**Tìm hiểu về android studio**

1. Tổng quan về hệ điều hành Android

Android là hệ điều hành mã nguồn mở, dựa trên Linux Kernel, dành cho các thiết bị di động nói chung (điện thoại, máy tính bảng, đồng hồ thông minh, máy nghe nhạc,…).

 Có nghĩa là Android không chỉ giới hạn trong phạm vi một hệ điều hành cho điện thoại! Nó có thể được nhà sản xuất cài đặt lên đồng hồ, máy nghe nhạc, thiết bị định vị GPS, thậm chí là ô tô (các thiết bị Android Auto).

 Android cũng không phải là một thiết bị hay sản phẩm cụ thể, nó là một hệ điều hành dựa trên Linux, nguồn mở, linh hoạt.

 Hiện Android là một thương hiệu của Google. Có khả năng tùy biến rất cao và có thể chạy trên nhiều thiết bị, nhiều kiến trúc vi xử lý (ARM / x86). Tính đến nay, Android đã có các phiên bản (kèm tên mã) lần lượt là:

* Android 1.5 Cupcake
* Android 1.6 Donut
* Android 2.1 Eclair
* Android 2.2 Froyo
* Android 2.3 Gingerbread
* Android 3.2 Honeycomb – Phiên bản Android đầu tiên được thiết kế cho máy tính bảng.
* Android 4.0 Ice Cream Sandwich
* Android 4.1 Jelly Bean
* Android 4.2 Jelly Bean
* Android 4.3 Jelly Bean
* Android 4.4 KitKat
* Android 5.0 Lollipop
* Android 6.0 Marshmallow
* Android 7.0 Nougat

Về kiến trúc của hệ điều hành Android, chúng ta có thể xem qua sơ đồ sau:



Thoạt nhìn có vẻ rối rắm, nhưng chỉ cần để ý đến các tầng màu sắc. Tóm tắt về các tầng của kiến trúc này như sau (từ trên xuống nhé):

* *Tầng Applications*: Là tầng chứa các ứng dụng Danh bạ, Gọi điện, Trình duyệt, Nghe nhạc,… các ứng dụng này thường mua máy về chúng ta đã có sẵn rồi.
* *Tầng Framework*: Là tầng chứa các API để làm việc với hệ điều hành như lấy thông tin danh bạ, quản lý các Activity (Activity là gì thì giờ chúng ta chưa cần quan tâm, các bài sau sẽ giải thích kĩ), quản lý địa điểm, quản lý các View (cũng chưa cần quan tâm).
* *Tầng Libraries*: Chứa các thư viện, API gần như là cốt lõi của Android, bao gồm bộ quản lý bề mặt cảm ứng (Surface Manager), OpenGL (phục vụ cho việc dựng đồ họa phức tạp),…
* *Tầng Android Runtime*: Chứa các thư viện lõi của Android và máy ảo Dalvik Virtual Machine (từ Android 4 trở lên chúng ta có thêm máy ảo ART).
* *Tầng Kernel*: Là nhân lõi của hệ điều hành, chứa các tập lệnh, driver giao tiếp giữa phần cứng và phần mềm của Android.

 Trong quá trình làm việc, chúng ta sẽ gần như chỉ làm việc với tầng xanh lam (Applicationsvà Application Framework) và xanh lá (Libraries). Chương trình Android được viết bằng ngôn ngữ Java và được máy ảo DVM / ART trong mỗi thiết bị Android biên dịch ra mã máy.

1. Giới thiệu về Android Studio

Tháng 5 năm 2013, Google công bố Android Studio, một môi trường phát triển ứng dụng tích hợp (IDE) dành riêng cho Android, mã nguồn mở, dựa trên IDE Java IntelliJ của hãng JetBrains (đối thủ với Eclipse và Netbeans, vốn khá quen thuộc với dân lập trình Java).

 Android Studio chạy trên Windows, Mac và Linux, nhằm thay thế cho Eclipse Android Development Tool (ADT) vốn được sử dụng làm IDE chính trong các năm trước đó.

