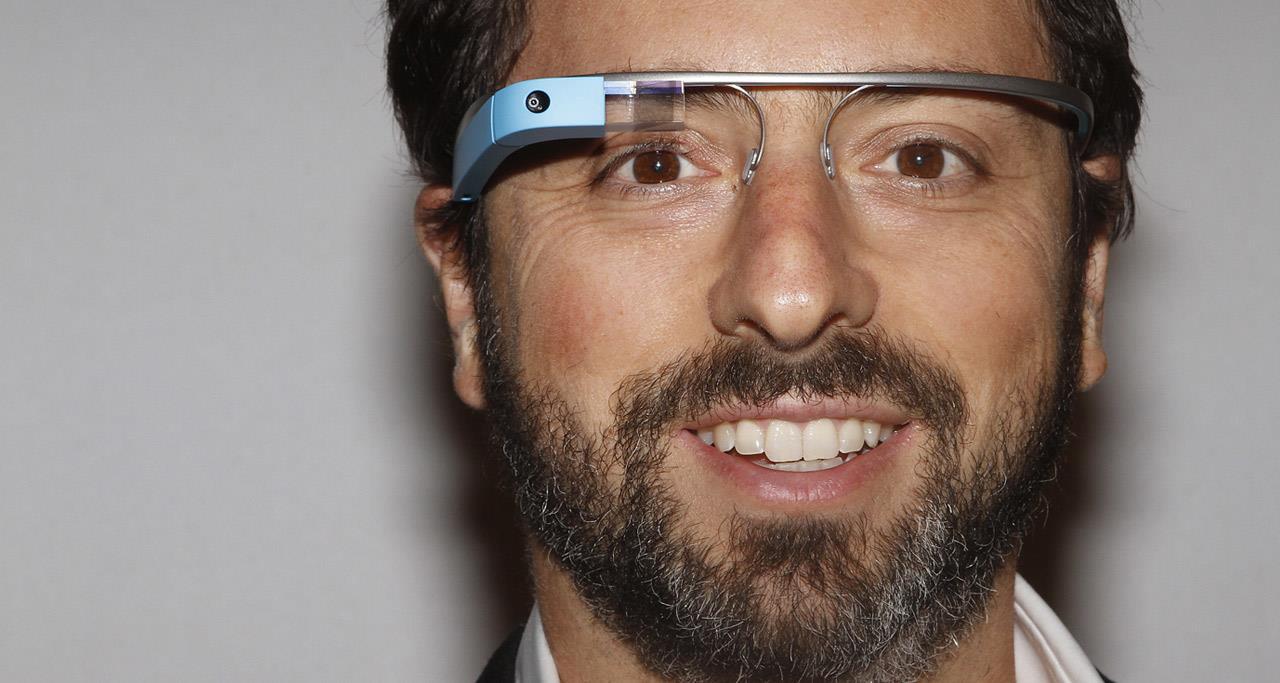
HMD є пристроєм, який закріплюється на голові користувача, або на спеціальному шоломі і розміщає зображення реального і віртуальною середовища перед очима користувача. HMD може бути або відео-прозорим або оптико-прозорим (Рис. 1.3).

Рис 1.3. Google Glass приклад оптико-прозорого HMD

Відео-прозорі системи більш вимогливі, ніж оптико-прозорі так, як вони потребують того, щоб користувач носив дві камери на голові, а отже обробки даних з обох камер, щоб відображати як "реальну частину" доповненої сцени, так і віртуальні об'єкти. У цей час оптико-прозорі системи використовують технологію половини срібного дзеркала, що дозволяє дивитися на реальний фізичний світ через спеціальну «лінзу», яка накладає додаткову графічну інформацію. Сцена, а також реальний світ в такій системі сприймаються більш природно. З іншого боку, у відео-прозорих системах доповнена реальність вже спочатку поєднана з реальною, що дає набагато більше контролю над результатом. Таким чином, контроль над часом кінцевої сцени може бути досягнутий шляхом синхронізації віртуального зображення до його відображення. В оптично-прозорому варіанті, відображення реального світу не може бути загальмовано, однак це призводить до затримки при введенні в систему, графіки і обробці зображень.

Під ручними дисплеями розуміються невеликі обчислювальні пристрої з дисплеєм, які користувач може тримати в руках. Вони використовують відео- прозорі методи для накладення графіки на реальне середовище, а так само використовують додаткові датчики, такі як компас, GPS і акселерометр. Для реалізації систем доповненої реальності часто використовуються системи розпізнавання маркерів, такі як ARToolKit, або методи комп'ютерного зору, такі як SLAM. Такими пристроями можуть бути: смартфони, КПК і планшетні ПК.

Смартфони портативні і широко поширені пристрої, що поєднують потужний процесор, камеру, набір необхідних сенсорів, що робить їх дуже перспективною платформою для AR систем. Планшетні ПК набагато потужніші, ніж смартфони, але вони значно дорожчі і занадто важкі для тривалого використання. Однак з розвитком технологій планшети стають все більш перспективною платформою для роботи з доповненою реальністю.

*Пристрої вводу*. Є багато типів пристроїв введення для AR систем. Деякі системи використовують рукавички. Інші, такі як ReachMedia використовують бездротові браслети. У разі смартфонів, телефон сам по собі може бути використаний як вказівний пристрій, наприклад, у додатку Google Sky Map на Android телефон вимагає від користувача направити камеру в напрямку зірки або планети, про яку він хоче дізнатися. Вибір пристроїв введення в значній мірі залежить від типу розроблюваної системи і типу обраного дисплея. Наприклад, якщо додаток вимагає, щоб руки користувача були вільними, пристрої введення будуть обрані відповідно. Аналогічним чином, якщо система використовує портативний дисплей, розробники можуть використовувати як пристрої введення сенсорний екран.

*Пристрої відстеження*. Пристроями стеження є цифрові камери або інші оптичні датчики,GPS, акселерометри, компаси, бездротові датчики і т.д. Кожна з цих технологій має різну ступінь точності і багато в чому залежить від типу розроблюваної системи.

