**Lab1**

**Android 環境建置與專案架構**

本節目的：

* 建置環境並實做出第一個APP。
* 了解Android Studio的開發環境與查看專案。
  1. Android 環境建置
     1. Android Studio開發工具

Step1 開啟Android Studio網址：https://developer.android.com/studio/，下載Android Studio開發環境，如圖1-1所示。

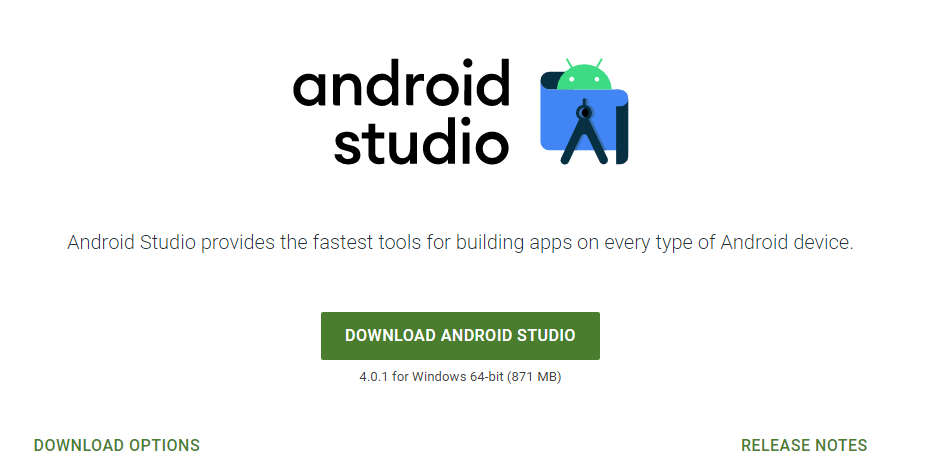


圖1-1 下載Android Studio

Step2 閱讀並同意Android Studio下載聲明，如圖1-2所示。

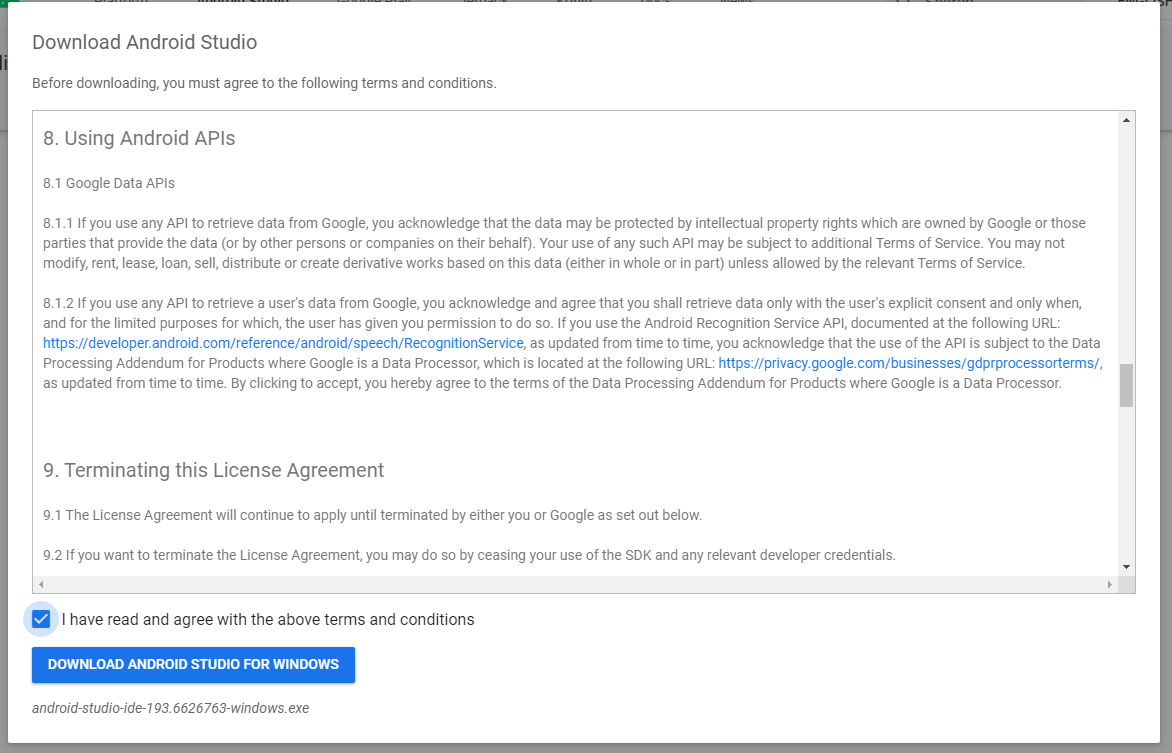


圖1-2 閱讀並同意下載聲明

Step3 開啟Android Studio安裝檔，然後點擊「NEXT」，如圖1-3所示。

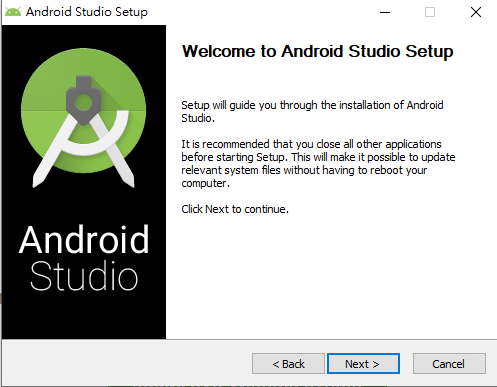


圖1-3 開始安裝Android Studio

Step4 選擇安裝套件，此處勾選「Android Virtual Device」，如圖1-4所示。

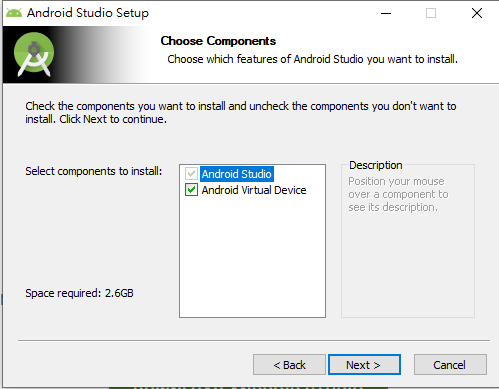


圖1-4 選擇安裝AVD套件

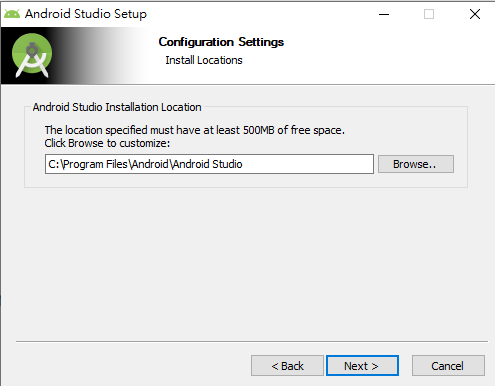