 Một số tính năng nổi bật:

* Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).
* Chức năng dò và sửa lỗi nhanh, hướng Android.
* Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.
* Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.
* Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.

Một số yêu cầu cấu hình lưu ý trước khi cài đặt:

**Windows**

* Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit)
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution

**Mac OS X**

* Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Java Runtime Environment (JRE) 6
* Java Development Kit (JDK) 7
* Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

On Mac OS, run Android Studio with Java Runtime Environment (JRE) 6 for optimized font rendering. You can then configure your project to use Java Development Kit (JDK) 6 or JDK 7.

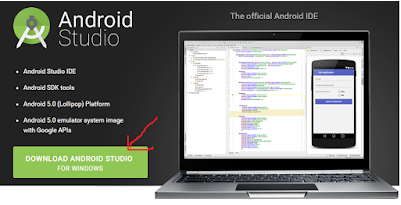
**Linux**

* GNOME or KDE desktop
* GNU C Library (glibc) 2.15 or later
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Oracle® Java Development Kit (JDK) 7

Tested on Ubuntu® 14.04, Trusty Tahr (64-bit distribution capable of running 32-bit applications).

1. Download và cài đặt android studio

Các bạn vào đường dẫn:  
                                         http://developer.android.com/sdk/index.html  
để download bản mới nhất và tiến hành cài đặt.



Sau khi tải về, tiến hành cài đặt.

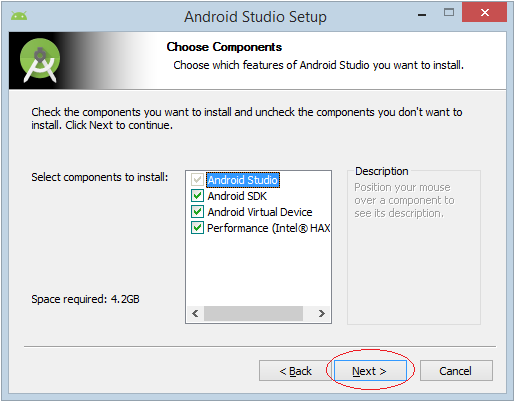


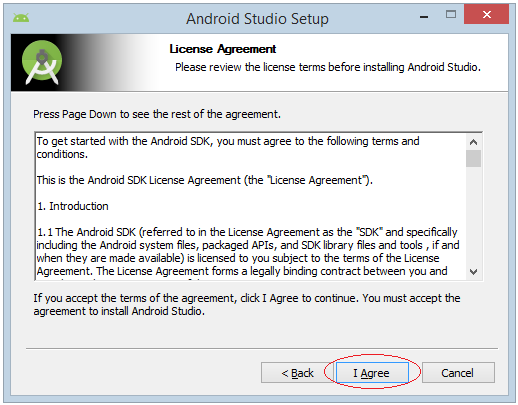
Lựa chọn tất cả các options.   
 The Android SDK (software development kit) là một tập hợp các công cụ được sử dụng để phát triển ứng dụng cho Android. Android SDK bao gồm:

* Các thư viện đòi hỏi
* Bộ dò lỗi (Debugger)
* Thiết bị mô phỏng (emulator)
* Các tài liệu liên quan cho Android API.
* Các đoạn code mẫu.
* Các hướng dẫn cho hệ điều hành Android.

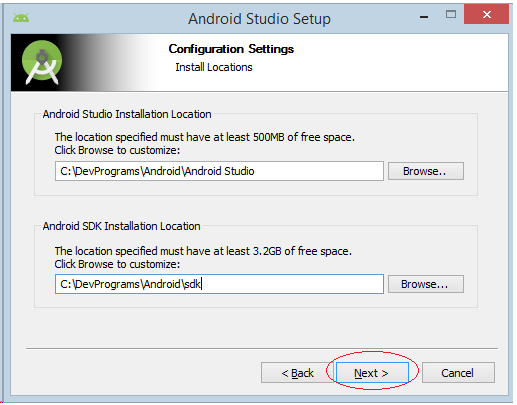
Android Virtual Device (AVD) là một cấu hình thiết bị, nó chạy với bộ giả lập Android (Android emulator). Nó làm việc với bộ giả lập để cung cấp một môi trường thiết bị ảo cụ thể, để cài đặt và chạy ứng dụng Android.

