

# ЛЕКЦІЯ 10. ВИКОРИСТАННЯ МОЖЛИВОСТЕЙ СМАРТФОНА У ДОДАТКАХ

## 10.1 ВВЕДЕННЯ

Особливістю більшості мобільних пристроїв є наявність сенсорного екрану і можливість управління пальцем (touch-interface), очевидно, що це необхідно враховувати і використовувати при розробці додатків. Смартфон, якщо вже з'являється у людини, стає його супутником завжди і скрізь, у зв'язку з цим, досить часто використовується, як фотоапарат або програвач музики, а також смартфони все частіше стають інструментами орієнтування на місцевості.

У даній темі передбачається розгляд питань розробки додатків, орієнтованих на тач-інтерфейс, роботу зі звуком, використання камери і глобальних систем позиціонування.

## 10.2 ВІДМІННІ РИСИ СМАРТФОНІВ

Прийшов час поговорити про найцікавіші можливості смартфонів, які можна використовувати в додатках. Ні для кого не секрет що смартфон є "розумним телефоном": передбачає обов'язкову наявність операційної системи і можливість установки додаткових додатків, істотно розширюють функціонал пристрою. З одного боку, смартфон виконує всі звичні функції мобільного телефону і, завдяки компактним розмірам, завжди під рукою. З іншого боку, завдяки наявності процесора і операційної системи, дозволяє виконувати багато функцій повноцінного комп'ютера. Додатково до всього, смартфони мають ряд цікавих особливостей, не характерних для телефонів і комп'ютерів.

Для початку звернемо увагу на екран смартфона. У сучасних смартфонах екран займає практично всю площу передньої панелі пристрою, має високий дозвіл і є чутливим до дотиків. Завдяки такій чутливості, для взаємодії з пристроєм і його додатками можна використовувати віртуальні елементи управління, найчастіше кнопки на екрані. У зв'язку з чим відпадає необхідність у фізичних кнопках. У смартфонах реалізується, так званий, touch-інтерфейс - інтерфейс, заснований на віртуальних елементах управління, вибір яких виконується простим дотиком, а

також на використанні жестів (gestures). Якщо точок дотику кілька (тобто використовується кілька пальців), такий інтерфейс, вже називається multi-touch.

Ще одна особливість смартфонів полягає в тому, що для більшості їхніх власників не останню роль відіграє можливість використання цього "розумного телефону" в якості аудіо або відеоплеєра, тому сучасні пристрої стають все більш і більш мультимедійними. У першій лекції обговорювалося, що до складу платформи Android входить набір бібліотек для обробки мультимедіа Media Framework, в якому реалізована підтримка більшості загальних медіа-форматів. У зв'язку з чим, в додатки, що розробляються для смартфонів під управлінням Android, можна інтегрувати запис і відтворення аудіо та відео, а також роботу з зображеннями.

Важливою і часто використовуваною особливістю смартфонів є наявність камери, яка дозволяє знімати все найцікавіше: від перших кроків дитини до падіння метеорита. Телефон завжди під рукою і готовий до роботи, у зв'язку з цим кількість фотографій і невеликих відеороликів різко збільшилася, і будь-яке цікава подія в житті індивідуума може бути відображене і збережено для нащадків. З ростом можливостей отримання фото та відео матеріалів збільшується потреба в додатках, здатних працювати з цими матеріалами. Платформа Android дозволяє розробляти такі програми, які надають користувачам можливості робити фотознімки або записувати відео, якимось чином обробляти отримані матеріали і використовувати їх далі.