1.1.4 Мобільні системи доповненої реальності

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

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

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



Рис. 1.4. AR на мобільному пристрої

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

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

Деякі системи покладаються тільки на комп'ютерний зір, але більшість з них розраховані на роботу в приміщеннях, де навколишнє середовище легко контролюється. Коли справа доходить до візуального відстеження на вулиці, з'являються зовнішні фактори, які значно ускладнюють завдання. Одіна 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.

1.1.5 Огляд існуючих технологій для роботи з AR

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

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

Vuforia забезпечує API для C++, Java, Objective-C і .Net мов. Є розширення ігрового движка Unity. Таким чином, SDK підтримує як рідні для IOS і Android мови, так і одночасно дозволяє розробляти додатки доповненої реальності в Unity, які можна легко портувати на обидві платформи. Саме тому, додатки, розроблені з використанням Vuforia, сумісні з широким спектром мобільних пристроїв, включаючи iPhone, iPad, Android - телефонів і планшетів під управлінням ОС Android версії 2.2 або вище і ARMv6 або 7 процесор з FPU.

ARToolkit - це бібліотека комп'ютерного стеження для створення додатків з доповненою реальністю Для цього він використовує можливості відео спостереження, розрахунок реального стану та орієнтації камери по відношенню до квадратного фізичного маркеру в режимі реального часу. Коли реальний стан камери відомо, віртуальна камера може бути розташована в тій же точці і 3D модель накладається на реальний маркер. Так ARToolKit вирішує дві ключові проблеми в доповненої реальності: відстеження погляду і віртуальної взаємодії об'єктів.

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

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

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

Таблиця 1.1. Порівняння технологій AR

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

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

1.2 Мобільні пристрої

1.2.1 Передмова

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 смартфону 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. Підтримується альянсом Open Handset Alliance (OHA). Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java. Android дозволяє створювати Java-додатки, що керують пристроєм через розроблені Google бібліотеки. Android Native Development Kit дозволяє перенести бібліотеки і компоненти додатків, написані на С та інших мовах.

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

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

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

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

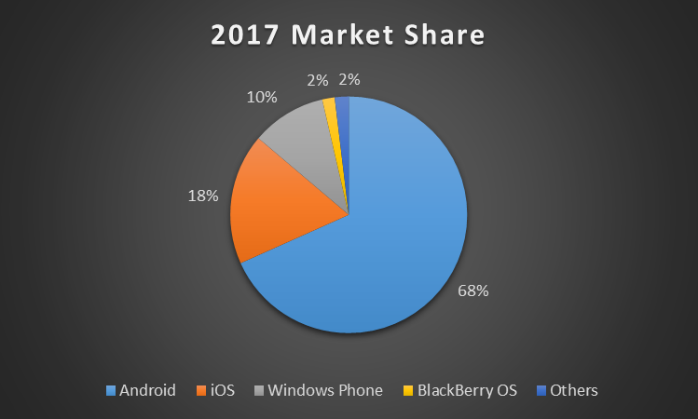


Рис. 1.6. Розподіл мобільних ОС станом на початок 2017

Огляд існуючих мобільних пристроїв та оперційних 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 найпопулярнішо операційною системою для мобільних пристроїв, її частка становить близько 70%, це означає, що додатки написані під Android матимуть більший попит, ніж на додатки, для інших операціних систем. Також для розробки додатків під Android є добре задокуметноване Android SDK і безкоштовне середовище розробки - Android Studio.

1.3 Програми-аналоги

Оскільки ціловою платформою було обрано операціну 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. Розглянемо найполярніші 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. Можна вибирати предмети, змінювати їх розмір і розташування. 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 є захоплюючим обєднанням смартфона, віртуальної і доповненої реальності. Призначений для професійних архітекторів, який дозволяє інтегрувати свої креслення в додаток, які потім будуть перетворені в SmartReality-сумісні файли. Після того, як файл буде завантажений і проаналізований , користувачі можуть використовувати свої ж креслення в якості маркерів і перевести їх в будь-який віртуальної реальності або доповненої реальності.

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

В AR режимі SmartReality створює 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.



Рис.1.7 Інтерфейс додатку SmartReality

*Augment*. Додаток Augment використовує маркерну технологію, серед плюсів якої є можливість самостійно управляти контентом, а простіше кажучи, вертіти маркер разом з моделлю, як заманеться. 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 придназначений для якоїсь конкретної задачі, від 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 дозавантажинти.

Порівняльна характеристика програм-аналогів наведена в таблиці 1.2.

Таблиця 1.2. Порівняльна характеристика програм-аналогів

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

Прооаналізувавши 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 професіний інструмент для архітекторів, за допомогою якого можна реалізувати власний задум просто на папер і виявити власні помилки. Augment - це додаток який звичайно можна використовувати як розвагу, але водночас він може слугувати і помічником в вирішенні побутових проблем.

РОЗДІЛ 2. Опис засобів розробки та роботи ДОДАТКУ

2.1 Постановка задачі дипломного проектування

Завданням на дипломне проектування є розробка мобільного додатку для візуалізаціїї 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, маштабувати. При розробці додатку потрібно ознайомитись з існуючими методами роботи з технологією доповненої реальності, роботою з камерою на ОС Android, роботою з багатопоточністю в Java.

