

## ĐỀ CƯƠNG BÀI GIẢNG

### **BÀI 1. GIỚI THIỆU VỀ LẬP TRÌNH DI ĐỘNG VÀ THIẾT KẾ GIAO DIỆN TRÊN ANDROID**

**Nội dung bài học trước khi lên lớp** (trang 2 đến 10)

- Tổng quan về Android
- Các đặc điểm, ưu điểm và nhược điểm
- Hướng dẫn cách cài đặt công cụ Android Studio
- Cài đặt project đầu tiên: HelloWorld

**Nội dung bài học thực hiện lên lớp** (Trang 10 đến hết):

- Cấu trúc ứng dụng của Android
- Tìm hiểu về Grade

**Nội dung bài học sau khi lên lớp:** Phiếu bài tập bài 1 đến bài 6

#### **NỘI DUNG BÀI HỌC**

1. Tổng quan về Android.....	2
1.1. Ưu điểm.....	2
1.2. Nhược điểm.....	2
1.3. Kiến trúc phân tầng của Android .....	3
2. Hướng dẫn cài đặt và chạy ứng dụng HelloWorld trên Android Studio .....	4
2.1. Hướng dẫn cài đặt Android Studio .....	4
2.2. Tạo project trong Android Studio .....	7
2.3. Hướng dẫn cấu hình máy ảo trong Android Studio .....	10
3. Cấu trúc ứng dụng của Android .....	12
3.1. Grade là gì? .....	12
3.2. Giới thiệu Tasks. ....	12
3.3. Làm Quen Với Gradle Wrapper.....	13

## 3.4. Các File Gradle..... 14

## 1. Tổng quan về Android

Android là một hệ điều hành mã nguồn mở được thiết kế dựa trên nền tảng Linux, dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.

Thời gian đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google, sau đó Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt sau đó 2 năm.

Giao diện người dùng của Android dựa trên nguyên tắc tương tác trực tiếp, sử dụng cảm ứng chạm đa điểm tương tự những động tác ngoài đời thực như vuốt, chạm, kéo dẫn và thu lại để xử lý các đối tượng trên màn hình.

### 1.1. Ưu điểm

Android là hệ điều hành có mã nguồn mở nên khả năng tùy biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp từ Google nhưng vẫn phải tuân thủ các quy tắc mà Google đưa ra trong việc thiết kế ứng dụng, bảo vệ người dùng.

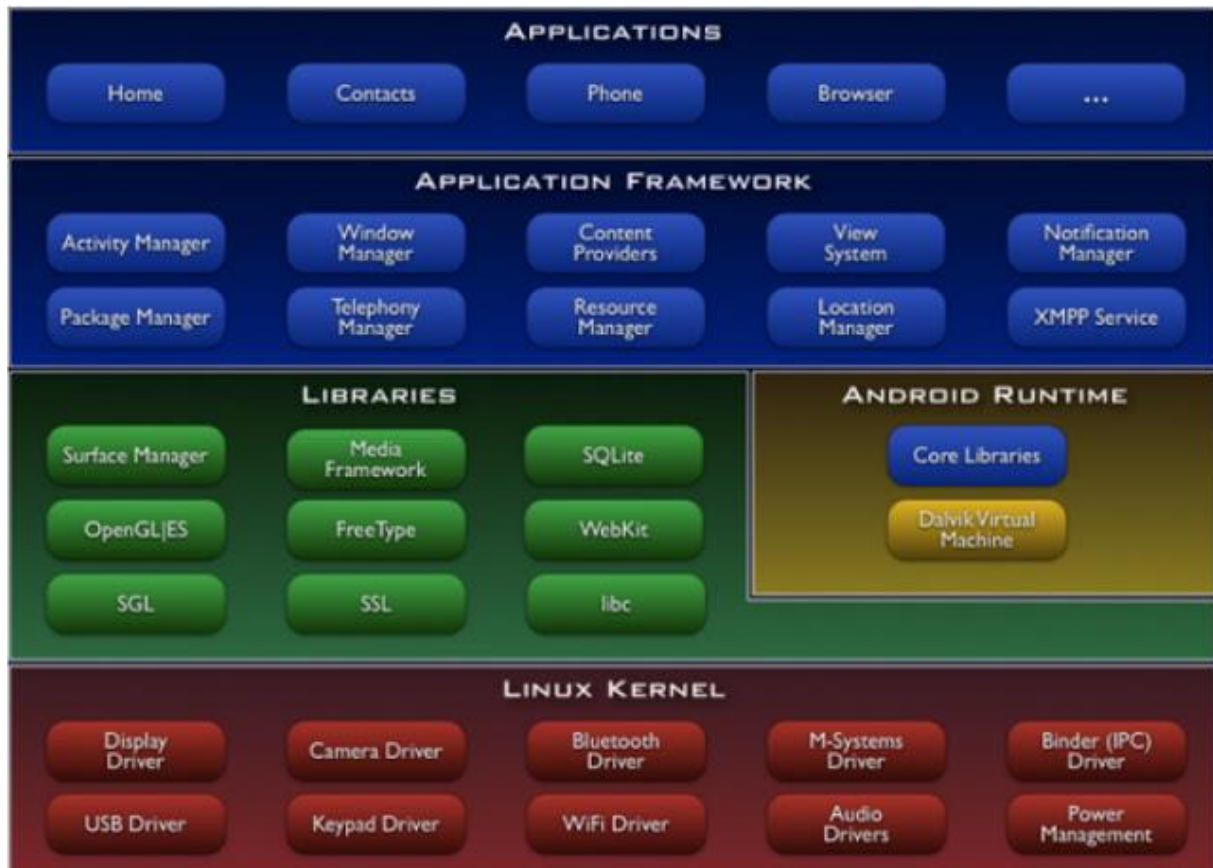
Sản phẩm đa dạng, rất nhiều hãng điện thoại, thiết bị công nghệ đã chọn Android làm hệ điều hành cho các thiết bị của họ với giá cả hợp lý từ thấp đến cao.

- Kho ứng dụng Google Play lớn
- Thân thiện và dễ sử dụng
- Khả năng đa nhiệm , chạy nhiều ứng dụng

### 1.2. Nhược điểm

- Do tính chất mã nguồn mở có thể được thiết kế bởi người dùng, nên nhiều phần mềm không được kiểm soát chất lượng có lỗi bảo mật vẫn được sử dụng. Vì vậy, phần mềm có thể dễ bị nhiễm virus và có một số lỗi bảo mật.
- Kho ứng dụng quá nhiều sản phẩm, do đó dẫn đến việc khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.
- Thường xuyên có nhiều phiên bản khác nhau dẫn tới phải cài đặt, cập nhật phần mềm liên tục đôi khi phải thay thế, nâng cấp thiết bị, phần cứng.

### 1.3. Kiến trúc phân tầng của Android



- Linux Kernel làm nhiệm vụ giúp hệ điều hành có thể giao tiếp với phần cứng của thiết bị. Linux Kernel chịu trách nhiệm cho các trình điều khiển thiết bị, quản lý nguồn điện, quản lý bộ nhớ, quản lý thiết bị và truy cập tài nguyên trong thiết bị di động.
- Android có Native Libraries bao gồm nhiều tập hợp của các thư viện như WebKit, OpenGL, FreeType, SQLite, Media, SSL...
- Android Runtime chứa thành phần quan trọng máy ảo gọi là DVM (Dalvik Virtual Machine) có trách nhiệm chạy ứng dụng android. DVM có phần tương tự như JVM (Java Virtual Machine) nhưng được tối ưu hóa cho các thiết bị di động như tiêu thụ ít bộ nhớ hơn và tăng hiệu suất hoạt động tốt hơn.
- Application Framework bao gồm tập hợp những thư viện cho phép các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.
- Application là tầng ứng dụng, nơi người lập trình sản xuất các ứng dụng và đẩy lên trên Google play hay Apple Store như những ứng dụng Facebook, Gallery, Music...