Більшість смартфонів оснащені GPS -модулем, а деякі навіть комбінованим модулем GPS/ГЛОНАСС, що дозволяє використовувати такий пристрій в якості інструменту для орієнтування на місцевості. У багатьох випадках смартфон з встановленим відповідним програмним забезпеченням цілком може замінити GPS навігатор. У розроблюваних додатках іноді буває дуже корисно додати можливість отримання координат пристрою і господаря, якщо обидва знаходяться в одному місці, і використовувати ці координати для будь-яких цілей. Наприклад, вже існують програми, які дозволяють відслідковувати параметри людини (спортсмена) під час подолання деяких відстаней бігом, на велосипеді, на лижах і т.д. Таке додаток працює під час тренування (пристрій повинен переміщатися разом зі спортсменом), по закінченню можна отримати повну статистику маршруту: точний час в дорозі, відстань, підйоми / спуски, середню швидкість, витрачені калорії і т.д.

Зауважимо, що більша частина інформації спирається на дані, отримані із супутників GPS.

Розгляд особливостей смартфонів буде неповним, якщо залишити без уваги датчики і сенсори, якими оснащені більшість пристроїв. Ці мікропристрою забезпечують зв'язок смартфона з навколишнім середовищем і додають нові дивовижні функції. За допомогою датчика наближення, наприклад, можна відключати підсвічування екрану при наближенні телефону до вуха користувача під час розмови, блокувати екран, щоб не було можливості випадково натиснути на відбій. Акселерометр може використовуватися для зміни орієнтації екрану, для управління в іграх, особливо симуляторах, а також в якості шагомера. Датчик освітленості дозволяє регулювати яскравість екрану. Гіроскоп може застосовуватися для визначення більш точного позиціонування пристрою в просторі.

Усі розглянуті особливості в сукупності збільшують привабливість смартфонів, дозволяють розробникам створювати додатки з різноманітними, корисними, цікавими і іноді несподіваними функціями. Далі в лекції розглянемо перераховані можливості смартфонів більш докладно і дізнаємося як можна їх використовувати при розробці додатків.

### 10.3 СЕНСОРНЕ (TOUCH) УПРАВЛІННЯ

У цьому розділі лекції розглянемо можливості додавання сенсорного управління в мобільні додатки під Android. Сенсорне управління уразі використання сенсорних жестів для взаємодії з додатком. У таблиці 10.1 представлений набір жестів, підтримуваний системою Android.

Таблиця 10.1. Система жестів Android

Торкання (touch). Використання: Запуск дії за умовчанням для вибраного елемента. Виконання: натиснути, відпустити.
Довге торкання (long touch). Використання: Вибір елемента. Не варто використовувати цей жест для виклику контекстного меню. Виконання: натиснути, чекати, відпустити.

<p>Ковзання або перетягування (swipe or drag).</p> <p>Використання: Прокрутка вмісту або навігація між елементами інтерфейсу одного рівня ієрархії.</p> <p>Виконання: натиснути, перемістити, відпустити.</p>
<p>Ковзання після довгого торкання (long press drag).</p> <p>Використання: Перегрупування даних або переміщення в контейнер.</p> <p>Виконання: тривале дотик, перемістити, відпустити.</p>
<p>Подвійне торкання (double touch).</p> <p>Використання: Збільшення масштабу, виділення тексту.</p> <p>Виконання: швидка послідовність двох дотиків.</p>
<p>Перетягування з подвійним торканням (double touch drag).</p> <p>Використання: Зміна розмірів: розширення або звуження по відношенню до центру жесту.</p> <p>Виконання: дотик, наступне за подвійним торканням зі зміщенням вгору або вниз при цьому:</p> <ul style="list-style-type: none"> <li>• зсув вгору зменшує розмір вмісту;</li> <li>• зміщення вниз збільшує розмір вмісту.</li> </ul>
<p>Зведення пальців (pinch close).</p> <p>Використання: зменшення вмісту, згортання.</p> <p>Виконання: торкання екрану двома пальцями, звести, відпустити.</p>
<p>Розведення пальців (pinch open).</p> <p>Використання: збільшення вмісту, розгортання.</p> <p>Виконання: торкання екрану двома пальцями, розвести, відпустити.</p>

Про можливість управляти додатком за допомогою сенсорних жестів можна говорити в тому випадку, коли додаток здатний розпізнати, що під набором торкань екрана ховається деякий жест і виконати відповідну дію. Процес розпізнавання жесту зазвичай складається з двох етапів: збір даних та розпізнавання жесту. Розглянемо ці етапи докладніше.