Step5 選擇檔案位置下一步，如圖1-5所示。

圖1-5設定安裝路徑

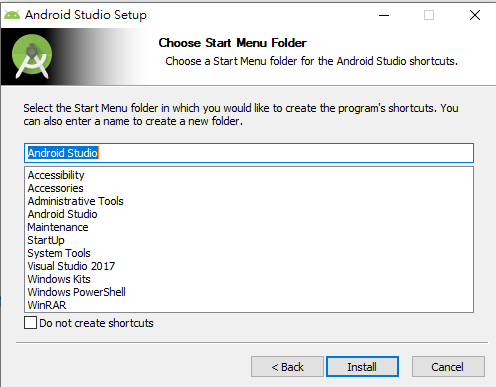
Step6 開始安裝Android Studio，如圖1-6所示。

圖1-6 安裝

* + 1. 建立模擬器

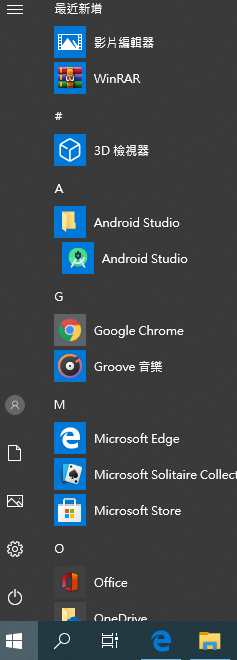
Step1 開啟位於開始目錄或桌面捷徑的Android Studio開發環境。

圖1-7 開啟Android Studio

Step2 第一次啟用會詢問是否匯入先前版本的設定，沒有的話選擇「Do not import ...」然後點選「OK」，如圖1-8所示。

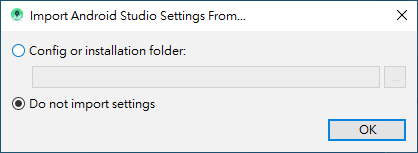


圖1-8 是否匯入先前版本設定

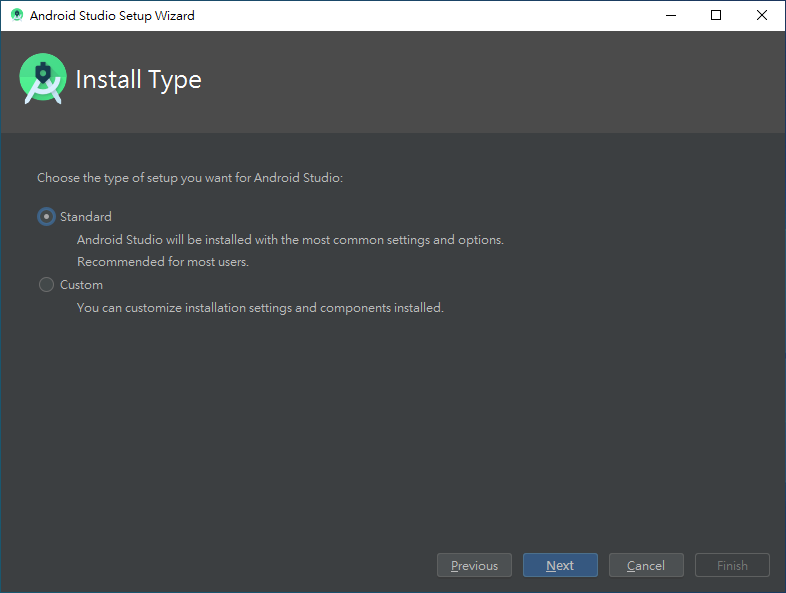
Step3 Android Studio程式下載完成，基本環境設定，如圖1-9所示。

圖1-9 選擇標準模式

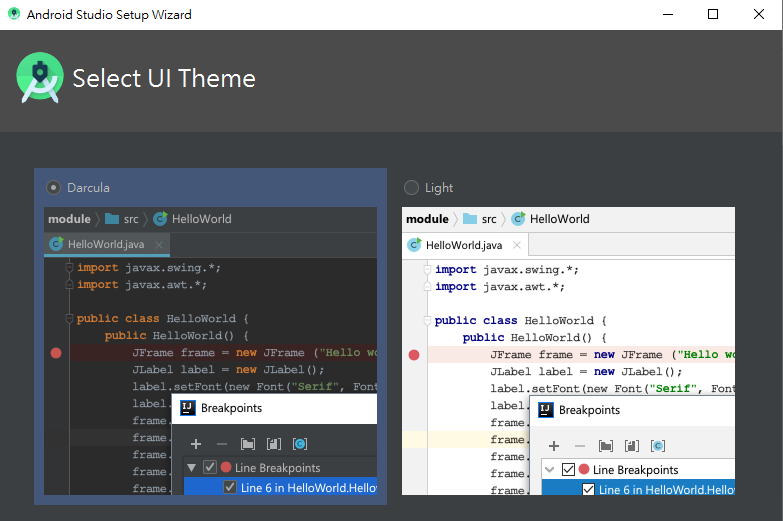
Step4 基本環境設定，設定背景色，如圖1-10所示。

圖1-10 選擇深色畫面

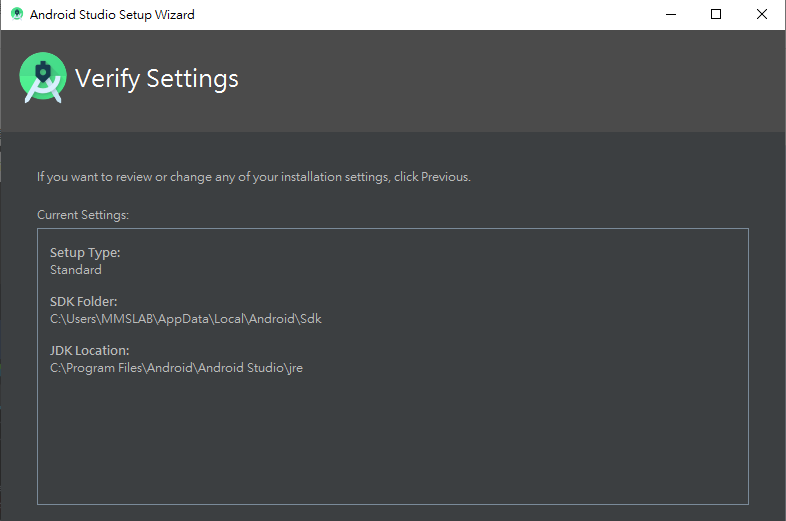
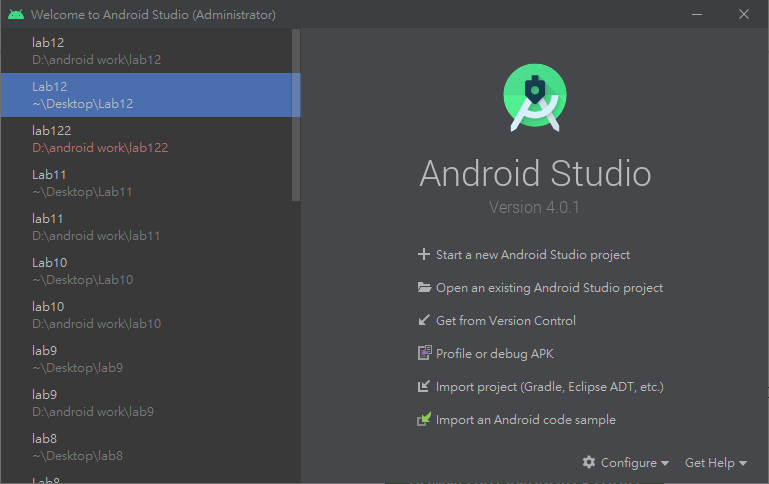
Step5 基本環境設定完成，下一步，如圖1-11所示。