## 2. Hướng dẫn cài đặt và chạy ứng dụng HelloWorld trên Android Studio

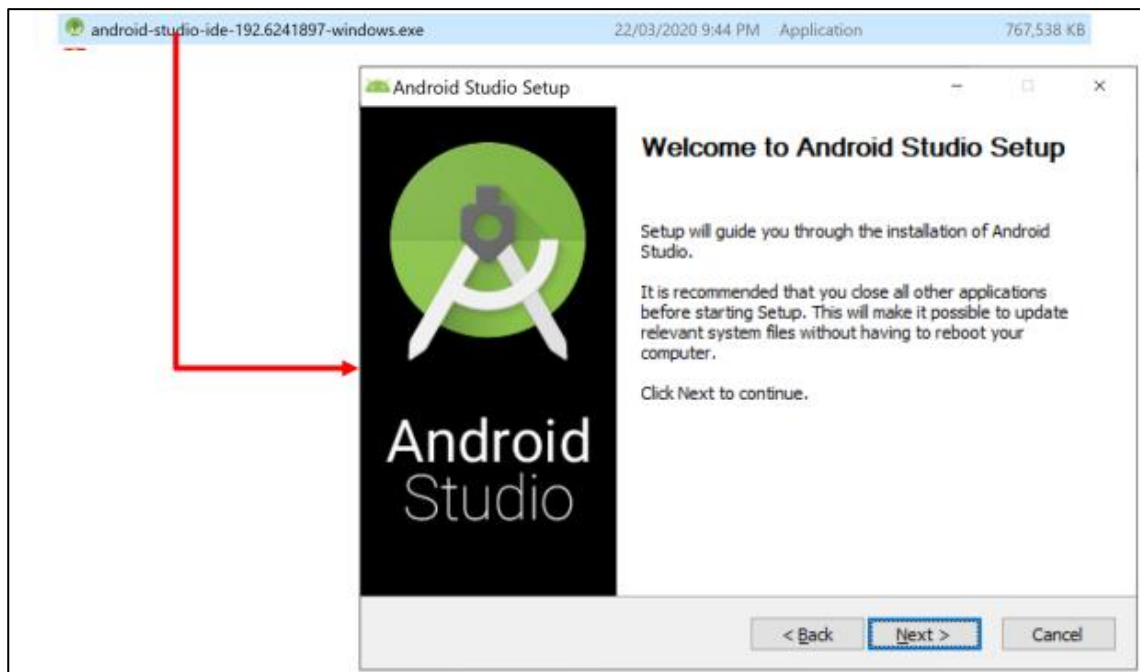
### 2.1. Hướng dẫn cài đặt Android Studio

Android Studio là bộ công cụ lập trình thông dụng nhất lập trình với Android. Để cài đặt, truy cập vào đường dẫn: <https://developer.android.com/studio>

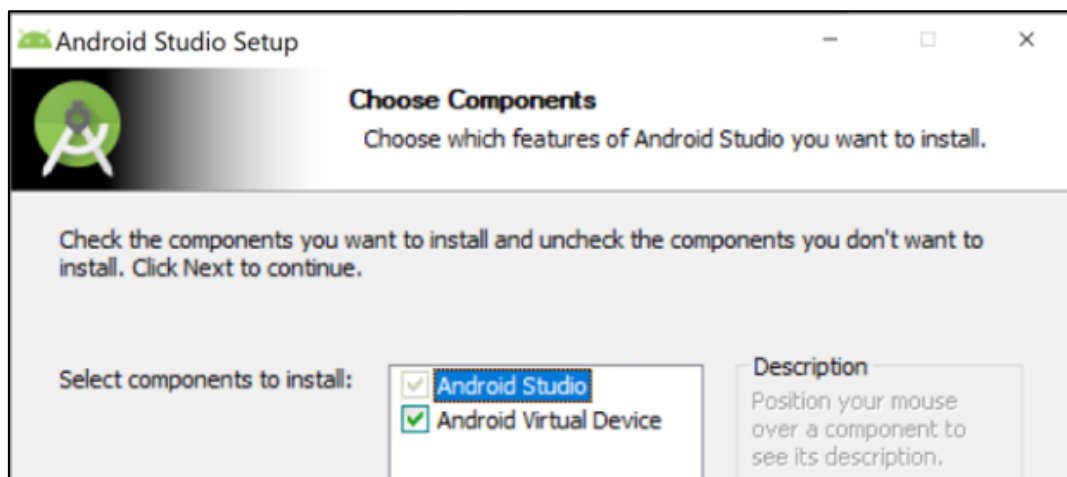


**Bước 1:** Nhấn chuột vào nút “DOWNLOAD ANDROID STUDIO”. Khi cài đặt sẽ có 2 thành phần đó là: Cài đặt công cụ lập trình (android studio) và cài đặt thư viện hỗ trợ lập trình Software development kit (SDK).

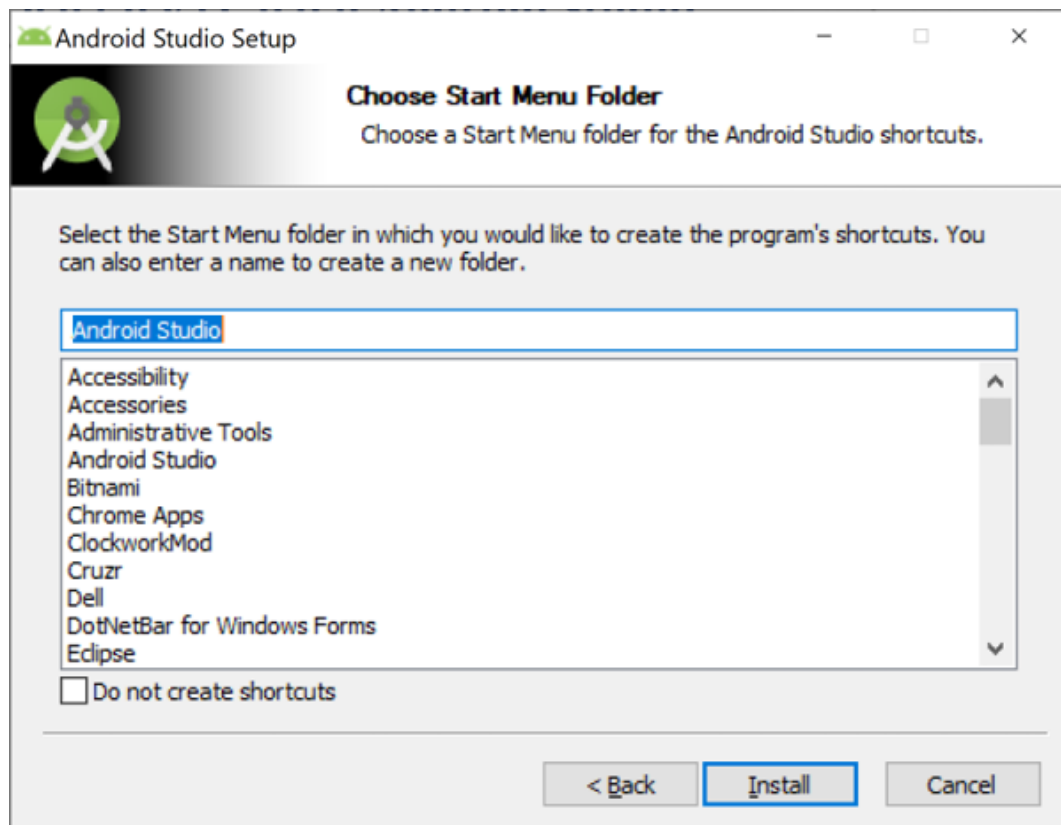
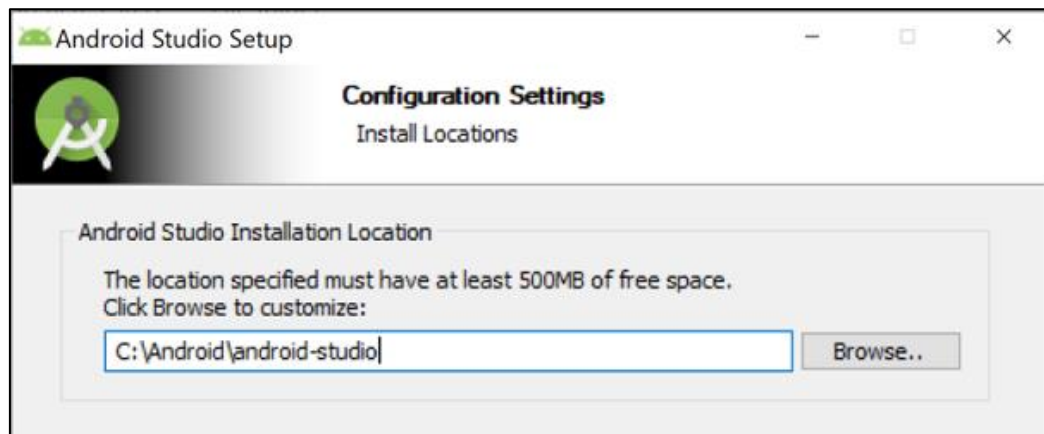
**Bước 2:** Tiếp theo nhấp đúp vào tệp cài mới tải ở trên về để cài đặt. chờ trong đây lát tiếp đến chọn next