### 10.3.1 ЗБІР ДАНИХ ПРО СЕНСОРНІ ПОДІЇ

Основні дії, які може призвести користувач при взаємодії з сенсорним екраном: торкнутися екрана пальцем, перемістити палець по екрану і відпустити. Ці дії розпізнаються системою Android, як сенсорні події (touch-події).

Кожен раз при появі сенсорного події ініціюється виклик методу `onTouchEvent ()`. Обробка події стане можливою, якщо цей метод реалізований в класі активності або деякого компонента, інакше подія просто ігнорується.

Жест починається, при першому торканні екрану, триває поки система відстежує положення пальців користувача і закінчується отриманням фінального події, що складається в тому, що жоден палець не стосується екрану. Об'єкт `MotionEvent`, переданий в метод `onTouchEvent ()`, надає деталі кожної взаємодії. Розглянемо основні константи класу `MotionEvent`, що визначають сенсорні події:

- `MotionEvent.ACTION_DOWN` - торкання екрану пальцем, є початковою точкою для будь-якого сенсорного події або жесту;
- `MotionEvent.ACTION_MOVE` - переміщення пальця по екрану;
- `MotionEvent.ACTION_UP` - підняття пальця від екрану.

Додаток може використовувати надані дані для розпізнавання жесту.

Можна реалізувати свою власну обробку подій для розпізнавання жесту, таким чином можна працювати з довільними жєстами у додатку. Якщо ж у додатку необхідно використовувати стандартні жєсти, описані в таблиці 10.1, можна скористатися класом `GestureDetector`. Цей клас дозволяє розпізнати стандартні жєсти без обробки окремих сенсорних подій.

### 10.3.2 РОЗПІЗНАВАННЯ ЖЄСТІВ

Android надає клас `GestureDetector` для розпізнавання стандартних жєстів. Деякі жєсти, які він підтримує включають: `onDown ()`, `onLongPress ()`, `onFling ()` і т.д. Можна використовувати клас `GestureDetector` в зв'язці з методом `onTouchEvent ()`.

Починаючи з версії 1.6, Android надає API для роботи з жєстами, який розташовується в пакеті `android.gesture` і дозволяє зберігати, завантажувати, створювати і розпізнавати жєсти. Віртуальний пристрій Android (AVD), починаючи все з тією ж версії 1.6, містить встановлений додаток, яке називається `Gesture Builder` і дозволяє створювати жєсти. Після створення жєсти зберігаються на SD

карті віртуального пристрою і можуть бути додані в додаток у вигляді бінарного ресурсу.

Для розпізнавання жестів необхідно додати компонент `GestureOverlayView`, в XML файл активності. Цей компонент може бути доданий як звичайний елемент графічного інтерфейсу користувача і вбудований в компоновку, наприклад `RelativeLayout`. С іншого боку він може бути використаний, як прозорий шар поверх інших компонентів, в цьому випадку в XML файлі активності він повинен бути записаний, як кореневий елемент.

Крім усього перерахованого вище, для використання власних жестів у додатку необхідно реалізувати інтерфейс `OnGesturePerformedListener` і його метод `onGesturePerformed()`.

## 10.4 РОБОТА З МУЛЬТИМЕДІА

Мультимедіа бібліотека Android включає підтримку відтворення безлічі найбільш поширених форматів, що дозволяє легко використовувати в додатках аудіо, відео та зображення. Можна програвати аудіо або відео з медіа файлів збережених як ресурси програми ( raw ресурси), з файлів, розташованих в файлової системі або з потоку даних, одержуваного через мережеве з'єднання, для всього цього використовується `MediaPlayer API`.

**Зауваження:** програвати аудіофайли можна тільки на стандартному пристрої виводу, неможливо відтворювати аудіо під час дзвінка.