圖1-11 基本環境安裝完成

Step6 安裝模擬器



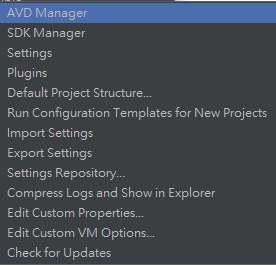
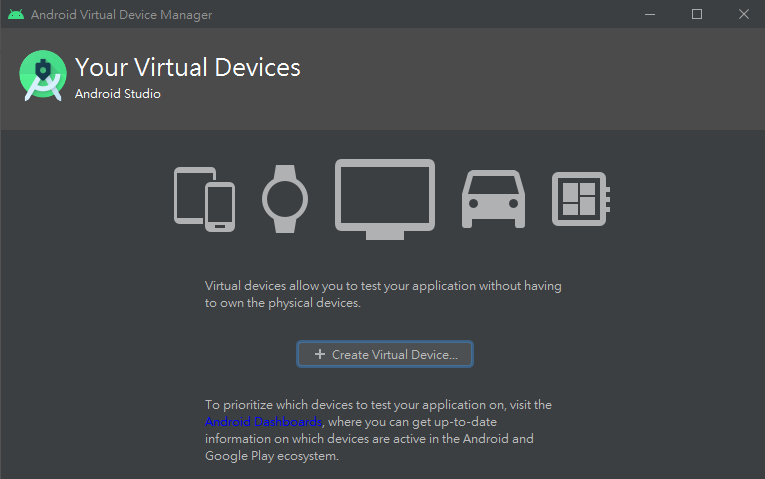