Intel Hardware Accelerated Execution Manager (Intel® HAXM) là một phần cứng hỗ trợ ảo hóa (hypervisor) có sử dụng công nghệ Intel Virtualization Technology (Intel® VT) để tăng tốc độ ứng dụng Android trên máy chạy phần mềm giả lập Android.





Chọn đường dẫn nếu muốn:

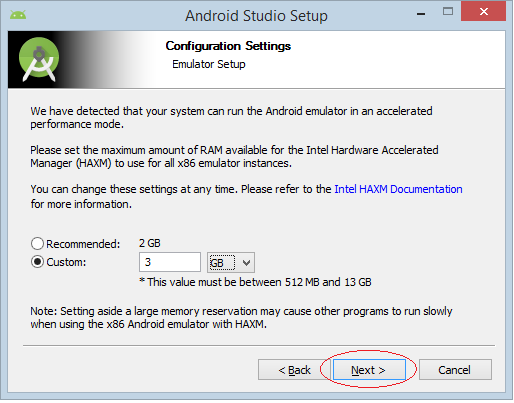


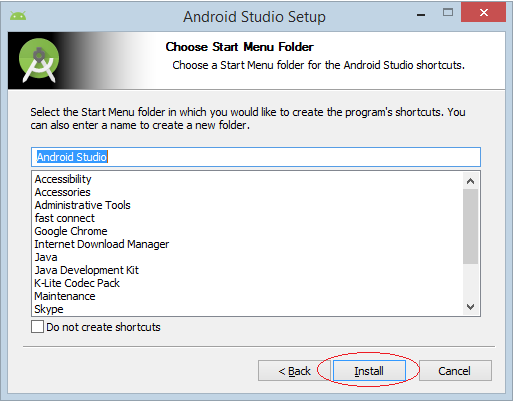
1. Đường dẫn cài đặt Android Studio (Dung lượng còn trống tối thiểu 500MB)
2. Đường dẫn lưu bộ Android SDK (Dung lượng tối thiểu còn trống 3.2 GB)

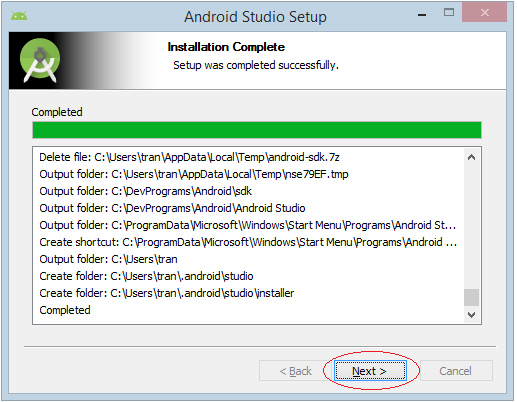
*Nếu không quan tâm các bạn có thể để mặc định*

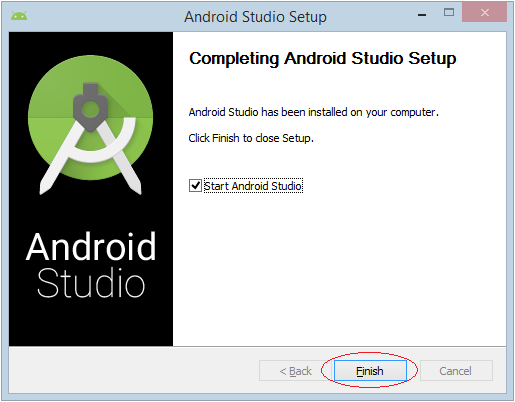
Nếu máy tính của bạn được trang bị phần cứng tốt, bộ giả lập Android (Android Emulator) có thể chạy được trong chế độ tăng tốc (Accelerated performance mode).

Bạn có thể cấu hình chỉ định số lượng RAM tối đa dành cho bộ quản lý tăng tốc phần cứng ( Intel Hardware Accelerated Manager – HAXM) . Khuyến nghị là 2GB.