Актуальна інформація про формати аудіо і відео наводиться за посиланням: <http://developer.android.com/guide/appendix/media-formats.html>.

Для відтворення аудіо та відео Android надає клас `MediaPlayer`. Причому при роботі з аудіоконтентом цей клас дозволяє відтворювати необроблені дані, тобто можливо програвання динамічно генерується аудіо.

Діаграма життєвого циклу екземпляра класу `MediaPlayer` представлена на рис. 10.1. Овали представляють стану об'єкта `MediaPlayer`, дуги показують виклики яких методів необхідно виконати, щоб змінити стан об'єкта `MediaPlayer`. Дуги з однією стрілкою представляють виклики синхронних методів, з двома стрілками - виклики асинхронних методів.

У ході життєвого циклу об'єкт `MediaPlayer` проходить через кілька станів:

- **бездіяльність (Idle)** - створений екземпляр класу MediaPlayer для створення може використовуватися оператор new або виклик методу reset () (див. рис. 10.1 );

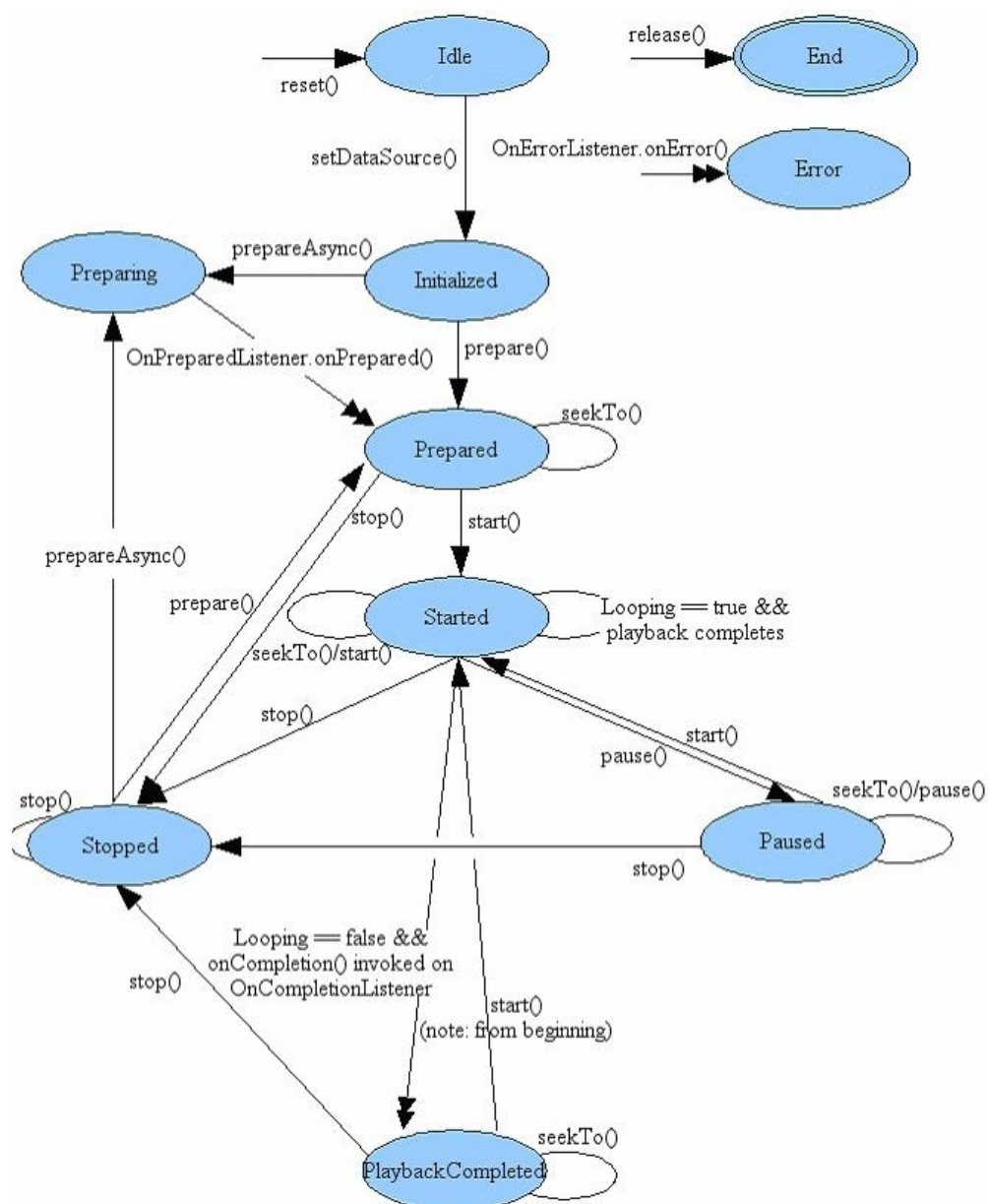


Рис. 10.1. Життєвий цикл екземпляра класу MediaPlayer

- **Ініціалізація (Initialized)** - задане джерело медіа-інформації, для завдання джерела використовується метод `setDataSource ()`;