2.2 Інструментальні засоби для реалізації

2.2.1 ARToolkit

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

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

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

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

Особливості:

1. Надійне відстеження, включаючи відстеження природних функцій
2. Сильна підтримка калібрування камери
3. Одночасне відстеження та підтримка стереозвуку
4. Підтримка декількох мов
5. Оптимізовано для мобільних пристроїв
6. Підтримка Full Unity3D і OpenSceneGraph
7. Одна з найскладніших частин розробки програми доповненої реальності - це точно розрахувати точку зору користувача в реальному часі, щоб віртуальні зображення були точно вирівняні з об'єктами реального світу. ARToolKit використовує методи комп'ютерного зору для розрахунку реального стану і орієнтації камери щодо квадратних фігур або плоских текстурованих поверхонь, що дозволяє програмісту накладати віртуальні об'єкти. ARToolKit в даний час підтримує класичний квадратний маркер, 2D-штрих-код, мультімаркер і природне відстеження функцій. Крім того, ARToolKit підтримує будь-яку комбінацію вищевказаного разом. Швидке і точне відстеження, що надається ARToolKit, дозволило швидко розвинути тисячі нових і цікавих програм AR.
8. Документація містить повний опис бібліотеки ARToolKit, її установку і використання її функцій в додатках AR. Кілька прості приклади додатки забезпечені ARToolKit, щоб програміст міг відразу почати роботу. ARToolKit включає бібліотеки відстеження і повні вихідні тексти для цих бібліотек, дозволяючи програмувати перенесення коду на різні платформи або налаштовувати його для власних додатків.
9. ARToolKit є набір інструментів, написаний, які працюють в операційних системах Windows, Mac OS X, Linux, IOS і Android. Функціональність кожної версії інструментарію однакова, але продуктивність може варіюватися в залежності від різних конфігурацій обладнання. ARToolKit можна легко перенести на інші нові і експериментальні платформи.
10. ARToolKit підтримує як відео, так і оптичний прозорий доповнена реальність. Відео-перегляд AR - це те, де віртуальні зображення накладаються на живе відео реального світу. Альтернативою є оптична прозора доповнена реальність, де комп'ютерна графіка приходиться безпосередньо на погляд реального світу. Оптична прозора реальна реальність зазвичай вимагає прозорого дисплея, встановленого на голівці, і має більш складні вимоги до калібрування і реєстрації камери.



Рис 2.1. Логотип ARToolkit

2.2.2 Android

Android - операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (OHA).

Хоча Android базується на ядрі Linux, він стоїть осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є віртуальна машина Dalvik (починаючи з версії 5.0 середовище виконання Android Runtime повністю замінило Dalvik). Додатки на цю платформу являють собою прогрмами з нестандартним байт-кодом. Для них був розроблений спеціальний формат установочних пакетів .APK.

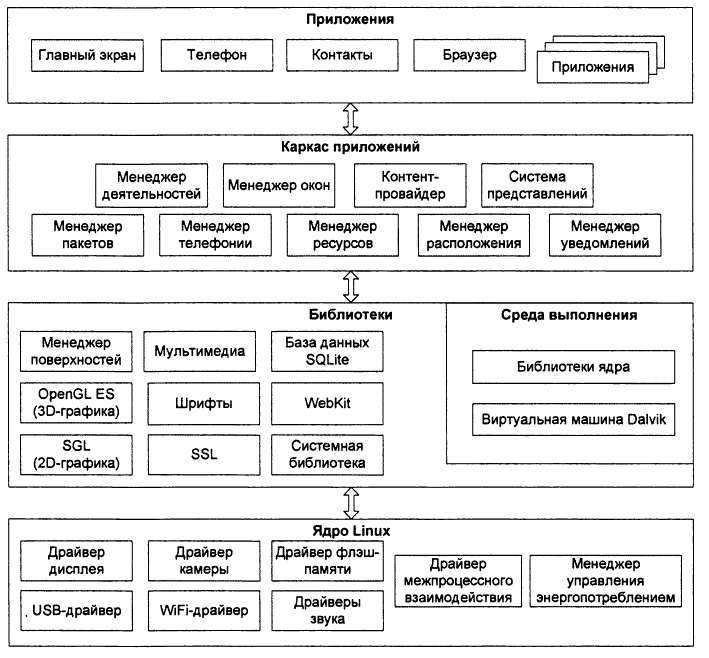


Рис. 2.2. Архітектура платформи Android

У 84 % смартфонів, проданих у 3-ому кварталі 2014 року, була встановлена операційна система Android[4].

У березні 2017 року ОС Android стала найпопулярнішою ОС, з якої виходили в інтернет.[5]. Так з 37,93% користувачів заходили в інтернет із Android'а, а з Windows лише 37,91% користувачів. В Азії показники ще вищі — 52,2% і 29,2% відповідно.

У липні 2005 року компанія Google купила Android, Inc.[6]. Усі засновники цієї стартап-компанії пішли працювати у Google. На той час, мало що було відомо про Android, Inc. окрім того, що вони займаються розробкою ПЗ для мобільних телефонів. Такий розвиток подій спричинив виникнення чуток про те, що Google планує увійти на ринок мобільних телефонів, але було незрозуміло, що саме компанія планує там робити.

У Google група на чолі з Рубіном розробила ОС на основі Linux (ядро v2.6), яку вони пропонували розробникам телефонів та операторам мобільного зв'язку як гнучку та розширювану систему. Повідомлялося, що Google планує співпрацю з рядом розробників апаратних компонентів та програмного забезпечення, і є відкритою для співпраці з операторами мобільного зв'язку.

У грудні 2006 року знову пішли чутки про те, що Google буде просуватися на ринок мобільних телефонів. Доповіді BBC і The Wall Street Journal зазначали, що Google хоче розмістити пошуковик Google і ПЗ Google на мобільних телефонах, і компанія постійно напружено працює для досягнення цієї мети.

Далі у пресі та онлайн-ЗМІ почали з'являтися чутки, що Google розробляє телефон під власним брендом. За цими були інші, які стверджували, що Google визначила технічні характеристики та вже презентує прототипи розробникам телефонів та операторам мобільного зв'язку. Повідомлялося, що буде реалізовано приблизно 30 прототипів. Network World повідомляє, що Google-телефон дійсно є телефоном з відкритою операційною системою, на відміну від схожих продуктів, таких як iPhone. Проектом створення смартфона з використанням відкритого коду, в тому числі з використанням ядра Linux.