圖1-12 安裝模擬器

Step7建立模擬器，選擇手機類型。

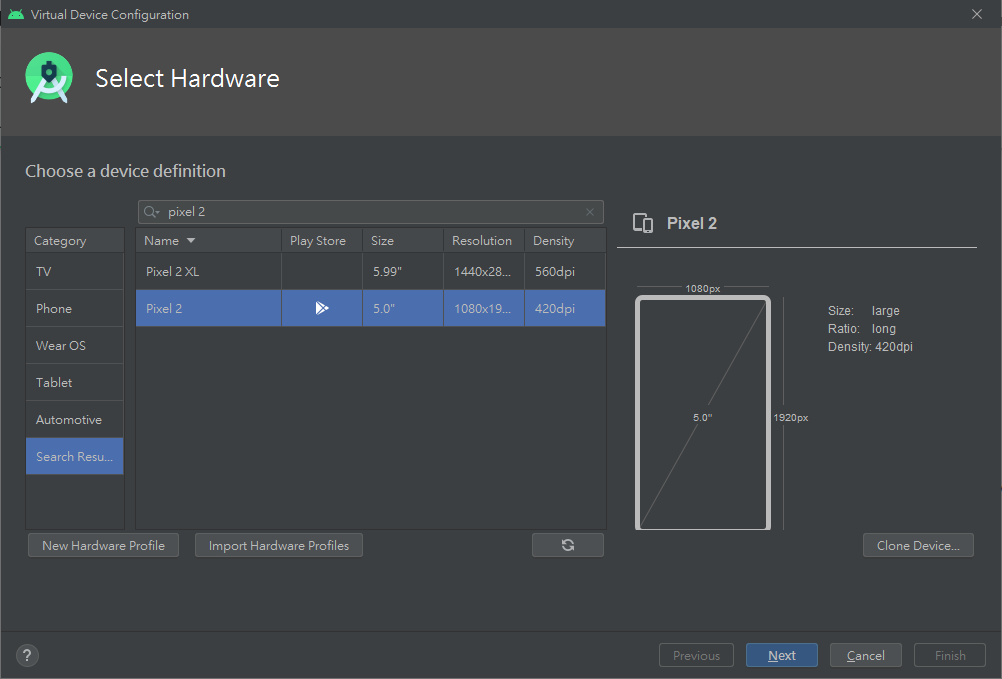


圖1-13 建立模擬器

Step8 下載作業系統版本。

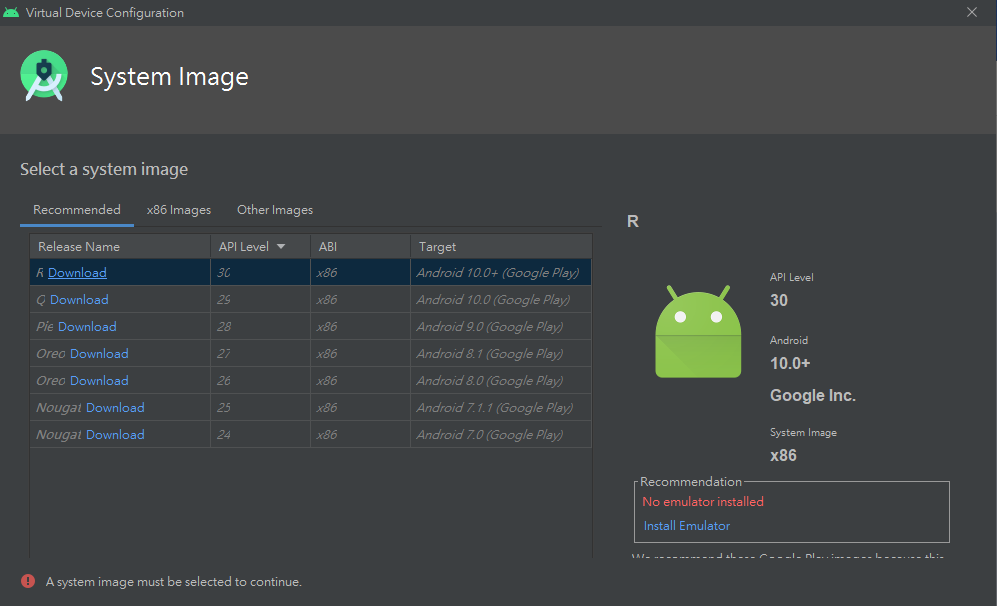


圖1-14 下載作業系統版本

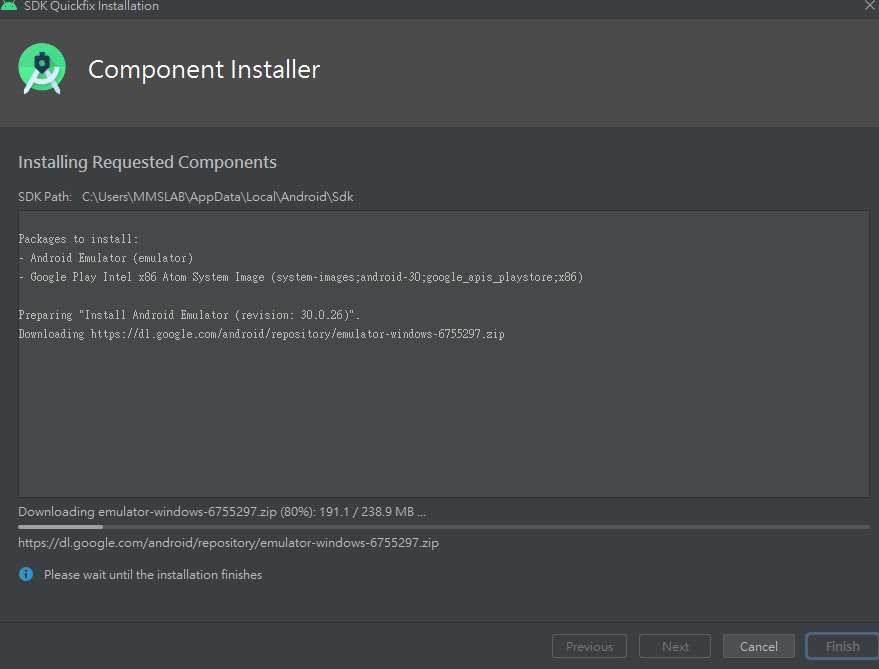
Step9 模擬器下載，如圖1-15所示。

圖1-15 安裝

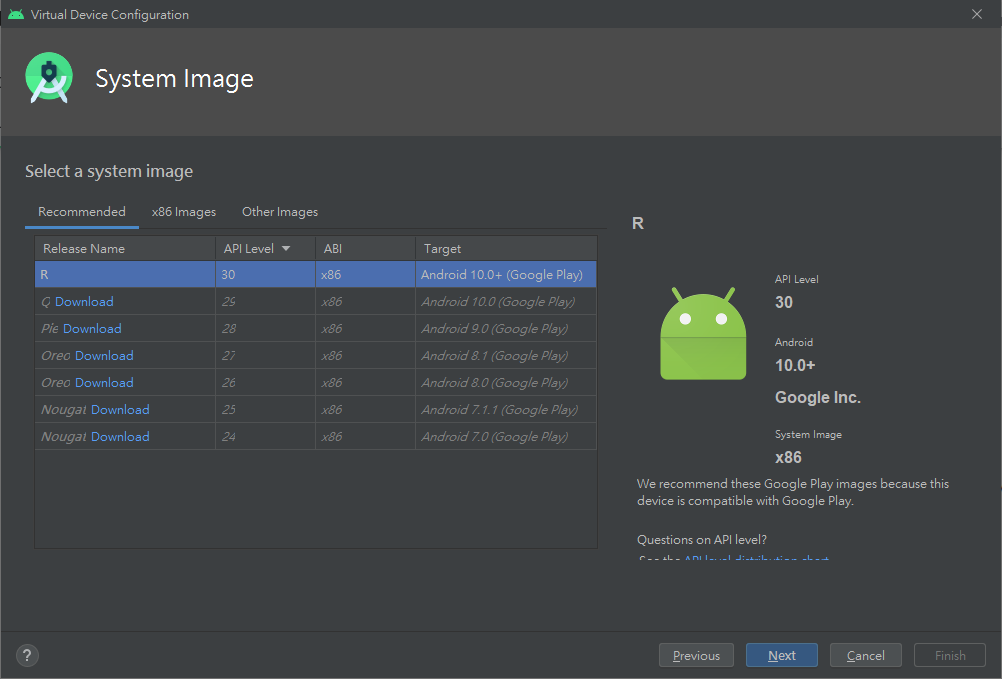
Step10 選擇作業系統版本，如圖1-16所示。

圖1-16選擇作業系統版本

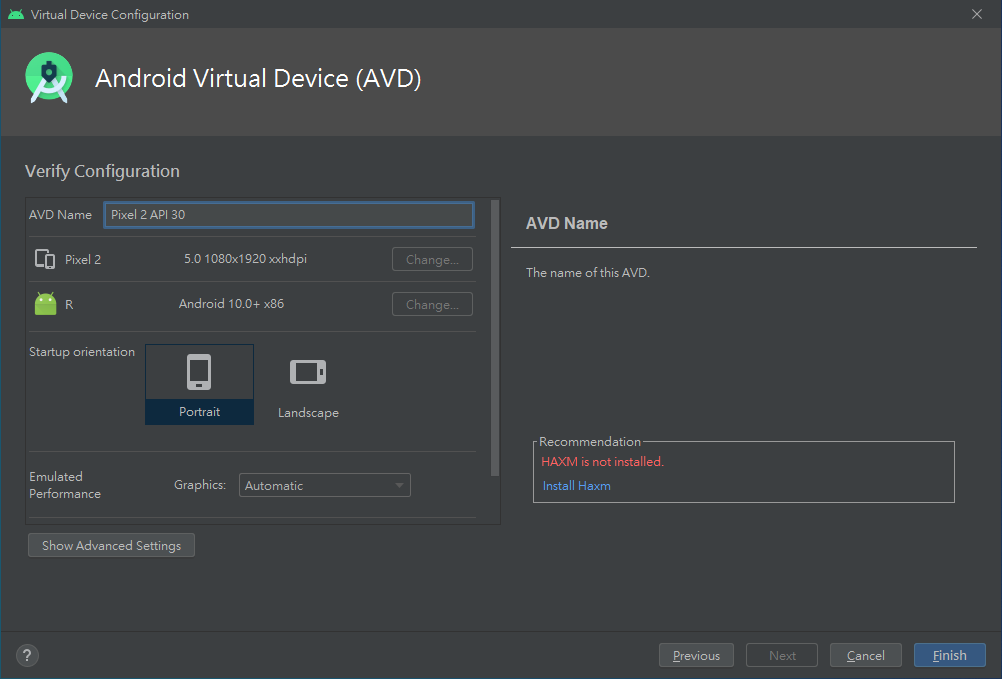
Step11 下載硬體加速器(Haxm)，如圖1-17所示。

圖1-17 選擇Install Haxm

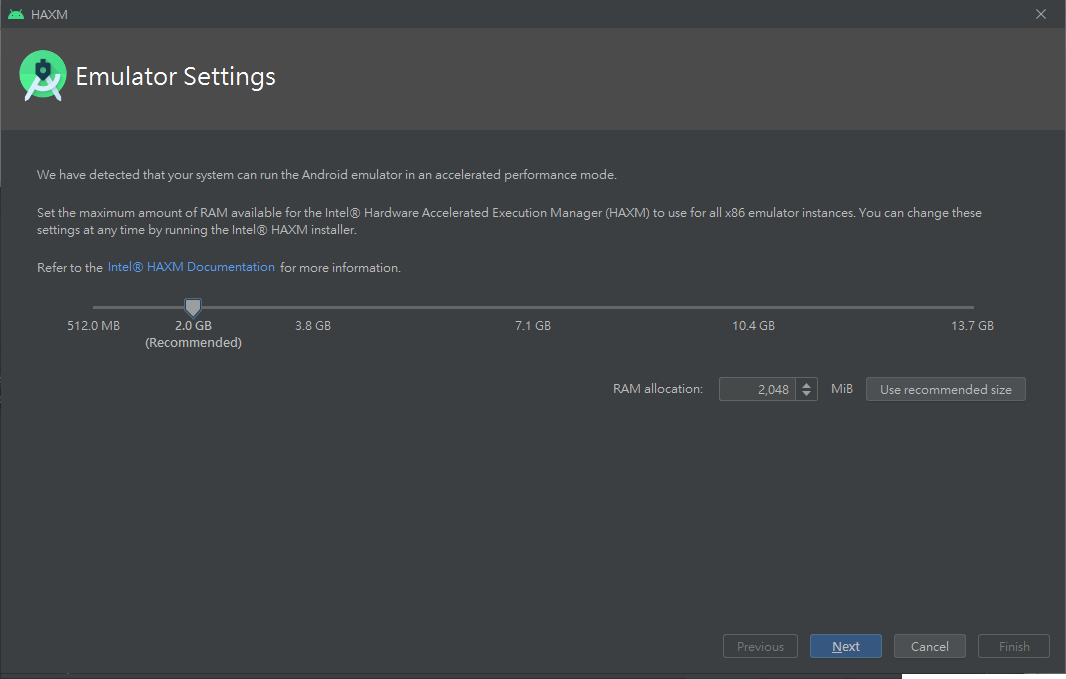
Step12 記憶體選擇2GB，如圖1-18所示。

圖1-18 記憶體選擇2GB

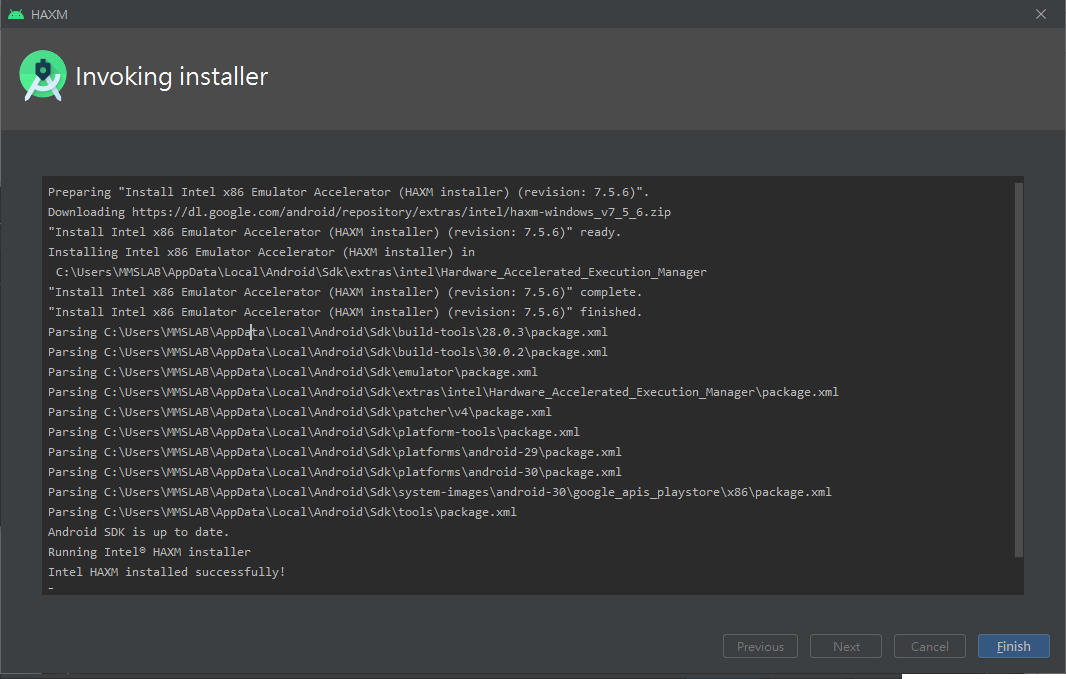