- **Помилка (Error)** - з'явилася якась помилка, наприклад, не підтримуваний аудіо / відео формат, занадто високий дозвіл, щоб вивести об'єкт із цього стану, необхідно викликати метод `reset ()`;

- **Підготовка (Preparing)** - MediaPlayer займається підготовкою медіаджерел до відтворення, підготовка ініціюється методом `prepareAsync ()`;

- **Готовий (Prepared)** - стан готовності до відтворення, може бути досягнуто двома способами:

- синхронний спосіб: виклик методу `prepare ()`, який переводить об'єкт в готове стан;

- асинхронний спосіб: спрацьовування методу `onPrepared ()` інтерфейса `OnPreparedListener ()` в стані підготовки, як реакція на подію готовності;

- **Запущений (Started)** - виконується відтворення медіа-контенту, в цей стан об'єкт переходить після виклику методу `start ()`;

- **Припинений (Paused)** - відтворення призупинено, `MediaPlayer` переходить в цей стан після виклику методу `pause ()`;

- **Зупинений (Stopped)** - відтворення зупинено, `MediaPlayer` переходить в цей стан після виклику методу `stop ()`;

- **Відтворення завершено (Playback Completed)** - досягнуто кінець відтвореного змісту, в цей стан об'єкт переходить після спрацьовування методу `onCompleted ()` інтерфейсу-слухача `OnCompletionListener`, як реакції на кінець відтвореного матеріалу;