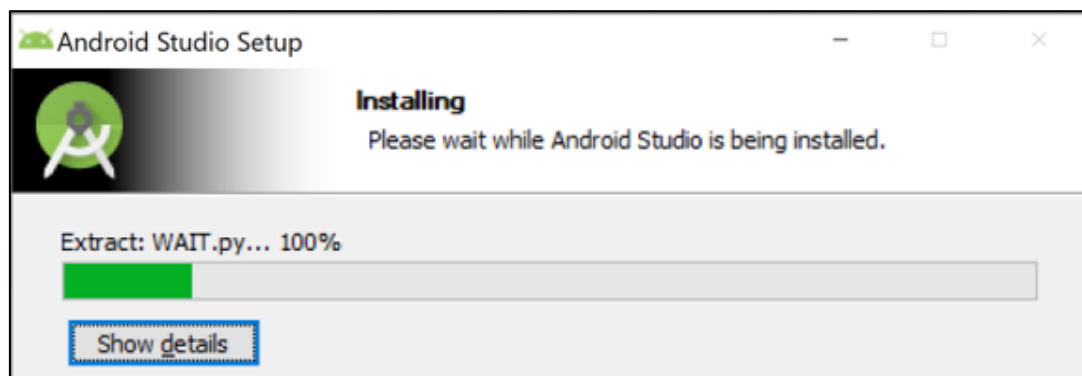
Bước 3: Để mặc định và chọn *Next* để tiếp tục



Bước 4: chọn thư mục cài đặt bộ công cụ phát triển. Ví dụ n đường dẫn C:\Android\android-studio như minh họa. Chọn Next.



Tiếp theo nhấn “Install” để bắt đầu cài đặt

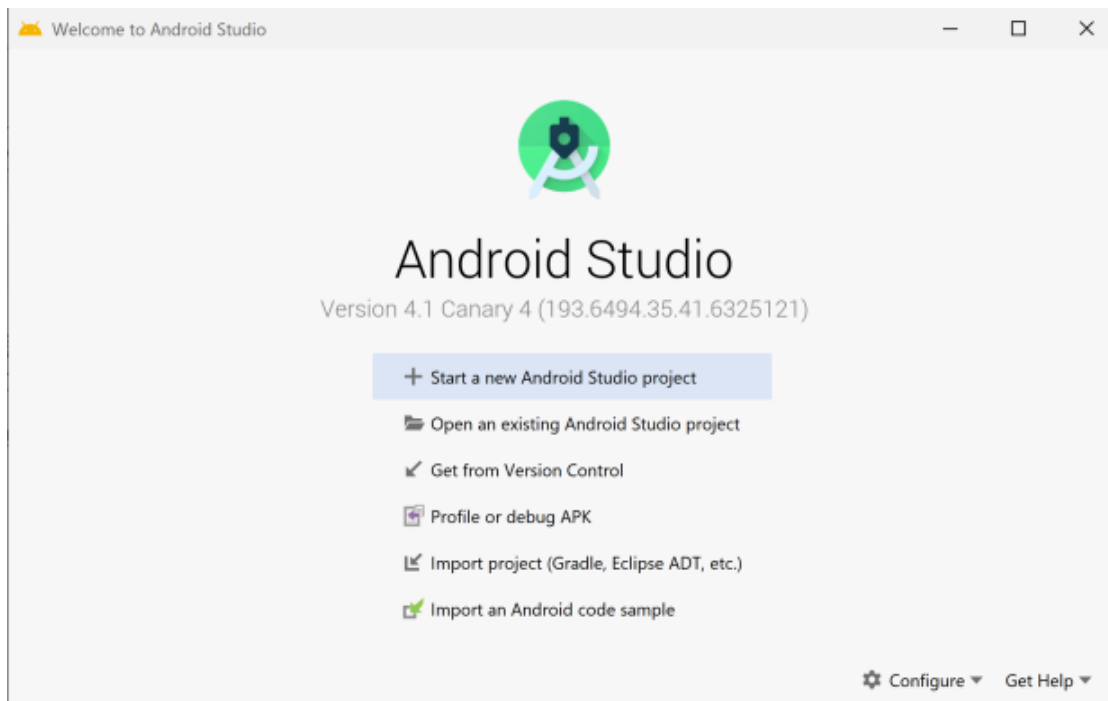


Sau khi cài đặt xong sẽ có thông báo hoàn thành; chọn finish để kết thúc.

## 2.2. Tạo project trong Android Studio

Trong phần này, nhóm tác giả minh họa cách tạo ra project “Hello World” . Bài toán minh họa ứng dụng khi chạy chương trình sẽ hiện thị dòng chữ “ Hello World“.

**Bước 1:** Khởi động phần mềm Android Studio :



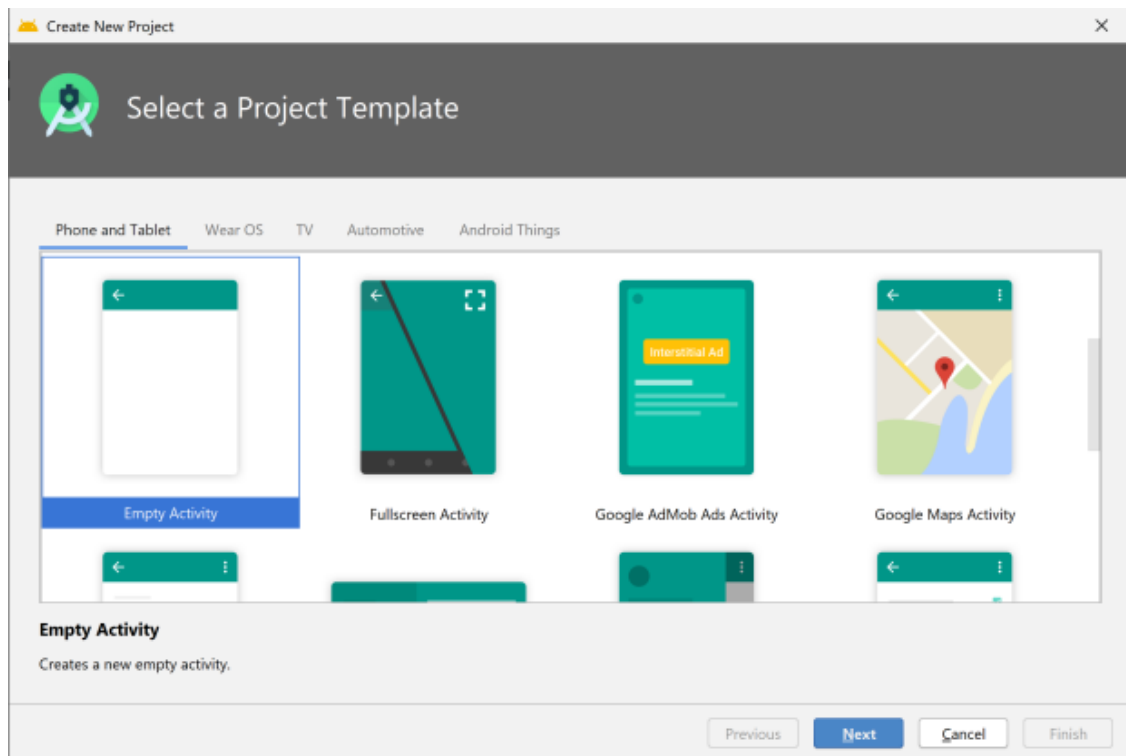
*Giải thích một số thuật ngữ trong tạo project ở Android Studio*

Tên chức năng	Ý nghĩa
Start a new Android Studio project	Tạo một dự án mới
Open an existing Android Studio project	Mở một dự án đã có, dự án này được viết bằng công cụ Android Studio
Get from Version Control	Lấy Code từ server về, chắc hạn như: Github, TFS...
Profile or debug APK	Kiểm tra đo lường, debug APK
Import Project (Gradle, eclipse ADT, etc.)	Mở một dự án Android được viết bằng Gradle, Eclipse....
Import an Android code sample	Mở một số code Android mẫu