У 2009 році було представлено цілих чотири оновлення платформи. Так, в лютому вийшла версія 1.1 з виправленням різних помилок. У квітні та вересні вийшли ще два оновлення - 1.5 «Cupcake» і 1.6 «Donut» відповідно. Оновлення «Cupcake» привнесло істотні зміни: віртуальна клавіатура, відтворення і запис відео, браузер і інші. У «Donut» вперше з'явилися підтримка різних дозволів і щільності екрану і мереж CDMA. У жовтні того ж року вийшла версія операційної системи Android 2.0 «Eclair» з підтримкою декількох акаунтів Google, підтримкою браузером мови HTML5 та інших нововведень, а також після невеликого оновлення в межах версії «Eclair» (2.1) з'явилися «живі шпалери» і був видозмінений екран блокування.

У середині 2010 Google представила Android версії 2.2 під найменуванням «Froyo», а в кінці 2010 року - Android 2.3 «Gingerbread». Після оновлення «Froyo» стало можливо використовувати смартфон в якості точки доступу, використовувати традиційну систему блокування смартфона цифровим або буквено-цифровим паролем та інші зміни, а оновлення «Gingerbread» привнесло більш повний контроль над функцією копіювання і вставки, поліпшення управління живленням і контролю над додатками, підтримку декількох камер на пристрої і т. д. 22 лютого 2011 року була офіційно представлена орієнтована на інтернет-планшети платформа Android 3.0 «Honeycomb».

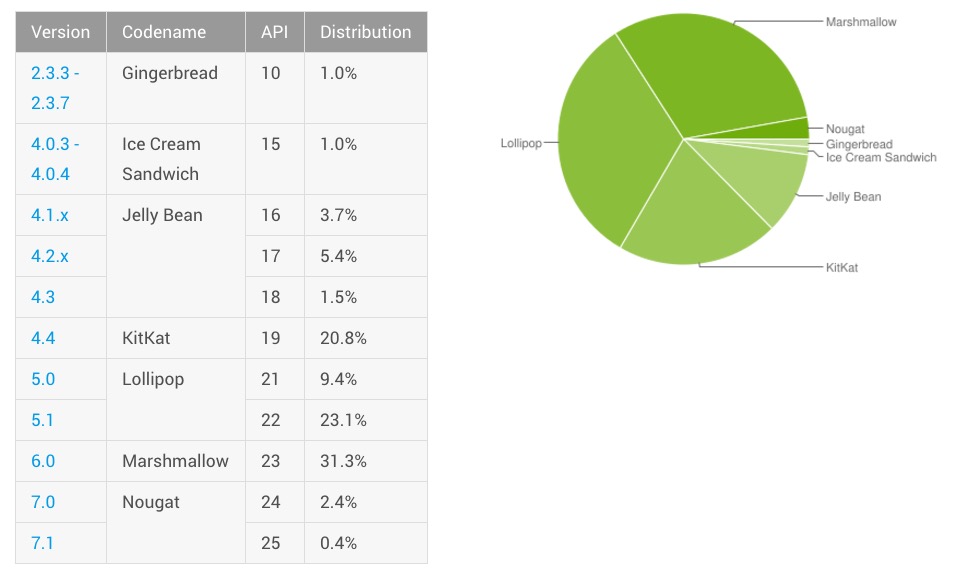


Рис. 2.3. Розподіл версій Android OS станом на 2017р.

Android 4.0 «Ice Cream Sandwich», що вийшла 19 жовтня 2011, - перша універсальна платформа, яка призначена як для планшетів, так і для смартфонів. Також оновлення привнесли новий інтерфейс «Holo», який використовувався до Android 4.4.4 Kitkat, на даний момент замінений на Material Design. У червні 2012 вийшло оновлення під назвою «Jelly Bean» з порядковим номером 4.1, який змінився на 4.2 внаслідок невеликого оновлення в кінці жовтня того ж року і на 4.3 після оновлення в липні 2013. 31 жовтня 2013 Google представила наступну версію операційної системи Android 4.4, яка отримала назву шоколадного батончика «KitKat» за угодою з компанією виробником Nestlé . Вперше KitKat з'явився на Nexus 5; ця версія Android оптимізована для роботи на більш широкому наборі пристроїв, що мають 512 МБ ОЗУ як рекомендований мінімум. 25 червня 2014 Google представили Android L, зараз доступний для розробників.

Деякі користувачі відзначають, що Android проявляє себе краще одного зі своїх конкурентів, Apple iOS, в ряді особливостей, таких як веб-серфінг, інтеграція з сервісами Google і інших[43]. Також Android, на відміну від iOS, є відкритою платформою, що дозволяє реалізувати на ній більше функцій.

Незважаючи на початкову заборону на установку програм з «неперевірених джерел» (наприклад, з карти пам'яті), це обмеження відключається штатними засобами в налаштуваннях пристрою, що дозволяє встановлювати програми на телефони та планшети без інтернет-підключення (наприклад, користувачам, які не мають Wi-Fi-точки доступу і не бажають витрачати гроші на мобільний інтернет, який зазвичай коштує дорого), а також дозволяє будь-кому безкоштовно писати програми для Android і тестувати на своєму пристрої.

Android доступний для різних апаратних платформ, таких як ARM, MIPS, x86.

Існують альтернативні Google Play магазини додатків: Amazon Appstore, Opera Mobile Store, GetUpps!, F-Droid.

У версії 4.3 з'явилась підтримка багатокористувацького режиму.

Також Android був підданий критиці:

Платформа базується на Java (спеціальна реалізація Dalvik), тому переваги і можливості операційної системи Linux на цій платформі практично не використовуються. Наприклад, не використовується жоден з популярних графічних тулкітів і бібліотек (наприклад Qt або GTK), що робить малоймовірною появу значної кількості застосунків, портованих з повноцінного десктопного варіанту Linux на цю платформу через відсутність поза вибором X-сервера і поширених графічних бібліотек.

З'явилася інформація про те, що Google на свій розсуд видалятиме застосунки на телефонах користувачів, якщо порушуються умови їх використання.