Step13 下載完成，如圖1-19所示

圖1-19 硬體加速器安裝完成

Step14 完成後，模擬器列表會出現你剛建立的模擬器，點擊右側箭頭開啟模擬器，如圖1-20所示。

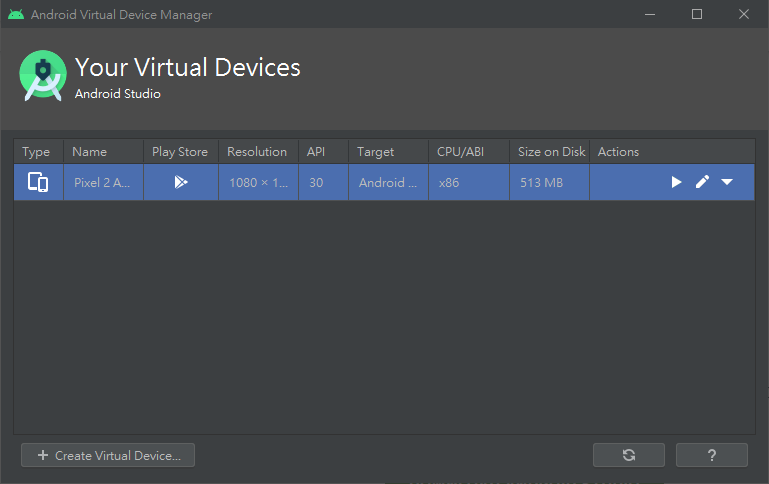


圖1-20 模擬器列表

Step15 看到如圖1-21的畫面，就代表模擬器啟動完成。

圖1-21 Android模擬器畫面

1.1.3建立APP專案

Step1 開啟位於開始目錄或桌面捷徑的Android Studio開發環境。

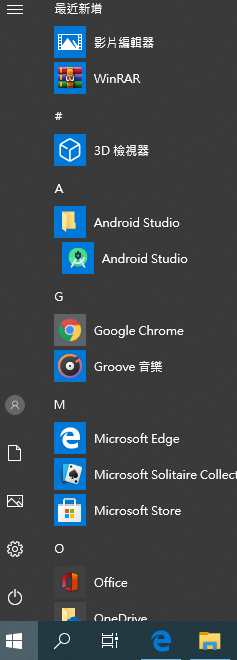


圖1-22 開啟 Android Studio

Step2 建立新專案。

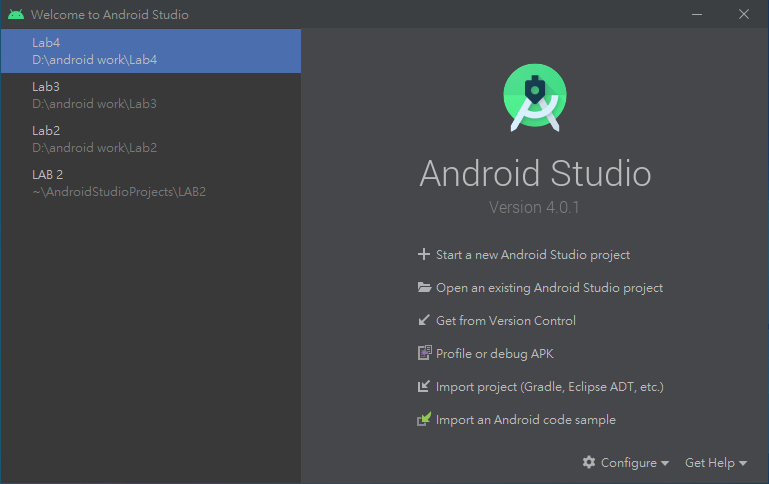


圖1-23 建立專案

Step3 選擇Empty Activity，下一步。

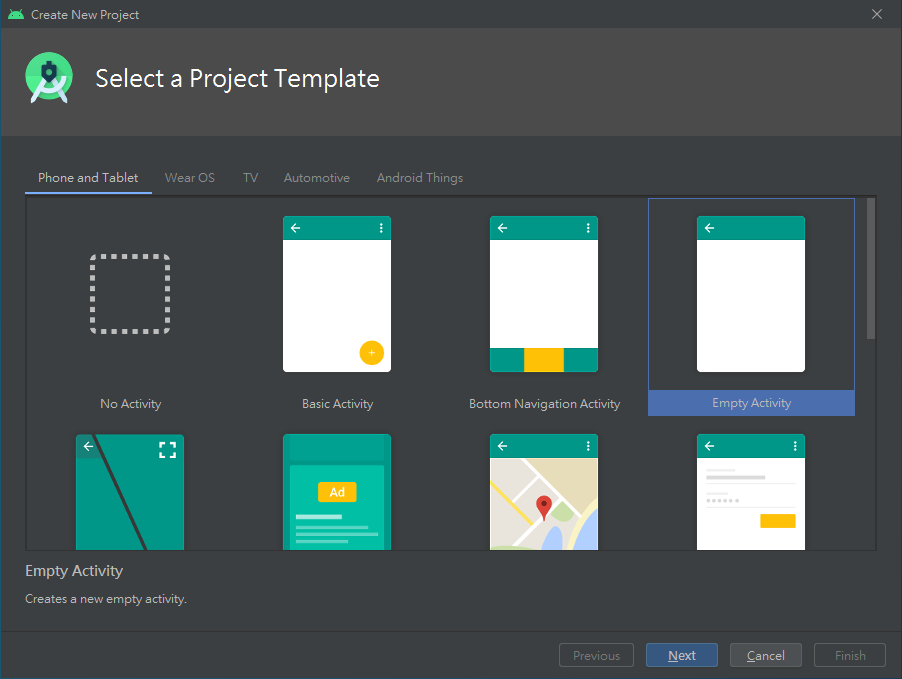


圖1-24 選擇開發模板

Step4 輸入專案名稱，完成。