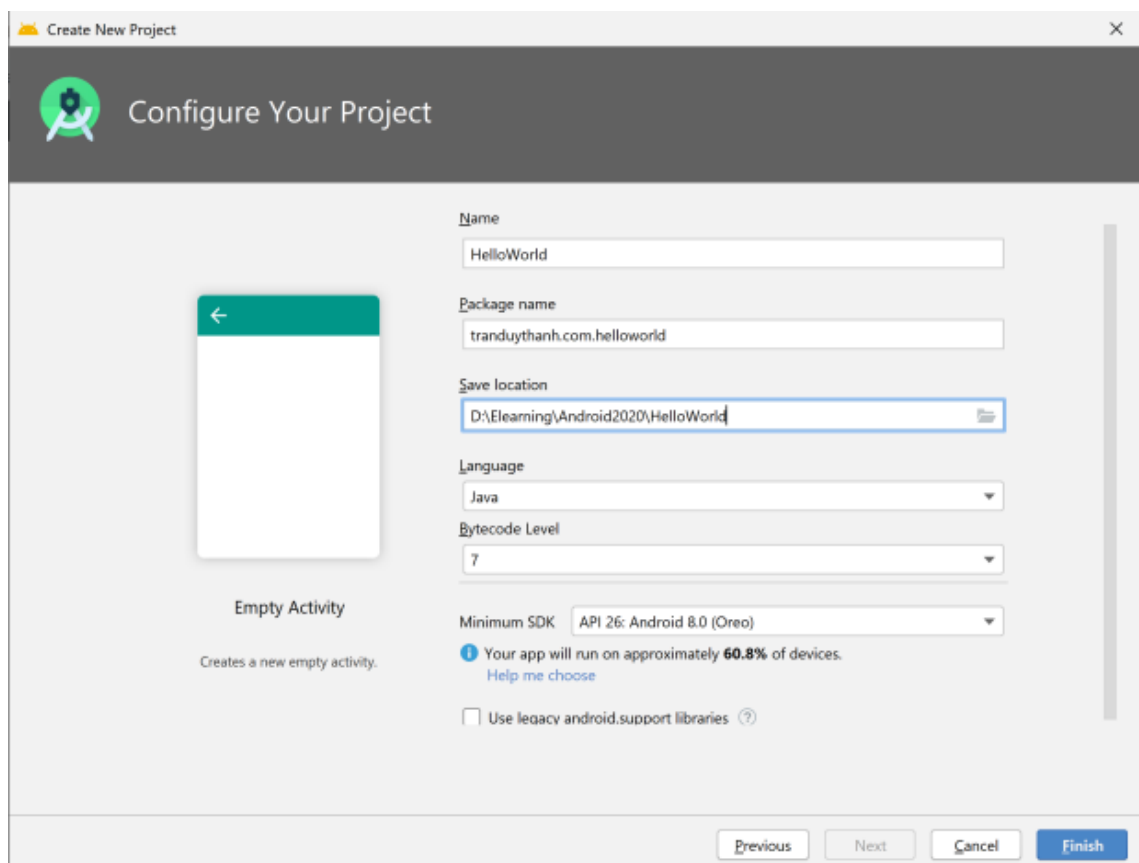
Từ màn hình khởi động phần mềm Android Studio, nhấn vào “**Start a new Android Studio project**“, bước này chúng ta chọn một giao diện mẫu phù hợp.

Chọn loại “Empty Activity” cho ứng dụng đầu tiên này





Sau khi chọn “Empty Activity”, nhấn “Next” để tiếp tục,



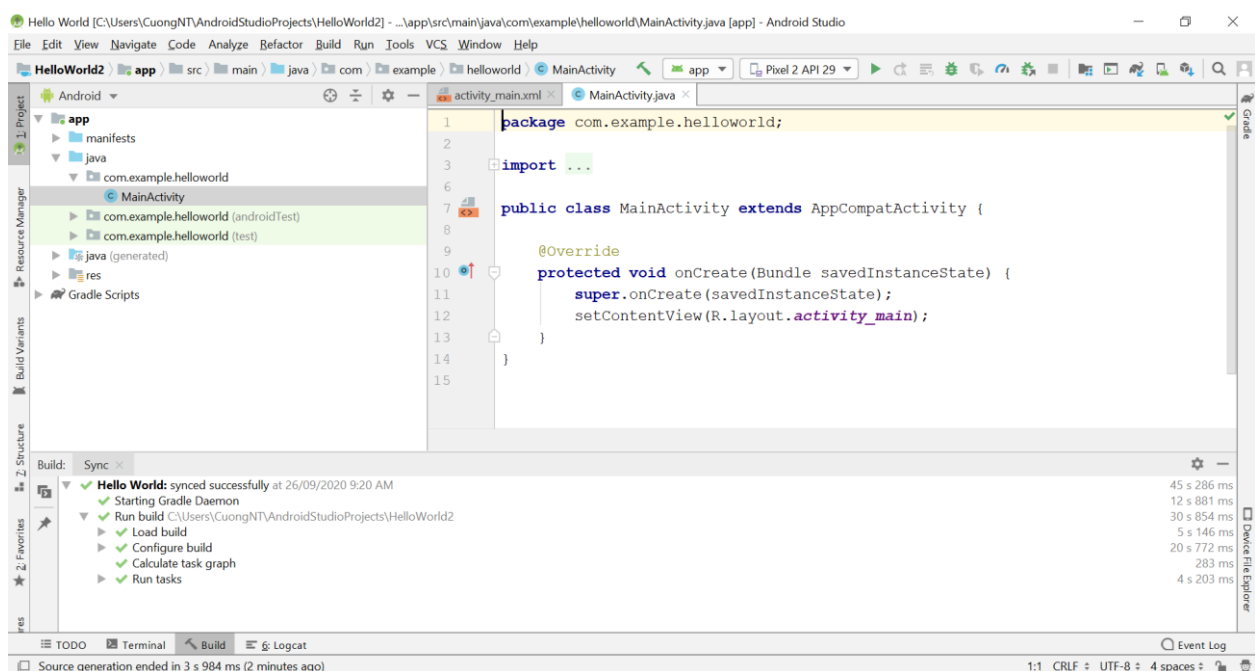
*Giải thích các thông tin trong phần cấu hình Project*

Tên chức năng	Ý nghĩa
Name	Tên Dự án, nên đặt tên có ý nghĩa.

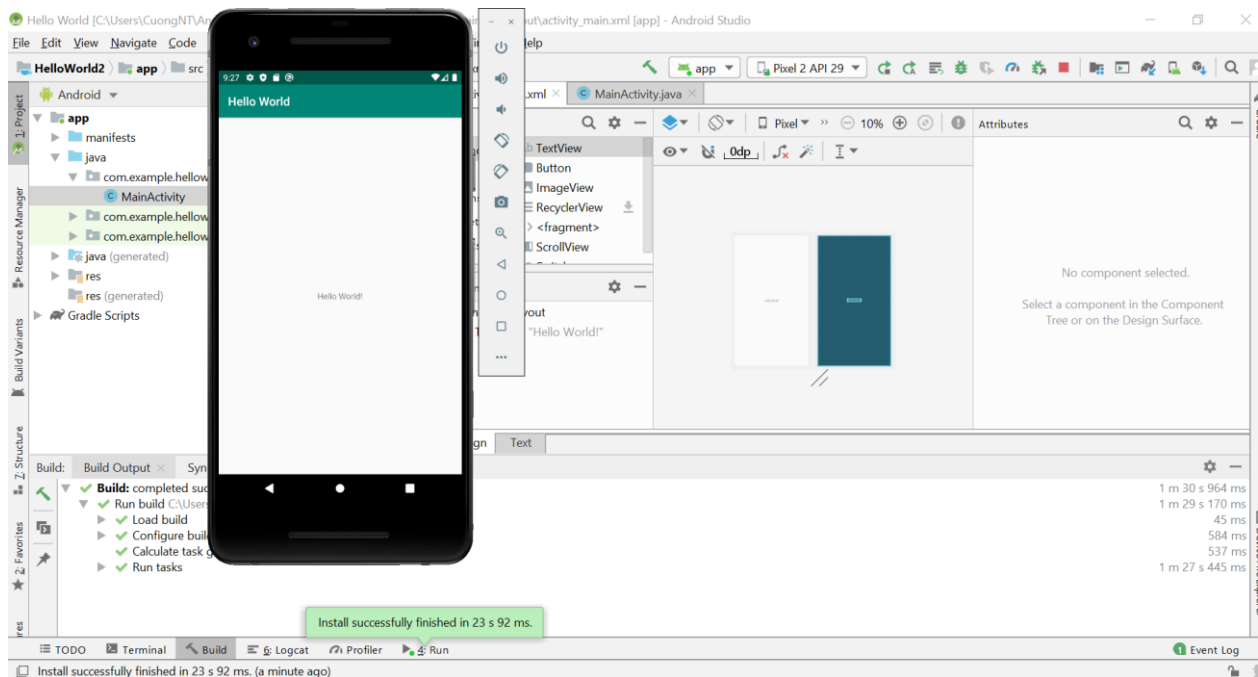


Package name	Package name rất quan trọng, nó được dùng để tải APK lên Google Play. Được xem như là ứng dụng. Nếu sản phẩm này viết cho ai thì nên đặt theo domain của người đó.
Save location	Nơi lưu trữ dự án
Language	Chọn Java hoặc Kotlin
Minimum SDK	Chọn SDK tối thiểu, gợi ý chọn CDK 26