Конкуренти Android виступили з критикою платформи, звинувачуючи її в надмірній фрагментації, що створює перешкоди розробникам[46][47]. Google спростувала всі звинувачення, заявивши, що ніяких подібних проблем немає[48].

Щоб користувачі телефонів мали доступ до Google Play та інших сервісів від Google, виробники цих телефонів мають укласти контракт з Google на використання відповідного пропрієтарного програмного забезпечення.

2.2.3 Java

**Java** — це об'єктно-орієнтована мова програмування. Синтаксис мови багато в чому походить від C та C++. 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 ліцезією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель С++, 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 С/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій java дозволяє викликати підпрограми, написані іншими мовами програмування.

Мова Java активно використовується для створення мобільних додатків під операційну систему Android. При цьому програми компілюється в нестандартних байтах-код, для використання їх віртуальна машина Dalvik (з Android починаючи 5,0 Lollipop віртуальної машини замінена на АРТ). 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 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 NDK

Android NDK (Android Native Development Kit)  — необхідний набір інструментарію для розробки компонентів програмного забезпечення для платформи [Android](https://uk.wikipedia.org/wiki/Android), який базується на [C](https://uk.wikipedia.org/wiki/C)/[C++](https://uk.wikipedia.org/wiki/C%2B%2B) та інших мовах програмування. Містить в собі лімітований набір загальновживаних низькорівневих (нативних) бібліотек та [API](https://uk.wikipedia.org/wiki/API), написаних на С/С++ та інших мовах програмування, документацію і мінімальний набір прикладів для демонстрації базового функціоналу. За допомогою NDK розробник застосунку для операційної системи Android може імплементувати окремі його частини, використовуючи такі мови, як C/C++, а не тільки [Java](https://uk.wikipedia.org/wiki/Java). Це надає можливість використати деякі переваги, оскільки в окремих випадках код, написаний на C/C++, виконуватиметься швидше в порівнянні з кодом на Java. Android NDK може бути використаний для платформи Android 1.5 (API Level 3) і новіших версій.

Для зручності [компіляції](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%96%D0%BB%D1%8F%D1%86%D1%96%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) коду Android NDK містить спеціальний скрипт ndk-build, завданням якого є наступне:

* автоматичне визначення правил компіляції і файлу додатку для цільових [бінарних файлів](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%BD%D0%B0%D1%80%D0%BD%D0%B8%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB)
* генерація бінарних файлів
* копіювання згенерованих бінарних файлів в потрібну директорію, визначену користувачем.

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

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

2.2.6 Java Native Interface

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

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

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

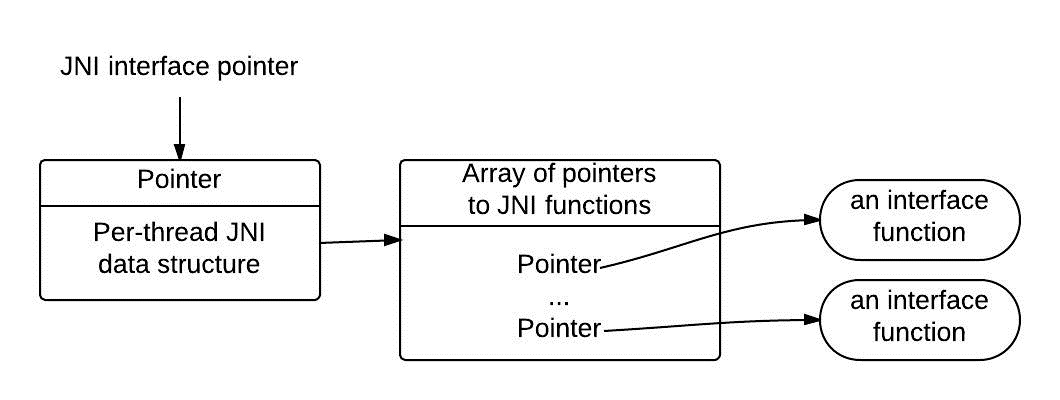
JNI таблиця, організована як таблиця віртуальних функцій в С ++. VM може працювати з декількома такими таблицями. Наприклад, одна буде для налагодження, друга для використання. Покажчик на JNI інтерфейс дійсний тільки в поточному потоці. Це означає, що покажчик не може гуляти з одного потоку в інший.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

Рис. 2.4 Структура JNI

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

2.2.7 Android Studio

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

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

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

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

Для прискорення розробки застосунків представлена ​​колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, 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](https://uk.wikipedia.org/wiki/GitHub).

До складу також включені пристосовані під особливості платформи Android розширені інструменти [рефакторингу](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3), перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.

2.2.7 Open GL

OpenGL (англ. Open Graphics Library — відкрита графічна бібліотека) — специфікація, що визначає незалежний від мови програмування крос-платформовий програмний інтерфейс (API) для написання застосунків, що використовують 2D та 3D комп'ютерну графіку. 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 (Compiz, Clutter), при візуалізації наукових даних, в системах автоматизованого проектування тощо.

OpenGL орієнтується на такі два завдання: cховати 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 картинки на екрані і/або в пам'яті. Векторні трансформації та растеризация виконуються графічним конвеєром (graphics pipeline), який власне являє собою дискретний автомат. Абсолютна більшість команд OpenGL потрапляють в одну з двох груп: або вони додають графічні примітиви на вхід в конвеєр, або конфігурують конвеєр на різне виконання трансформацій. OpenGL є низькорівневим процедурним API, що змушує програміста диктувати точну послідовність кроків, щоб побудувати результуючу растрову графіку (імперативний підхід). Це є основною відмінністю від дескрипторних підходів, коли вся сцена передається у вигляді структури даних (найчастіше дерева), яке обробляється і будується на екрані. З одного боку, імперативний підхід вимагає від програміста глибокого знання законів тривимірної графіки та математичних моделей, з іншого боку — дає свободу впровадження різних інновацій.

2.2.8 Формат .obj і .mtl

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

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

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

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

Інформація про зовнішній вигляд об'єктів (матеріали) передаються в файлах-супутники 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.