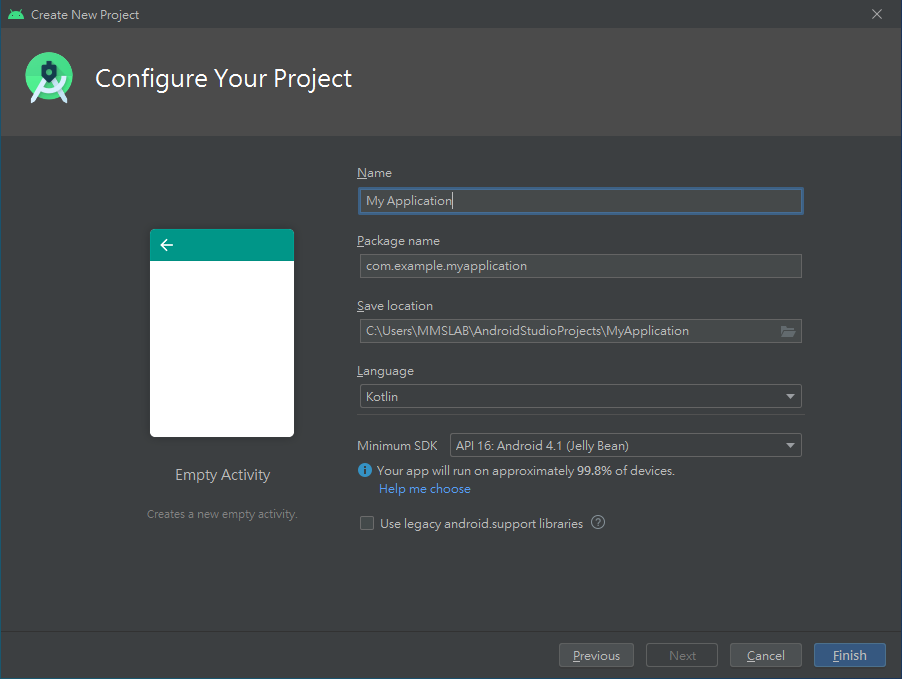


圖1-25 建立完成

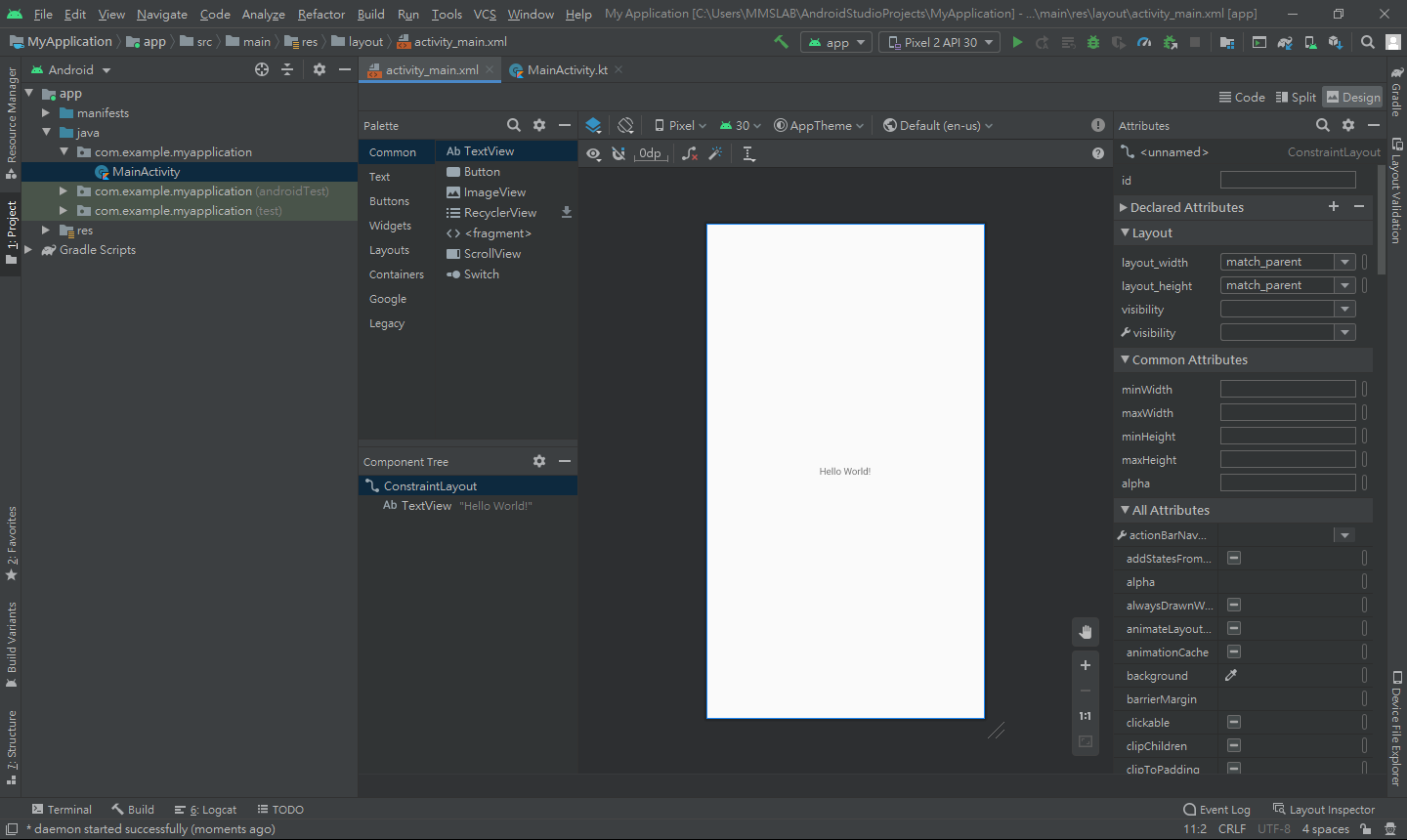
Step5 等待編譯完成。

圖1-26 建置完成

1.1.4執行APP專案

Step1 確認模擬器啟動且可運作之後，下一步開始編譯你的程式。點選位於Android Studio工具列中的綠色箭頭編譯程式，如圖1-27所示。



圖1-27 編譯你的程式

Step2 安裝完成，預設的Android專案會顯示「Hello World!」，如圖1-28所示。

圖1-28專案安裝成功

* 1. Android 專案架構

建置完成Android應用程式專案後，Android Studio左側表單內會顯示應用程式的配置目錄，如圖1-29所示，一般預設的配置模式下會是Android顯示配置。在Android顯示配置的app目錄之下會放置專案的主體，包含三個主要的子目錄：

* manifests－應用程式設定檔
* java－類別目錄
* res－資源目錄
* Gradle－自動化建構工具

以下分別對其介紹。

配置模式



圖1-29Android專案配置目錄