Sau khi kết thúc cấu hình dự án, chương trình sẽ hiện thị ra màn hình lập trình với giao diện mặc định như hình sau:



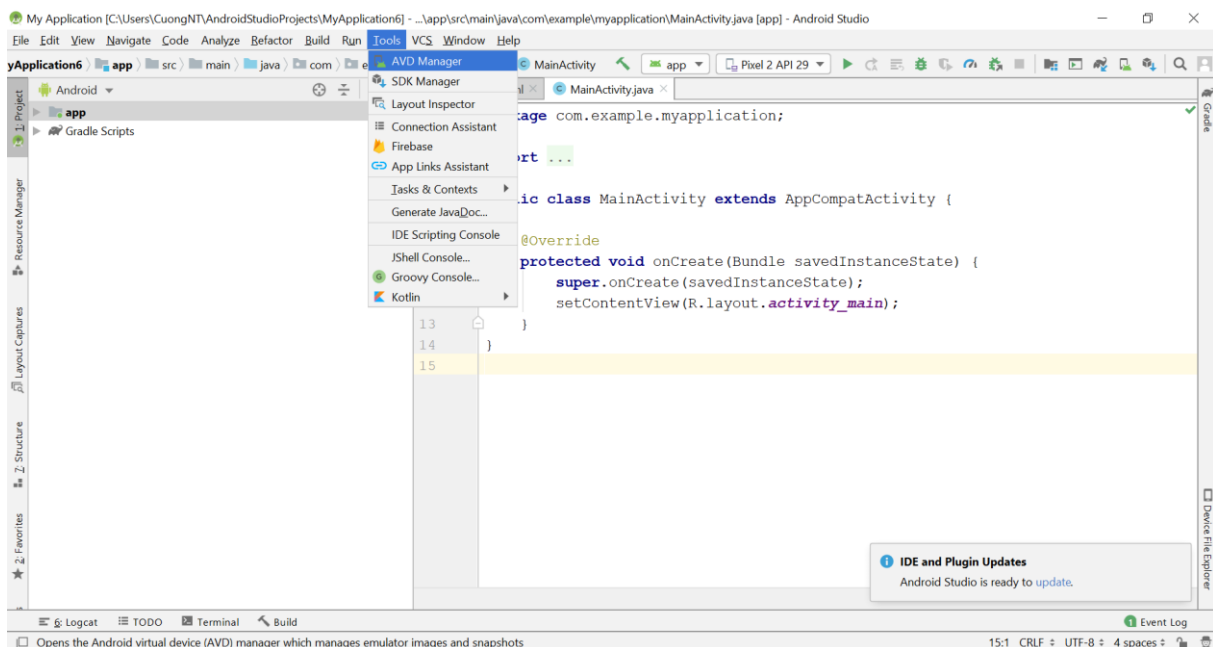
Chạy project, ta được kết quả sau :



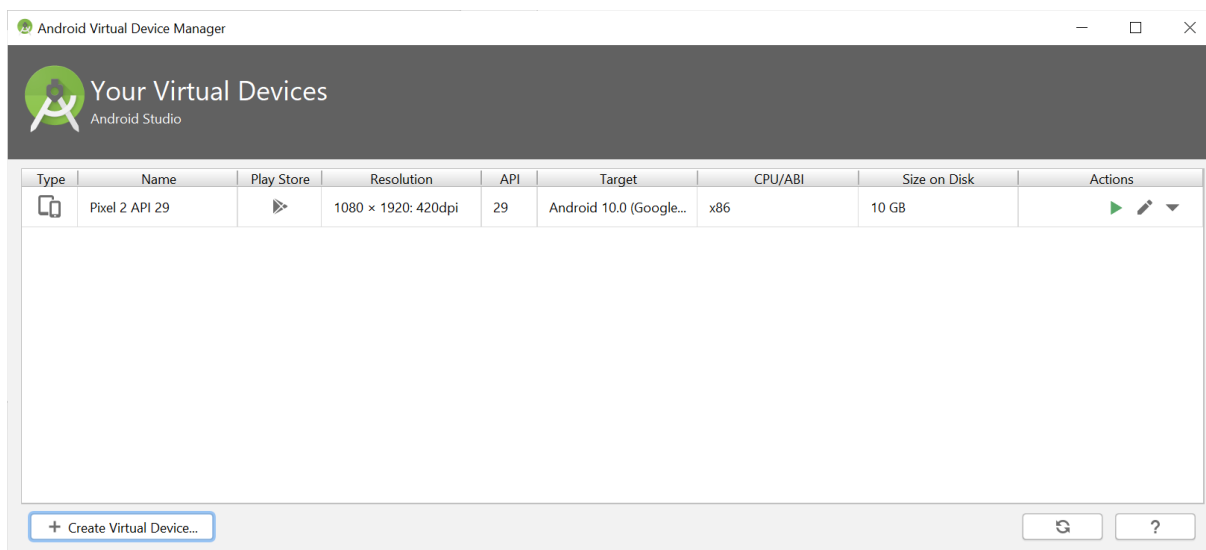
### 2.3. Hướng dẫn cấu hình máy ảo trong Android Studio

Máy ảo trong Android Studio (ADV -Android virtual Device) là trình giả lập cho phép công cụ tạo ra một thiết bị giống như điện thoại, máy tính bảng, tivi .. để lập trình viên có thể chạy thử.

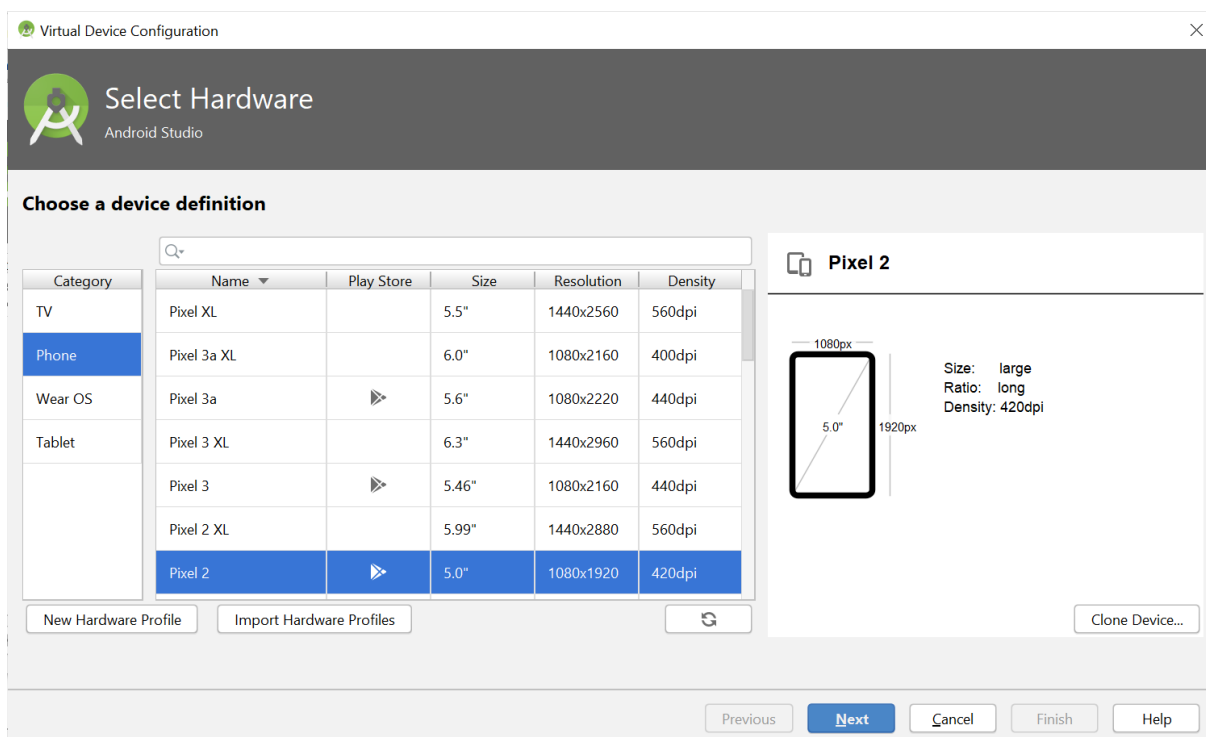
*Để tạo được máy ảo trong Android Studio, chúng ta vào Tools/ ADV Manager như hình :*



Sau đó, chúng ta *chọn* Create Virtual Device



Chọn thiết bị cần tạo máy ảo rồi bấm Next để tải về cài đặt, chờ Finish sẽ hoàn thành cài đặt máy ảo.