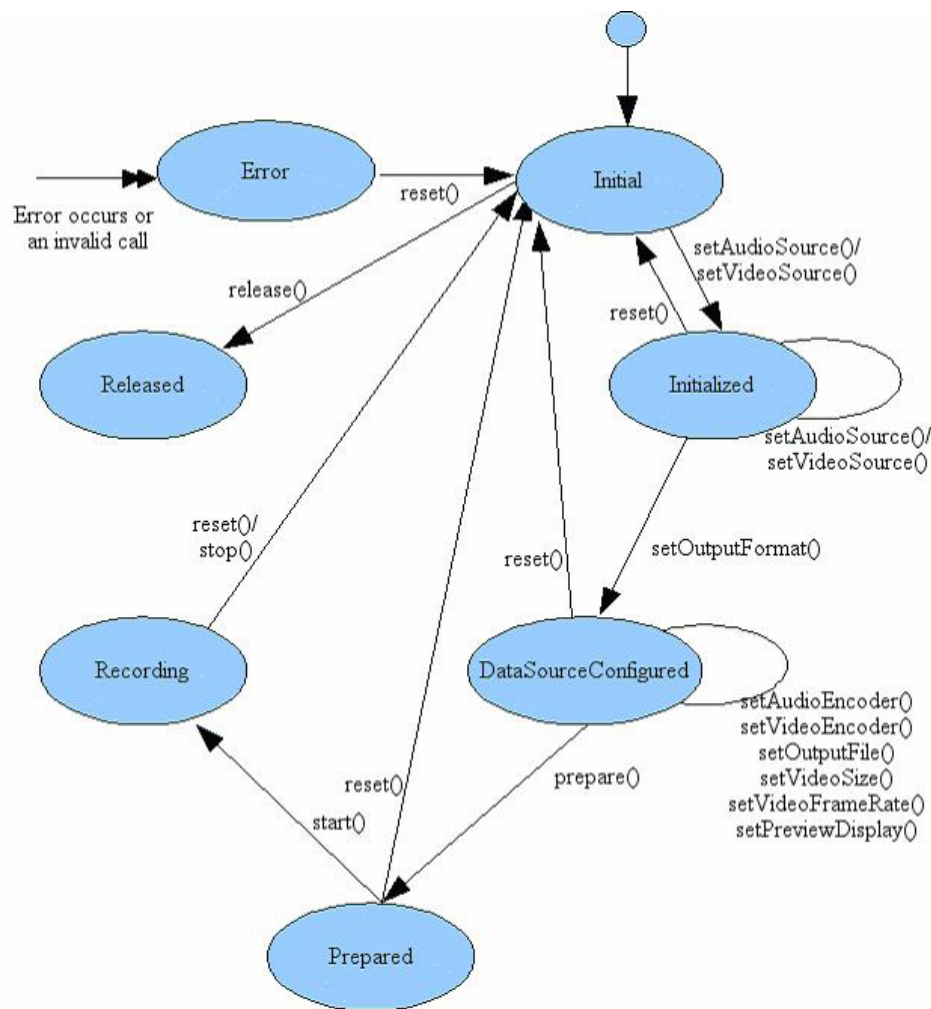
**Зауваження:** з станів `Paused`, `Playback Completed` можна повернутися до відтворення викликом методу `start ()`. Зі стану `Stopped` перш, ніж повернутися в стан відтворення, необхідно пройти через підготовку медіа-вмісту.

Виклик методу `seekTo ()` дозволяє поміняти місце відтворення.

- **кінець (End)** - кінець життєвого циклу `MediaPlayer` об'єкта, в цей стан об'єкт переходить після виклику методу `release ()`.

Для запису аудіо та відео `Android` надає клас `MediaRecorder`. Діаграма життєвого циклу екземпляра класу `MediaRecorder` представлена на рис. 10.2. Овали представляють стану об'єкта `MediaPlayer`, дуги показують виклики яких методів необхідно виконати, щоб змінити стан об'єкта `MediaPlayer`. Дуги з однією стрілкою представляють виклики синхронних методів, з двома стрілками - виклики асинхронних методів.





**Рис. 10.2.** Життєвий цикл екземпляра класу MediaRecorder

## 10.5 ВИКОРИСТАННЯ ВБУДОВАНОЇ КАМЕРИ

Платформа Android включає підтримку камери, доступною на пристрої, що дозволяє додаткам отримувати фотографії та записувати відео. Для вирішення цих завдань, існує два способи:

1. Безпосереднє звернення до камери;
2. Використання намірів (Intent) для виклику наявної програми.

## 10.6 ВЗАЄМОДІЯ З СИСТЕМАМИ ПОЗИЦІОНУВАННЯ

Системи позиціонування дозволяють визначити місце розташування в деякій системі координат, зазвичай визначаються широта і довгота.

Так як смартфон є мобільним телефоном, йому доступні методи, які зазвичай використовуються мобільними телефонами для визначення свого місця розташування.