* 1. Вхідні та вихідні дані

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

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

3. РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1 Підготовка проекта

3.1.1 Інcталяція 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 прогрму інсталяції. Далі вимагається вказати два шляхи. Перший шлях буде використаний для установки Android Studio. Другий - для установки Android SDK (рис. 3.1).

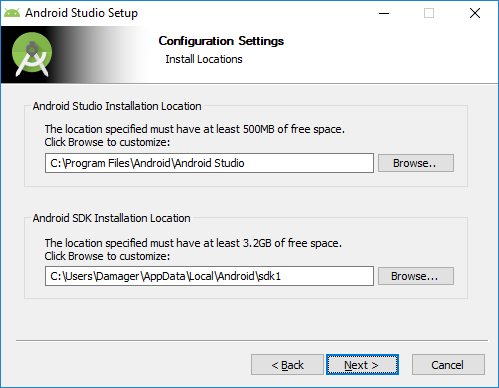


Рис. 3.1

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

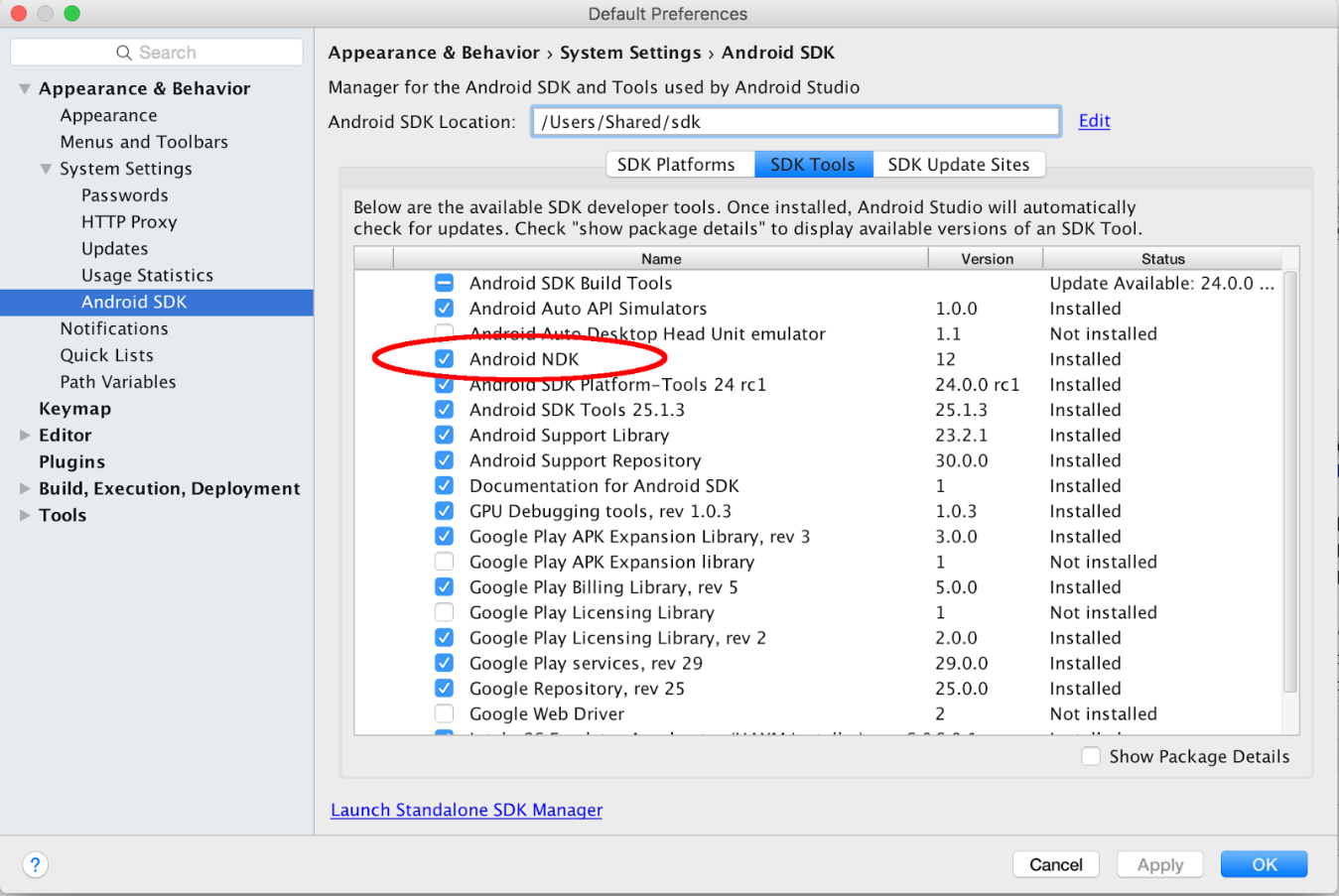
Для написання нативного коду С/С++ в Android необхідно завантажити Android NDK. Для його завантаження необхідно відкрити Android SDK Manager і відмітити пункт “Android NDK” (рис. 3.2).

Рис. 3.2

На цьому налаштування середовища розробки можна вважати завершеним.

3.1.2 Створення проекту та інтеграція JNI

Для створення проекту необхідно запустити Android Studio. У відкритому вікні вибрати «Start a new Android Studio project» (рис 3.3).