Để ứng dụng Android chạy được trên thiết bị thật, chúng ta cần thực hiện theo các bước sau :

- ✓ Cắm dây usb kết nối thiết bị ( điện thoại , máy tính bảng ) với máy tính
- ✓ Bật chế độ usb debugging
- ✓ Cài driver của thiết bị

Có thể tham khảo thêm hướng dẫn kết nối thiết bị qua wifi tại đường link sau :

<https://developer.android.com/studio/run/device>

)

### 3. Cấu trúc ứng dụng của Android

#### 3.1. Gradle là gì?

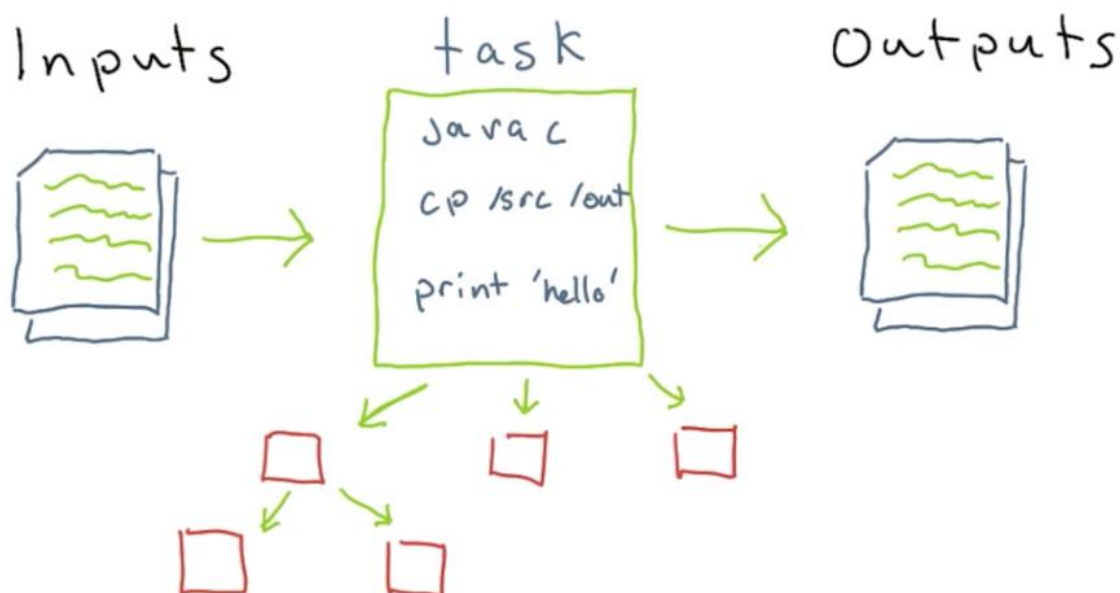
Tổng quan xây dựng một ứng dụng hay phần mềm thì đơn giản chỉ là biên dịch, đóng gói mã nguồn (compiling và packing source code) – build application.

Nhưng hiện nay, ngoài biên dịch, đóng gói, các ứng dụng còn tự động chạy các kiểm thử (test) sau khi biên dịch, đóng gói (build), tìm và lấy các đoạn lệnh theo các thư viện từ trên mạng (fetch code resources), sinh tài liệu (generate documentation), tạo ra nhiều bản đóng gói khác nhau (develop, release v.v.v), sau đó thì tự động công bố (publish app) lên kho ứng dụng (store) vvv. Vậy nên công việc build một app hiện tại không đơn giản như trước đây.

Google chọn Gradle thực hiện đóng gói và biên dịch (build system) cho Android Studio. Android Studio ủy thác toàn bộ quá trình này cho Gradle. Gradle đã giải quyết rất nhiều vấn đề cho Android developers:

- ✓ Làm sao để tự động hóa quá trình build và kiểm tra ứng dụng để đạt được năng suất cao?
- ✓ Làm sao để quản lý được các khai báo tham chiếu (dependencies) và các version của ứng dụng để cho phép người viết chương trình có thể đóng gói ứng dụng với nhiều phiên bản dễ dàng.

#### 3.2. Giới thiệu Tasks.



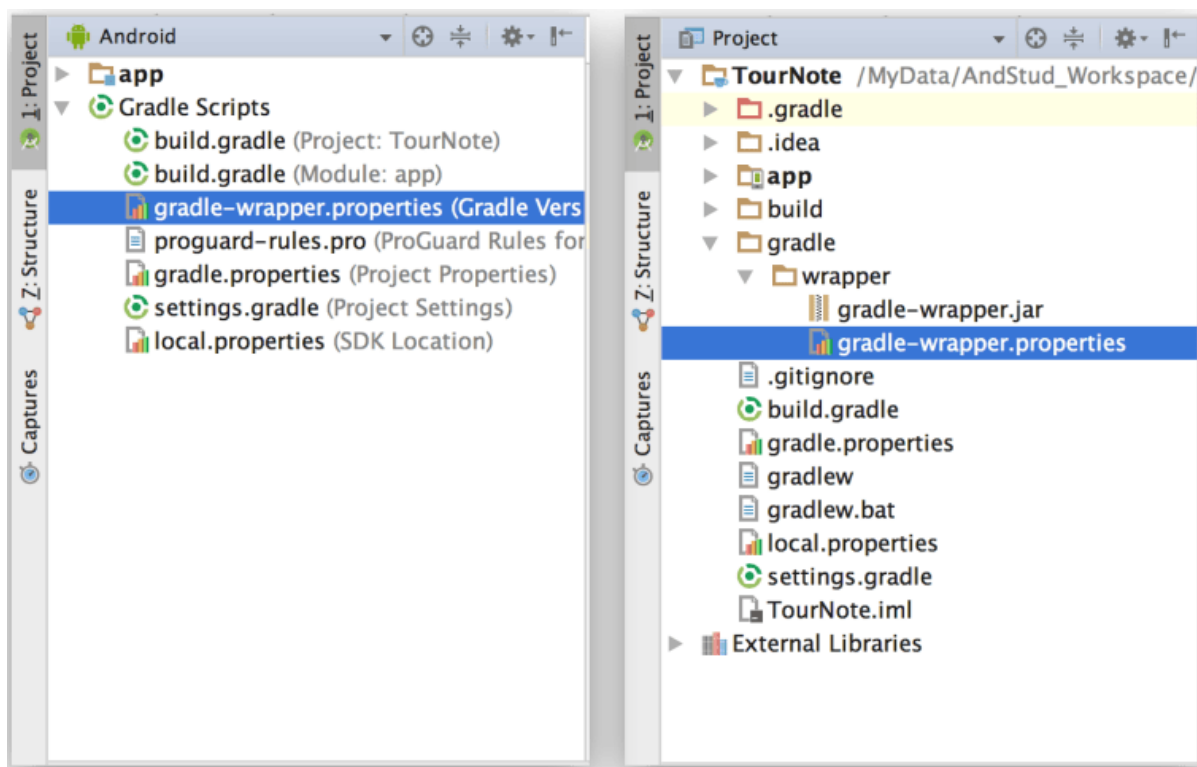
Một Task trong Gradle được hiểu là một đơn vị công việc mà Gradle có thể hiểu và triển khai được (run). Phần cốt lõi (core) của Task chính là Action. Ví dụ Một task còn có thể định

nghĩa inputs và outputs như một java method, ngoài ra 1 task còn có thể cấu hình để đọc và ghi file chẳng hạn.

Gradle sử dụng ngôn ngữ Groovy, có nhiều cú pháp giúp xây dựng các đoạn mã (scripts) tương đồng với ngôn ngữ tự nhiên hơn Java.

### 3.3. Làm Quen Với Gradle Wrapper

Trong Android Studio file Gradle Wrapper này nằm ở thư mục gradle/wrapper/. Tên file là gradle-wrapper.properties. Bạn có thể thấy file này như hình sau, hình trên máy bạn sẽ giống với hình bên trái hay bên phải là tùy vào cách view của cửa sổ này



nội dung của Gradle Wrapper rất ngắn nhưng rất quan trọng.