Android顯示配置的目錄不等於實際目錄。實際目錄需切換至Project顯示配置。

說明

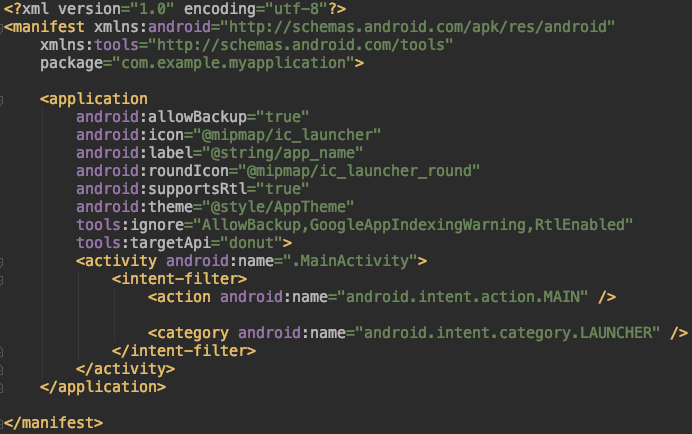
* + 1. 應用程式設定檔－AndroidManifest.xml



圖1-30應用程式設定檔AndroidManifest

於manifests目錄下，展開後可見AndroidManifest.xml，每一個應用程式的根目錄都必須包含圖1-30的AndroidManifest.xml檔案。系統在執行該應用程式的程式碼之前，需要向Android系統宣告應用程式的基本資訊，如應用程式的標題、圖示等，而這些將會被描述在AndroidManifest.xml之中。

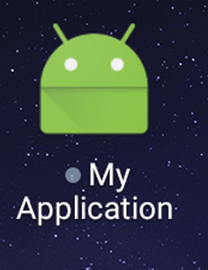
*詳細可見：*[*https://developer.android.com/guide/topics/manifest/manifest-intro.html*](https://developer.android.com/guide/topics/manifest/manifest-intro.html)