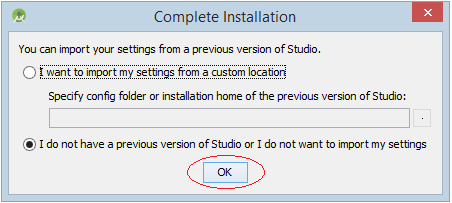




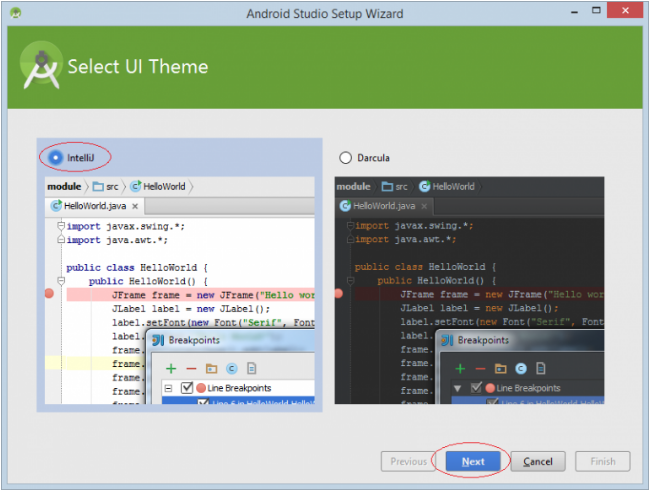


Bạn đã cài đặt xong **Android Studio**.

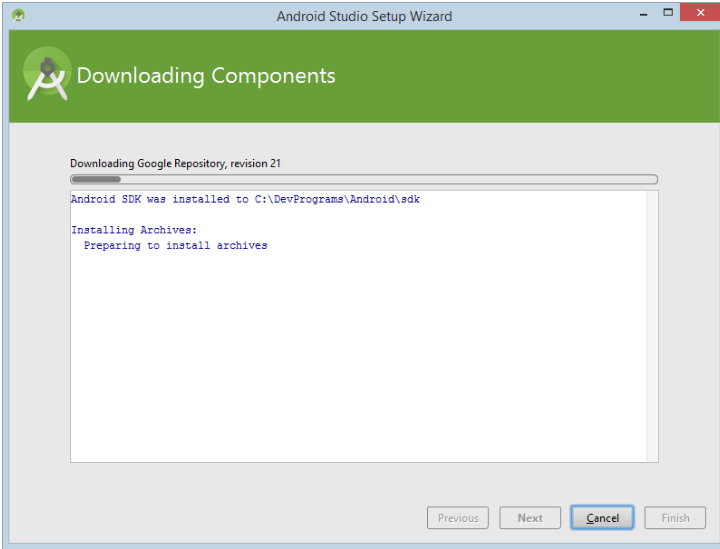
Trong lần chạy đầu tiên, **Android Studio** hỏi bạn có nhập khẩu các sét đặt từ phiên bản Android Studio mà bạn có thể đã cài đặt trước đó hay không. Bạn có thể chọn NO.