- По-перше, смартфон постійно зв'язується з стільниковою вишкою, в зоні дії якої він знаходиться. Кожна стільникова вишка в світі має унікальний ідентифікатор, званий ідентифікатором стільникового (Cell ID), а також для неї точно відомі широта та довгота її розташування. У зв'язку з цим, смартфон, знаючи ідентифікатор стільникового, в якій він знаходиться, може отримати географічні координати центру цієї стільникового. Радіуси сот варіюються залежно від того, наскільки активний мережевий трафік очікується в конкретному районі. Зрозуміло, такий спосіб позиціонування дає дуже приблизні результати, що називається: "плюс-мінус трамвайна зупинка".

- По-друге, найчастіше смартфон опиняється в зоні дії більш, ніж однієї стільникової вишки. В сучасних мобільних технологіях, починаючи з покоління 2G, стільниковий вишка може визначити, з якого напрямку приходить сигнал. У випадку, коли телефон знаходиться в зоні дії двох або трьох стільникових веж, вони можуть виконувати тріангуляцію його розташування. Телефон може запитати у мережі інформацію про те, де він знаходиться. Така техніка визначення місця розташування може бути дуже точною і не вимагає установки додаткового обладнання.

Додатково до можливостей визначення місця розташування, доступним звичайних мобільних телефонів, більшість смартфонів укомплектовані супутниковими системами глобального позиціонування (Global Positioning System, GPS). В даний час найбільш поширеними у світі системами глобального супутникового позиціонування є: GPS, розроблена та реалізована в США, і система ГЛОНАСС (Глобальна навігаційна супутникова система), радянська, а пізніше російська супутникова система навігації. Багато смартфони можуть використовувати сигнали відразу від двох навігаційних систем, що дозволяє серйозно збільшити надійність і точність визначення координат, насамперед, в міських умовах.

В додаток до перерахованих вище методів позиціонування, додається можливість використання сигналів WiFi, Bluetooth і NFC, а також внутрішнього сенсора для більш точної геолокації, особливо всередині приміщень.

У цьому розділі нас, в першу чергу, буде цікавити можливість додавання в додатки здібностей визначати координати пристрою і працювати з картами. При

створенні додатків, що враховують поточне місце розташування, під Android можна скористатися GPS і визначенням місця розташування в мережі (за допомогою Network Location Provider ). Незважаючи на те, що GPS дає більш точні результати, він не дуже добре працює в приміщеннях (частіше не працює), він сильно витрачає заряд батареї і швидкість визначення координат не завжди відповідає очікуванням користувача. Network Location Provider визначає координати, використовуючи сигнали стільникових веж і WiFi, може працювати як на вулиці, так і всередині приміщень, більш економно витрачає заряд батареї і працює швидше в порівнянні з GPS. Для отримання координат в додатку можна використовувати обидва способи або один з них на вибір.

Android надає додаткам доступ до геолокаційні можливостям мобільного пристрою, через класи пакета android.location. Центральним класом цього пакету є клас LocationManager, який надає доступ до системних сервісів для визначення координат пристрою.

У додатки можна додавати карти, використовуючи Google Maps Android API, яке автоматично управляє доступом до серверів Google Maps, завантаженням даних, відображенням карт і сенсорними жестами на карті. Також можна використовувати виклики API для додавання маркерів, багатокутників і зовнішніх прозорих шарів, а також для зміни користувацького уявлення окремих ділянок карти.

Ключовим класом в Google Maps Android API є клас MapView, який відображає карту з даними отриманими з сервісу Google Maps. Коли MapView має фокус, він може перехоплювати натискання клавіш і сенсорні жести для виконання автоматичного переміщення і зміни масштабу карти, а також може управляти мережевими запитами для отримання додаткових фрагментів карти. Цей клас так само надає всі елементи користувацького інтерфейсу, необхідні для управління картою.

Google Maps Android API не є частиною платформи Android, але доступний на будь-якому пристрої з Google Play Store, працюючому, починаючи з Android 2.2, через Google Play Services. Щоб забезпечити можливість інтеграції Google Maps в додатки, в Android SDK необхідно встановити бібліотеку Google Play Services.