`distributionBase=GRADLE_USER_HOME`

`distributionPath=wrapper/dists`

`zipStoreBase=GRADLE_USER_HOME`

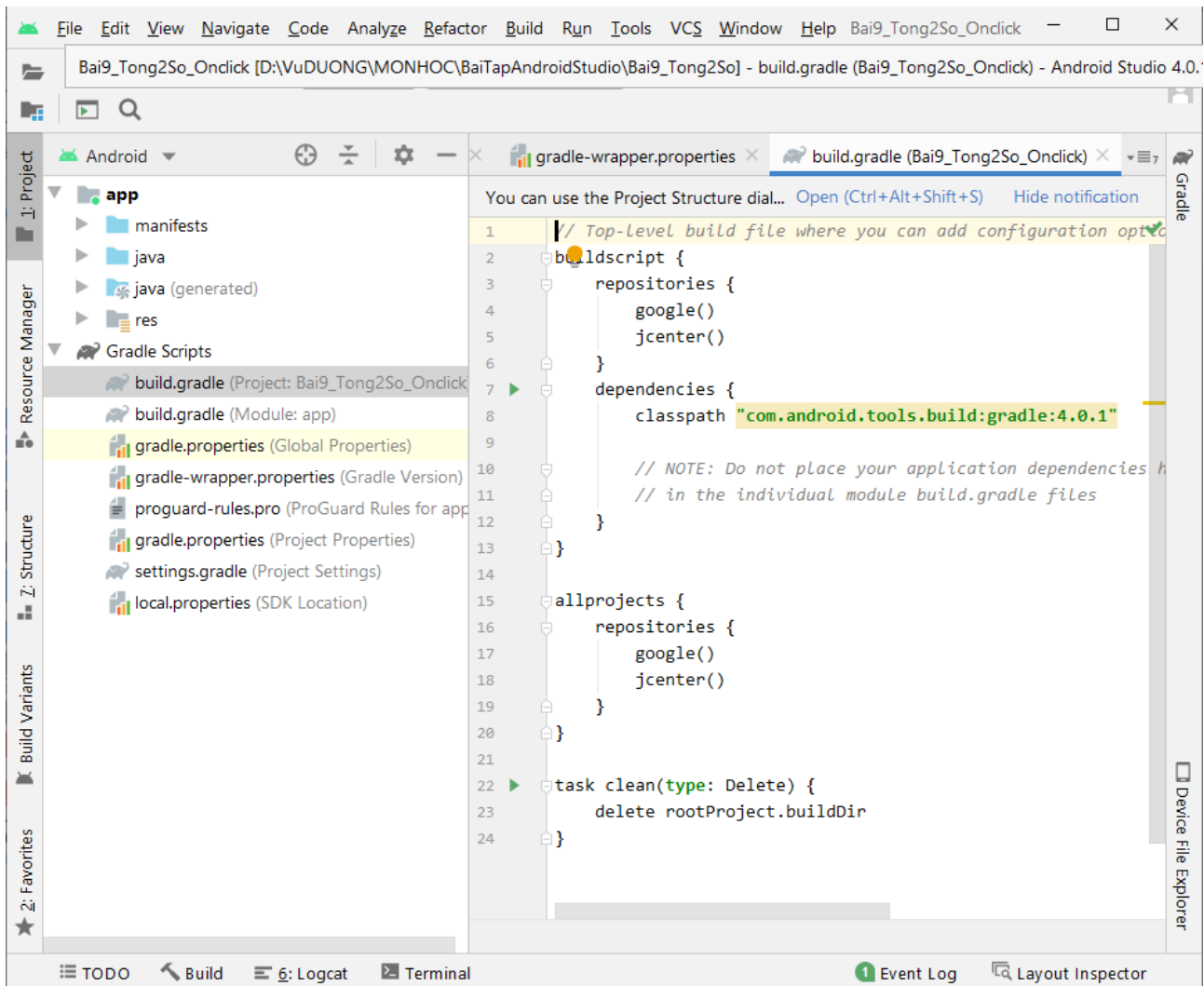
`zipStorePath=wrapper/dists`

`distributionUrl=https://services.gradle.org/distributions/gradle-6.1.1-all.zip`

File Gradle Wrapper chứa đường dẫn quyết định đến việc download Gradle ở đâu, version của Gradle. Dòng lệnh `distributionUrl` mô tả nội dung đó.

Thường thì khi tạo mới một project, Android Studio đã để sẵn đường dẫn này nên không cần cấu hình gì cả.

### 3.4. Các File Gradle



#### a. build.gradle ở cấp độ Project

Theo đường dẫn ở hình trên, build.gradle sẽ cấu hình cho tất cả các module bên trong project. Thông thường thì các project chỉ có một module mà thôi, vẫn có project có nhiều. – Mỗi một mục buildscript, allprojects, task clean trong file này được gọi là một khối.

##### ➤ Khối buildscript

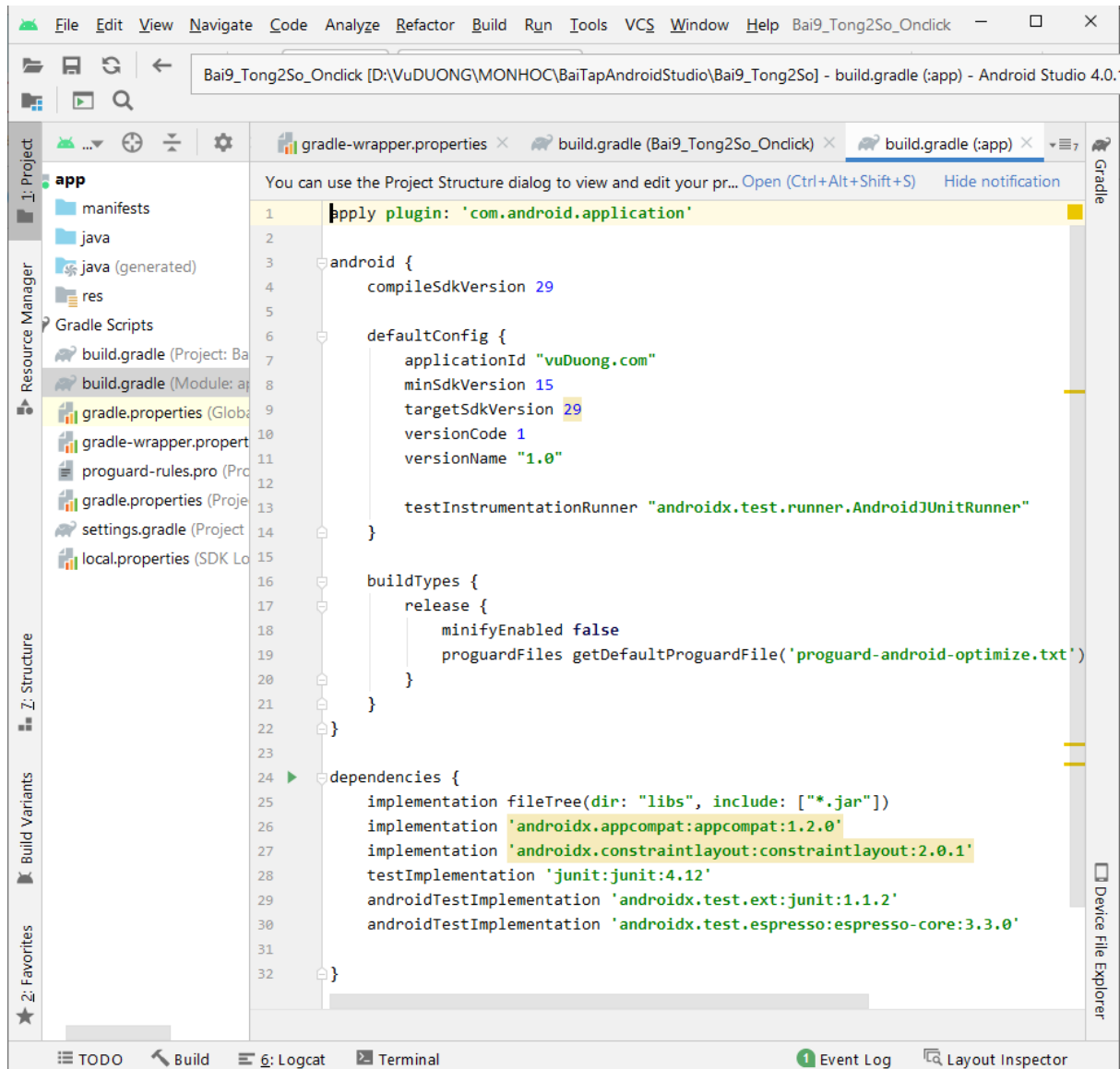
Chỉ ra vị trí download phiên bản gradle. Khối khai báo kho chứa (repositories) là ở jcenter() (hàm jcenter() này giúp định nghĩa thông tin về repository cho Maven Central, nơi thu thập các dependencies mã nguồn mở). Trong khối buildscript này còn có khối con dependencies là nơi liệt kê các thư viện dùng chung cho project, như hiện tại bạn thấy nó khai báo sẽ dùng một thư viện gradle, với thông tin về đường dẫn và version của gradle cần down.

##### ➤ Khối allprojects

Là nơi định nghĩa các thiết lập cho tất cả project. Trong trường hợp này đang định nghĩa jcenter() là repositories để tìm tất cả các thư viện cần thiết cho các module thuộc project hiện tại.