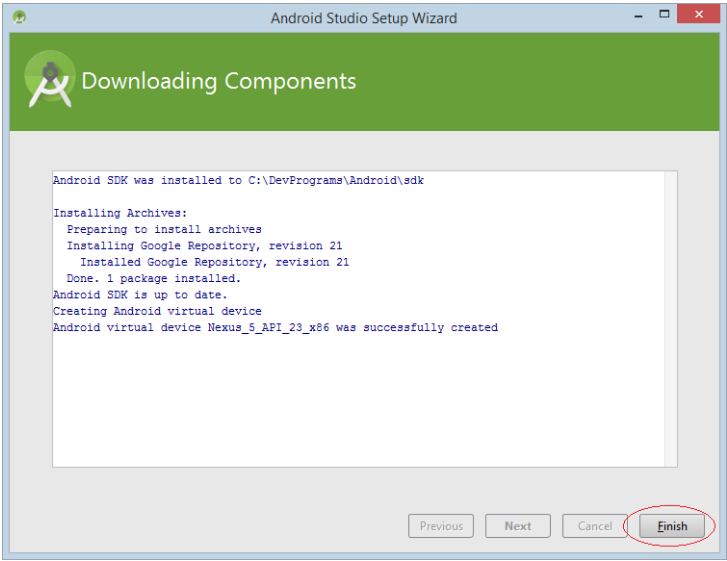


Lựa chọn một Theme mà bạn thích



Trong lần chạy đầu tiên, Android cần download một vài thành phần. Bạn cần chờ đợi cho tới khi tiến trình download và cài đặt hoàn thành.

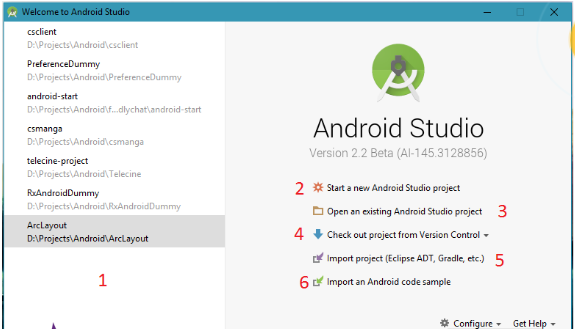




Chọn finish là xong.

1. Tạo 1 project với android studio

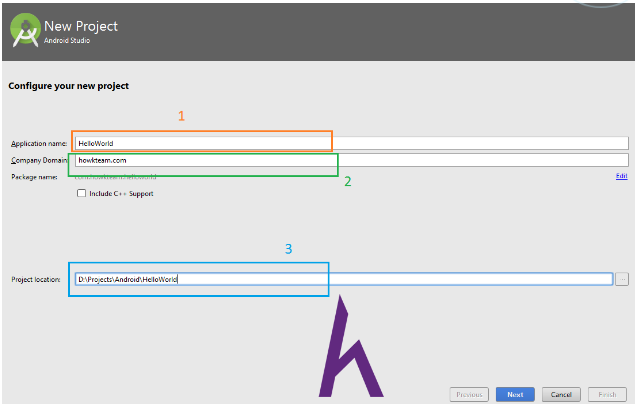
Click đúp vào biểu tượng Android Studio trên màn hình, giao diện mới đầu vào sẽ có dạng như thế này (phiên bản Android Studio 2.1 trở lên):



Để các bạn tiện theo dõi, mình đánh số các phần như trên. Nhiệm vụ của từng phần là:

1. Các project đã từng mở gần đây (nếu là lần đầu chạy Android Studio thì sẽ không thấy cái này).
2. Khời tạo một project Android mới để code, lát nữa chúng ta sẽ chọn phần này.
3. Mở một project Android có sẵn.
4. Lấy code Android từ một hệ thống quản lý mã nguồn đã có sẵn (Git, SVN, Mercurial,…).
5. Nhập code từ một project khác. Cái này cũng giống cái 2 nhưng nó sẽ mở và convert các project Android cũ (từ thời sử dụng Eclipse với Android SDK trước đây) sang dạng project Android mới của IntelliJ.
6. Lấy một số mẫu code ví dụ nguồn mở có sẵn của Google. Có rất nhiều ví dụ hữu ích.

Nào, chúng ta click vào lựa chọn thứ 2: **Start a new Android Studio project.**

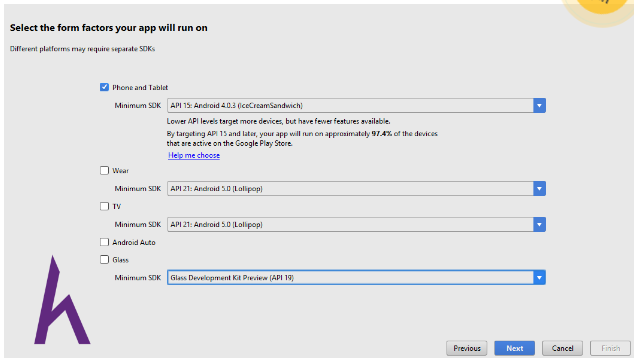


Và chúng ta đang ở màn hình đầu tiên của Wizard tạo project Android. Với các bạn đã từng lập trình Java thì nhìn lướt qua đoán mò cũng biết những ô nhập và các nút dùng để làm gì ở màn hình này. Nhưng nếu không biết thì mình xin giải thích như sau:

1. **Tên ứng dụng**. Nó cũng dùng để tạo một số thông tin khác, nhưng hiện tại thì các bạn chỉ cần biết là tên ứng dụng là đủ.
2. **Tên miền công ty**. Đặt gì cũng được, miễn là bao gồm số, gạch dưới và chữ. Phần này sẽ được Android Studio bóc tách để đặt làm tên package, theo thứ tự ngược lại, ngăn cách bởi dấu chấm (xem phần **Package name** ở ngay dưới).
3. **Nơi các file của project được lưu**. Các bạn có thể thay đổi bằng cách nhấn vào nút ba chấm bên cạnh.

Project chúng ta làm có tên HelloWorld. Các bạn có thể lấy ngay cách đặt tên như trên làm ví dụ mẫu. Sau khi điền xong các thông tin thì nhấn nút **Nex**

Sau khi hoàn thành, chúng ta sẽ đến một màn hình thứ hai của Wizard có dạng như sau:



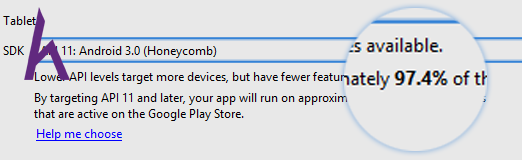
Nhiệm vụ của chúng ta bây giờ là chọn mức API. Các bạn thường nghe lỏm được ngoài đường người ta kháo nhau “máy mày an-roi mấy? thì nó chính là cái này đây.

Các bạn cũng thấy một loạt các dấu tick ở trên, tức là bạn sẽ cần chọn ít nhất một loại thiết bị để ứng dụng Android của bạn chạy trên đó.

Dĩ nhiên ví dụ của chúng ta làm cho điện thoại và máy tính bảng, đúng không? Ngoài ra còn có thêm:

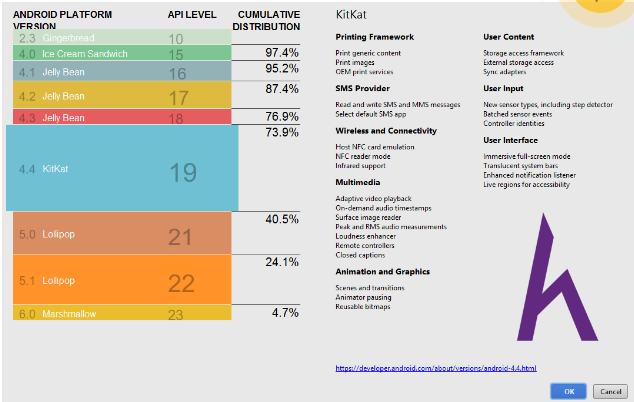
* **Wear**: Ứng dụng dành cho các thiết bị đeo tay tiện lợi như kính mắt, đồng hồ thông minh.
* **TV**: Ứng dụng dành cho Smart TV, hình như có mấy mẫu TV của Samsung chạy được.
* **Auto**: Dành cho các phương tiện đi lại (ô tô thông minh – cái này nước ngoài mới có hay sao ý).
* **Glass**: Kính thông minh, gần đây có bài báo nói về loại kính này. Đại khái là bạn có thể chạm vào gương kính để xem nhiệt độ, thời tiết. Cái kính này hay để ở phòng tắm.

Chúng ta cần chọn mức Minimum API cho ứng dụng. Ở phía dưới danh sách lựa chọn có một đoạn description ngắn nhưng các bạn chỉ cần để ý đến con số % . Con số này càng cao thì tỉ lệ ứng dụng của bạn chạy được trên các máy Android cũng càng cao.