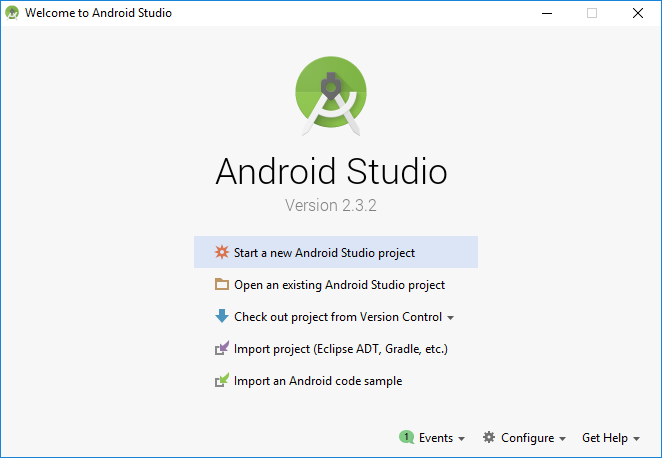
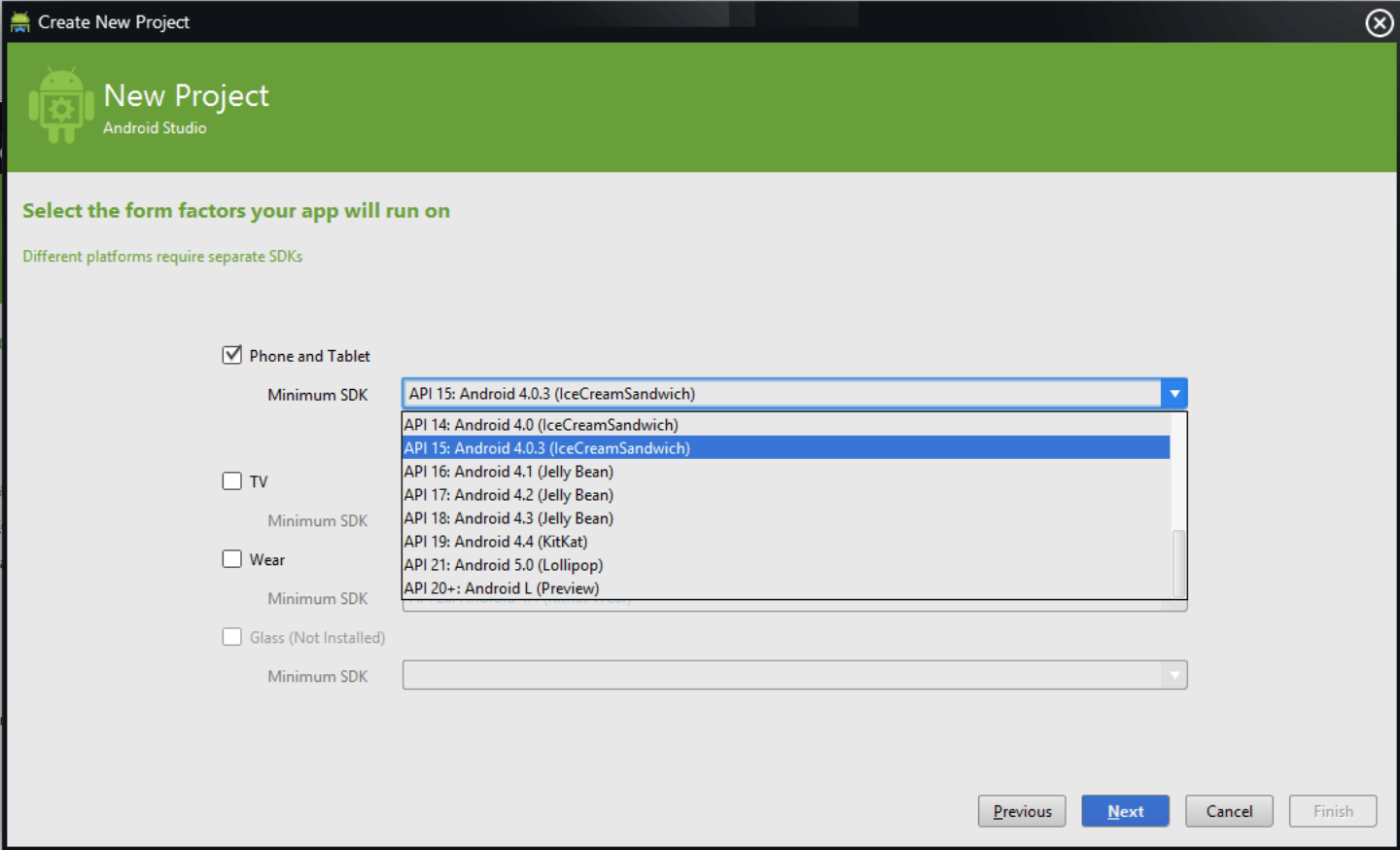


Рис. 3.3

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 зачить, що додаток буде працювати лише на мобільних теоефонах і планшетах. Ще необхідно вибрати мінімальну версію андроід, для якої буде запускатися додаток. Я виберу «API 15, Android 4.0.3» (рис 3.4), це значить, на версіях Android нижче ніж 4.0.3 додаток працювати не буде. Хоча можна було б обрати і меншу версію, це буде недоцільно з огляду на те, що версіїї Android 4.0.3 і вище охоплюють понад 97% всього ринку Android пристроїв на сьогодні, а ще вагомою проблемою стало б те, що для підтримки старіших версій, довелося б відмовитися від деякої функціональності введеної в нових версіях.

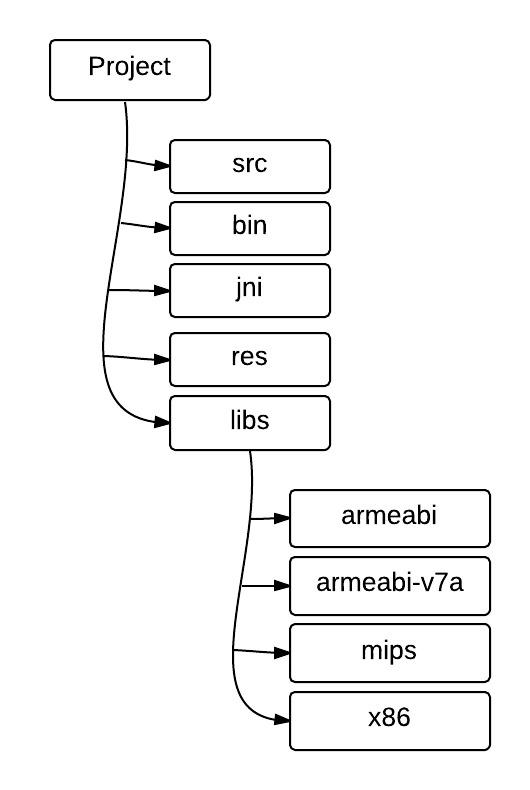
Рис. 3.4

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

Далі необхідно додати підтримку нативного коду необхідно виконати наступні кроки:

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

Загалом структура проекту повинна бути як на рис. 3.5.