一個基本的AndroidManifest.xml中的描述的宣告內容如下：

* package：為應用程式的Java封裝命名。上架發佈的應用程式中package必須是唯一的，不能與其他應用程式有所重複，可以將package想像程式用於辨識或搜尋以發布的應用程式的識別名稱。
* application：用於定義應用程式相關的元件，設計中的屬性可預設應用程式的基本資訊，如下圖1-31所示。
  + 「android:icon」定義應用程式的圖示，預設為Android機器人圖示。
  + 「android:label」定義應用程式的標題，預設為專案建置時所設的名稱。
  + 「android:theme」定義應用程式的主題風格，也可說是應用程式的基本樣式，其設定於application中將會預設給所有子頁面。

Icon:android機器人

Label: My Application

Theme：

灰底綠色標題欄風格

圖1-31 APP的Icon與Theme樣式

* activity：「application」底下需要描述應用程式在執行中會使用到的所有類別方法，activity為其中的Activity類別。
  + 包含使用者介面「activity」、後台服務「Service」、廣播「Broadcast」，這些類別在被執行前，系統會去查閱「application」是否有對應的描述，如果有個activity要使用，但是卻沒有描述在「application」之中，那系統將無法呼叫該activity，這是初學者最常犯的錯誤之一。
  + 一個新創建的專案，一開始系統就會產生一個「MainActivity」的activity類別，需要更多的activity或是其他類別就需要手動增加。
  + 類別名稱的描述必須包含package名稱，如「demo.myapplication.MainActivity」，不過如果該類別屬於同個package，則可省略為「.MainActivity」。



如果要使用Service或receiver元件就需要在此處加入對應的描述。

此章節中不需要加入。

* + 1. java－類別目錄

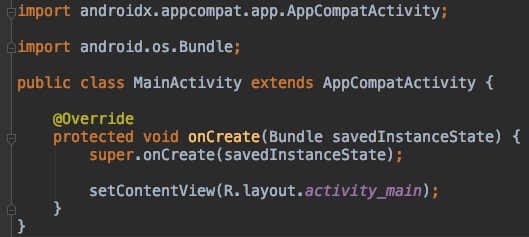
android的主要語言為Java與Kotlin，所有的程式碼都會被描述成類別結構放置於「src」目錄下，在Android顯示配置下則會顯示於圖1-32「java」目錄。



圖1-32專案的Java目錄

為了讓使用者能夠快速地對android進行開發，Google提供了相關的SDK套件，SDK套件提供使用者開發上的基礎框架，使用者可直接套用框架實作其功能。

如最常見使用的activity類別，使用者端可見的程式碼僅有短短數行，執行後便可產生介面，實際上應用程式當然不可能如此簡單就能產生畫面，能實現此功能的主要原因是透過繼承名為「AppCompatActivity」的類別框架，該框架中本身就有編寫能產生畫面的完整程式，而透過繼承，使用者可以完全不需要對於產生畫面的部分作程式編寫，僅需要引用內建的程式資源並在其之上加入自己的客製化程式碼，就可以簡單的實作自己的應用程式。



可以客製化修改的程式部分

繼承AppCompatActivity原生框架

* + 1. res－資源目錄

android的專案中，除了程式碼之外，還包含其他的專案資源，如放置於應用程式內部的圖檔、版面配置、顏色、風格主題等等，都算是專案的資源之一，會被放置於圖1-33「res」目錄下。



圖1-33專案的res資源目錄

依據用途的不同，最少又會區分成「drawable」、「layout」與「value」三個目錄，以下做簡單的說明：

* drawable：當應用程式需要使用圖檔時，會統一將需要的圖檔至於圖1-34目錄之下，圖檔資源可為「.png」或「.jpg」，或以xml格式設計的素材，也會放置於此目錄下。



圖1-34圖檔drawable

* layout：為使用者介面的版面配置檔，像是成形的畫面、按鈕畫面等等會放置於此目錄下。



圖1-35布局layout

* value：應用程式中可能使用的變數值，可放於此圖1-36目錄中，根據性質不同又可被分為字串（string）、顏色（color），區域大小（dimes）、風格（styles）等。



圖1-36數值Values

資源目錄在有新檔案加入或修改後，使用該資源的方法有分成XML的形式與程式碼的形式。

在XML中，我們使用「@目錄/檔名」的命名描述指定特定資源，如我們要使用value中的字串（string），可以透過「@string/app\_name」得到對應的字串。



圖1-37字串資源Strings

在Android中則需要透過R類別來連接資源。R類別由系統自動產生，系統會分配資源目錄下的所有檔案一組路徑，並被描述在R類別之中，可以用R.id.xxx的命名描述去指定特定資源，如圖檔（R.drawable.xxx）、版面配置（R.layout.xxx）與字串（R.string.xxx），透過命名描述去查閱R類別的對照表去找到的實體路徑。

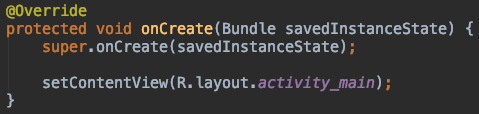




圖1-38透過R類別對照版面配置文件

* + 1. Gradle－自動化建構工具

Gradle是一個基於Apache Ant和Maven概念的專案自動化建構工具，在Android Studio中負責管理專案的設定如圖1-39所示，其中包含模組（Module）的設定檔，混淆碼規則與本地設定檔。



圖1-39自動化建置工具Gradle

* build.gradle：程式建構文件。在Android Studio中，每項專案可擁有多個模組（Module），而每個模組（Module）都會有一個自己的設定檔，用來紀錄模組（Module）所需的屬性、簽署或是依賴項目。



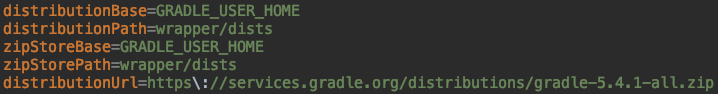
擴充

建置設定

依賴項目

專案的基本設定，包含專案的識別名稱、支援的最低與最高Android版本以及專案自身版本

* gradle-wrapper.properties：gradle-wrapper配置文件。一般這個文件是自動產生，開發者無須更動，除非你想手動指定gradle的版本。



* proguard-rule.pro：程式混淆規則配置文件。在APP發佈上架的過程中，保護程式碼是很重要的一步，透過Proguard將對類別、屬性和方法進行命名，增加反編譯後程式閱讀的難度，同時也可以降低apk檔案大小。
* gradle.properties：Gradle設定文件。專門用來設定全域資料，將敏感信息存放在gradle.properties中，可以避免將其上傳到版本控制系統。
* settings.gradle：程式設定文件。管理專案中的模組（Module），當我們要使用其他的Module時，也必須在settings.gradle中加入該模組（Module）的路徑。
* local.properties：本地設定文件。在實作專案的時候，通常會有些屬性是僅限於個人或開發用，不需要或是被禁止上傳到GitHub的屬性，例如SDK path或developer key之類的，這邊就提供將這類屬性定義在local.properties以供使用。