(Giả sử nếu bạn chọn API 11 thì sẽ có khoảng 97.4% thiết bị Android trên thế giới có Google Play Store có thể chạy được ứng dụng của bạn).

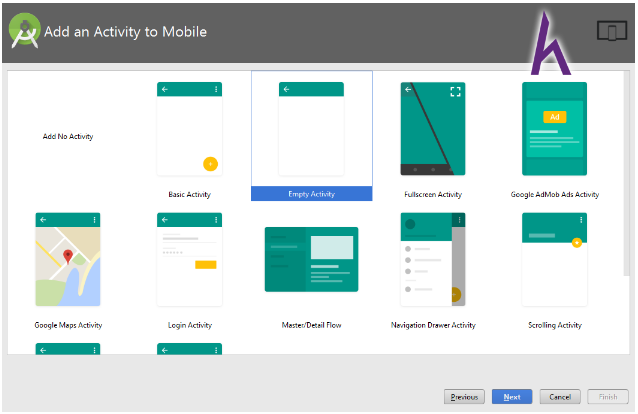
Để có cái nhìn cận cảnh hơn, nếu muốn, bạn click vào chữ **Help me choose**. Các bạn sẽ thấy một biểu đồ chi tiết về tỉ lệ % phổ biến của các mức API (cột trái) và các tính năng nổi bật của mức API đó (cột phải):



Chúng ta sẽ lấy mức API có mức % phổ biến tương đối cao là API 14: Android 4.0 (IceCreamSandwich).

Nhấn **Next**tiếp.

Sau đó chúng ta hiện đang ở đây:



Và lúc này chúng ta cần chọn mẫu màn hình (**Activity**) mà Android Studio tự động cung cấp cho chúng ta để làm việc nhanh hơn.

**Activity** là một màn hình đơn lẻ của ứng dụng Android. Ví dụ 2 màn hình dưới đây là 2 **Activity** nhưng cùng thuộc 1 app:

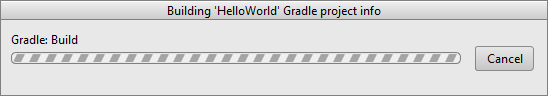
Danh sách các mẫu màn hình có ở đây là:

* Add no Activity: Không có activity nào cả. Ứng dụng trắng bóc luôn.
* Basic Activity: Một màn hình đơn giản, bao gồm một thanh điều hướng phía trên (xanh lá), một nút menu thu gọn ở góc trên bên phải, và một nút (**Floating Action Button**) trôi nổi ở góc dưới bên phải màn hình.
* Empty Activity: Một **Activity** cơ bản, chỉ có một thanh điều hướng trên cùng và một nút mũi tên trên đó.
* Fullscreen Activity: **Activity** này nó sẽ bao trùm toàn màn hình. Tức là bạn sẽ không thấy thanh status (cái thanh hiển thị vạch pin, giờ, sóng) sau khi vào màn hình này.
* Google Admob Ads Activity: Activity này cũng cơ bản, nhưng có gắn kèm quảng cáo giữa màn hình.
* Google Maps Activity: Activity này cũng như một activity cơ bản, chỉ là ở giữa màn hình có thêm cái bản đồ Google Maps.
* Login Activity: Activity bao gồm 1 ô nhập username, 1 ô nhập password và 1 nút Login. Rất tiện cho bạn nào muốn làm ngay một trang login cơ bản.
* Master/Detail Flow: Trang này thì hơi phức tạp một chút. Đại khái là trên các máy điện thoại, màn hình nhỏ thì nó sẽ là một danh sách gồm nhiều item, nhấn vào thì sẽ ra nội dung ở một Activity mới. Còn với màn hình lớn thì sẽ chia làm 2 cột: item ở cột trái tương ứng với nội dung chứa trong **Fragment** ở cột phải.