Рис. 3.5. Структура проекту

Весь нативний код має знаходиться в папці jni. Після складання проекту, в папці libs створиться чотири папки під кожну архітектуру процесора, в якій буде лежати нативна бібліотека.

3.1.3 Налаштування ARToolkit

Бібліотеку Artoolkit для роботи з доповненою реальністю потрібно завантажити з офіційного сайту <artoolkit.org>. Далі у вкладці «Downloads» необхідно вибрати «ARToolkit for Android» і дочекатись завантаження. В завантажемому архіві є сама бібліотека, документація по ній, приклади проектів для Android, готові маркерні файли. Для інтеграціїї Artoolkit в додаток необхідно скопіювати файли бібліотек *libARWrapper.so* і *libc++\_shared.so* в кожну папку архітектур процесора.

3.1.3 Налаштуванння маніфест файлу

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

<uses-permission android:name="android.permission.CAMERA" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

Після цього підготовку проекту можна вважати завершеною.

3.2 Архітектура програмного забезпечення

3.2.1 MVC

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

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

Моде́ль–вигляд–контро́лер (або Модель–представлення–контролер, англ. Model-view-controller, MVC) — архітектурний шаблон, 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 3.6). 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 (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле) 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, наприклад гістограма для керівництва компанії й таблиці для бухгалтерії.

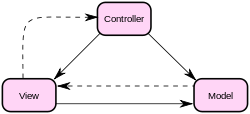
Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Рис. 3.6. MVC

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

3.2.2 Модель

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

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

ObjectsHelper – клас для зберігання і керування списком 3D моделей. Реалізує шаблон програмування Singleton для забезпечення тільки одного екземпляра даного класу в системі.

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

Native - клас, що визначає інтерфайс функцій при проботі з нативним кодом. Особливістю є те, що в даному класі відбувається статичне підключення С/С++ бібліотек, а також всі функції оголошені із зарезервованим словом словом «native», що вказує на точку входу в JNI. Також слід відзначити, всі функції даного класу мають модифікатор доступу «public» і ключове слово «static».

NativeRenderer - клас, реалізує функції для рендеринга тривимірної графіки в OpenGL.

3.2.3 Контролер

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

До категорії контролера відносяться наступні Java класи:

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

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

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

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

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

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

3.2.4 Представлення

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

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

fragment\_controll\_model.xml - визначає представлення класу контролера ControllModelDialog. Розміщені view-компоненти, які реалізують основні функції трансформаціїї моделей.

fragment\_models\_list.xml - визначає представлення класу контролера ModelsListDialog. Розміщений компонент RecyclerView, який дозволяє відображати список елементів.

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

model\_item.xml – визначає вид компонентів списку, який відображається в ModelsListDialog.

screenshot\_dialog\_menu.xml – файл ресурса типу menu, який реалізує відображення меню на Android ToolBar.

3.2.5 С/С++ функціонал

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

Загалом функцію JNI в С/С++ коді можна пресдставити в наступному вигляді:

jtype Java\_package\_class\_method(JNIEnv \*env, jobject obj, args) {

//Тіло метода

**return** jtype;

}

Написання JNI функцій має свої особливості, яких необхідно дотримуватися.

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

Функція приймає два обов’язкових параметра: \*env – вказівник на JNI інтерфейс; оbj – посилання на об’єкт в якому описаний нативний метод. Інші args параматри є параметрами, які передаються з java коду, вони можуть бути відсутні. Типи параметрів визначає JNI інтерфейс, який має власні типи даних, що мають аналогії в Java. Примітивні типи даних JNI розміщені в таблиці 3.3.

Таблиця 3.3. Примітивні типи даних JNI

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

Native.cpp має наступний інтерфейс:

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

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

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

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

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

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

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

3.3 Інтеграція ресурсів в додаток

Файли ресурсів в Android проекті можуть міститися в двох директоріях: res і assets(активи). Обидва дуже схожі. Головна відмінність між ними полягає в тому, що в каталозі res коженому файлу призначається попередньо скомпільований ID, який 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 ви хотіли б там. Ви можете отримати доступ до кожного з файлів в цій системі, як при доступі до будь-якого файлу в будь-який файлової системи за допомогою Java.

3.3.1 Завантаження 3D моделей

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

3.3.2 Генерація ARToolkit маркера

3.1 Геометричні перетворення в тривимірному просторі

У лінійній алгебрі, базис векторного простору розмірності n - це послідовність з n векторів (α1, α2,…. αn) 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 β1, β2… βn виражаються через вектори α1, α2… αn як:

β1 = a11 α1+ a12α2+…+ a1nαn

β2 = a21 α1+ a22α2+…+ a2nαn

...

βn = an1 α1+ an2α2+…+ annαn

то матрицею переходу від базиса (β1, β2… βn)до базиса(α1, α2… αn) буде:

При множенні матриці, оберненої до матриці переходу, на стовпець, складений з коефіцієнтів розкладання вектора по базису (α1, α2… αn) ми отримуємо той же вектор, виражений через базис (β1, β2… βn).

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

Таблиця 3.1

|  |  |
| --- | --- |
| Масштабування  де a, b і с - коефіцієнти масштабування відповідно по осях OX, OY і OZ: |  |
| Поворот  де φ - кут повороту зображення в двовимірному просторі | Відносно ОХ на кут φ  Відносно ОY на кут  Відносно ОZ на кут |
| Переміщення  де a, b і с - зміщення відповідно по осях OX, OY і OZ. |  |

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