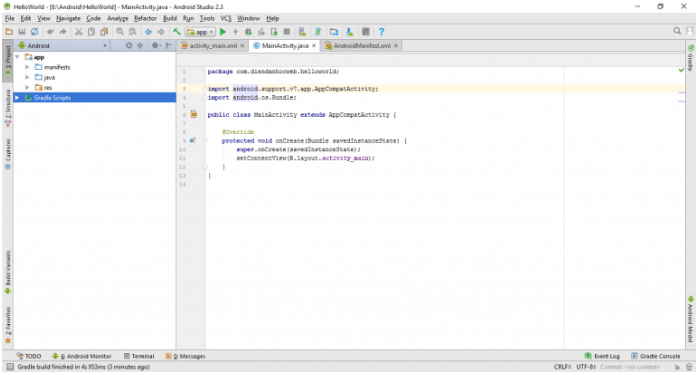
Chúng ta sẽ không chọn các mẫu có sẵn này, mà chọn **No Activity**. Để các bạn có thể hiểu được cách tạo một Activity từ đầu là như thế nào, và cách tạo một Activity thủ công theo đúng ý muốn mà không dựa dẫm vào mẫu có sẵn.

Ta chọn **Add no Activity** và nhấn **Next**.

Android Studio sẽ mất một lúc để tạo ra project. Nếu máy tính của bạn có ổ SSD và cấu hình cao thì sẽ rút ngắn được thời gian hơn một chút.



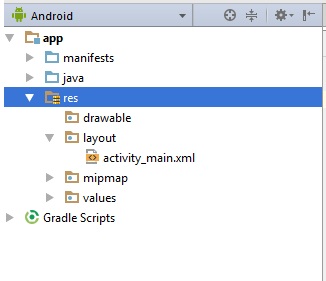
Sau đó thì chúng ta vào màn hình làm việc chính của Android Studio.



Vùng 1: Thanh công cụ giúp bạn thao tác nhanh các chức năng thường dùng khi lập trình trong Android Studio. Trong đó, quan trọng là chức năng Run   , Debug ứng dụng   và quản lý máy ảo

Vùng 2: Cấu trúc hệ thống tài nguyên của ứng dụng

* Thư mục manifests: chứa thông tin cấu hình của ứng dụng AndroidManifest.xml: tập tin XML chứa tất cả các thông tin cấu hình dùng để build ứng dụng và các thành phần của ứng dụng (activity, service,…). Mỗi ứng dụng đều có một tập tin AndroidManifest.xml. Trong ứng dụng, Activity nào muốn sử dụng đều bắt buộc phải có khai báo AndroidManifest.xml
* Thư mục java: chứa tất cả các file mã nguồn .java của ứng dụng
* Lúc này do ứng dụng của chúng ta chỉ có một màn hình màn hình MainActivity nên các bạn chỉ thấy MainActivity.java. Tương ứng với mỗi Activity thì file mã nguồn sẽ chứa các xử lý trên Activity đó. Activity nào được khởi chạy đầu tiên khi ứng dụng hoạt động sẽ được khai báo đầu tiên trong tập tin AndroidManifest.xml.
* Thư mục res: chứa các tài nguyên của ứng dụng, bao gồm các tập tin hình ảnh, các thiết kế giao diện, thực đơn,… của ứng dụng.



Vùng 3: Danh sách các control mà Android Studio hỗ trợ, bạn có thể tạo giao diện bằng cách kéo thả trực tiếp các control này vào vùng giao diện (Vùng 5) và Android Studio sẽ phát sinh ra mã lệnh XML cho bạn.

Vùng 4: hiển thị giao diện theo cấu trúc cây giúp bạn dễ dàng quan sát và lựa chọn control.

Vùng 5: vùng giao diện của thiết bị cho phép kéo thả các control. Chúng ta có thể chọn cách hiển thị theo chiều nằm ngang, nằm đứng, phóng to, thu nhỏ, lựa chọn các loại thiết bị hiển thị…

Vùng 6: Cửa sổ thuộc tính của control đang chọn, cho phép bạn thiết lập các thuộc tính cần thiết. Mặc định bạn sẽ kéo thả các control vào vùng giao diện (Design), nhưng nếu muốn, bạn có thể chuyển sang Text để thiết kế giao diện bằng cách viết các thẻ XML tương ứng.