### b. build.gradle ở cấp độ Module

Đây là file này sẽ cấu hình cho từng module trong project.



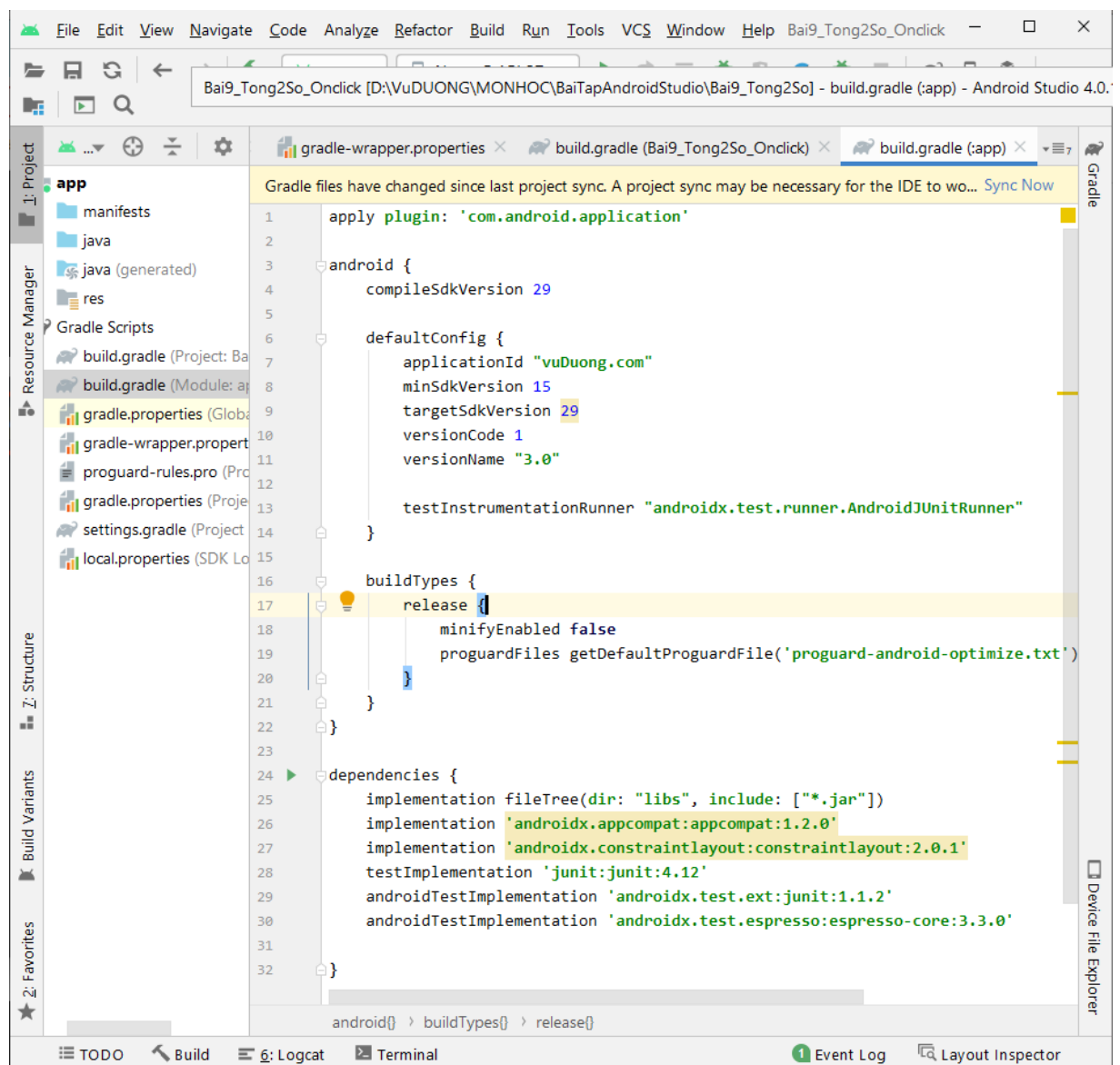
#### ➤ Khối android

Khối này định nghĩa rất nhiều cấu hình cho project. Khối android{} hoạt động là vì khi lập trình đã khai báo các thông số ứng dụng và tại khối này ta yêu cầu Gradle áp dụng plugin Android đã được khai báo trước đó

- compileSdkVersion: version của Android chỉ định để project phát hành. Cần giải thích thêm minSdkVersion là thông số version Android nhỏ nhất mà project của bạn hỗ trợ, targetSdkVersion là version Android mới nhất mà project của bạn hỗ trợ.



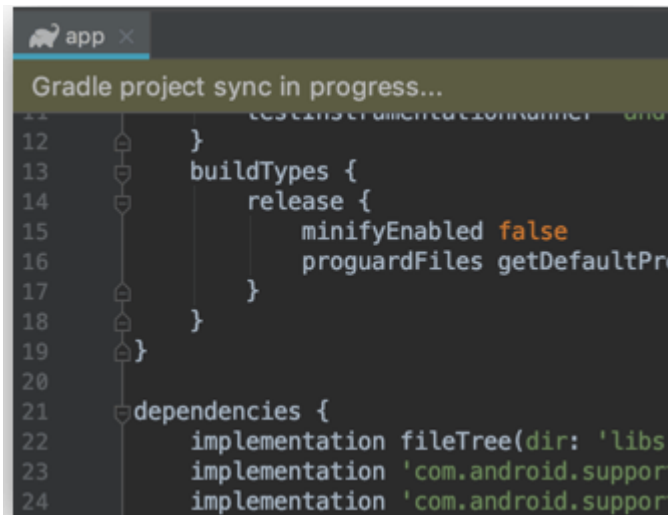
- Vậy khi biên dịch và đóng gói, project của bạn vẫn phải dựa vào một version Android cụ thể để nó còn biết gắn các thư viện hỗ trợ vào, chính vì vậy mà `compileSdkVersion` ra đời. Theo mặc định thì giá trị này sẽ được thiết lập là version Android lớn nhất (giống với version ở `targetSdkVersion`).
- Tuy nhiên người lập trình hoàn toàn có thể chỉ định một version thấp hơn version của target, nhưng compile với một version lớn nhất sẽ cho phép ứng dụng của bạn có được các thư viện Android mới nhất, như giao diện của Android 12.0 chẳng hạn, giúp tạo trải nghiệm tốt nhất cho user.
- Với mỗi thay đổi trong file gradle thì dự án đều được đồng bộ lại. Ví dụ như hình sau, tay đổi version name từ 1.0 thành 3.0. màn hình xuất hiện nhắc đồng bộ hệ thống.



Chọn Sync Now ở bên phải vạch vàng phía trên và đợi trong giây lát. Trong lúc đang đồng bộ hệ thống cũng có thông báo trạng thái.

➤ Khởi dependencies

Khởi dependencies cũng giống như khởi dependencies dùng trong buildscript ở file build.gradle cấp độ project đã nói ở trên đây. Cả hai đều dùng để định nghĩa ra các thư viện. Chúng khác nhau ở chỗ một định nghĩa thư viện dùng chung cho project hay cho từng module. Đây là nơi bạn sẽ thêm tất cả các thư viện của bên thứ ba như là phụ thuộc Gradle.

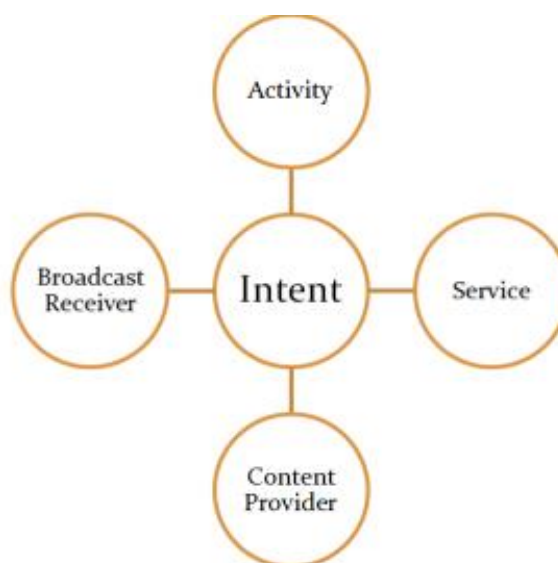


c. Gradle Daemon

Thông thường Gradle sẽ tốn nhiều thời gian cho việc khởi động và Gradle Daemon - là một quá trình chạy ngầm giúp người lập trình biên dịch- đóng gói (build) nhanh hơn. Đối với Gradle version < 3.0 thì mặc định thì Android Studio sẽ sử dụng Gradle Daemon nhưng nếu. Kể từ phiên bản 3.0 thì Gradle Daemon luôn.

## 4. Activity trong Android

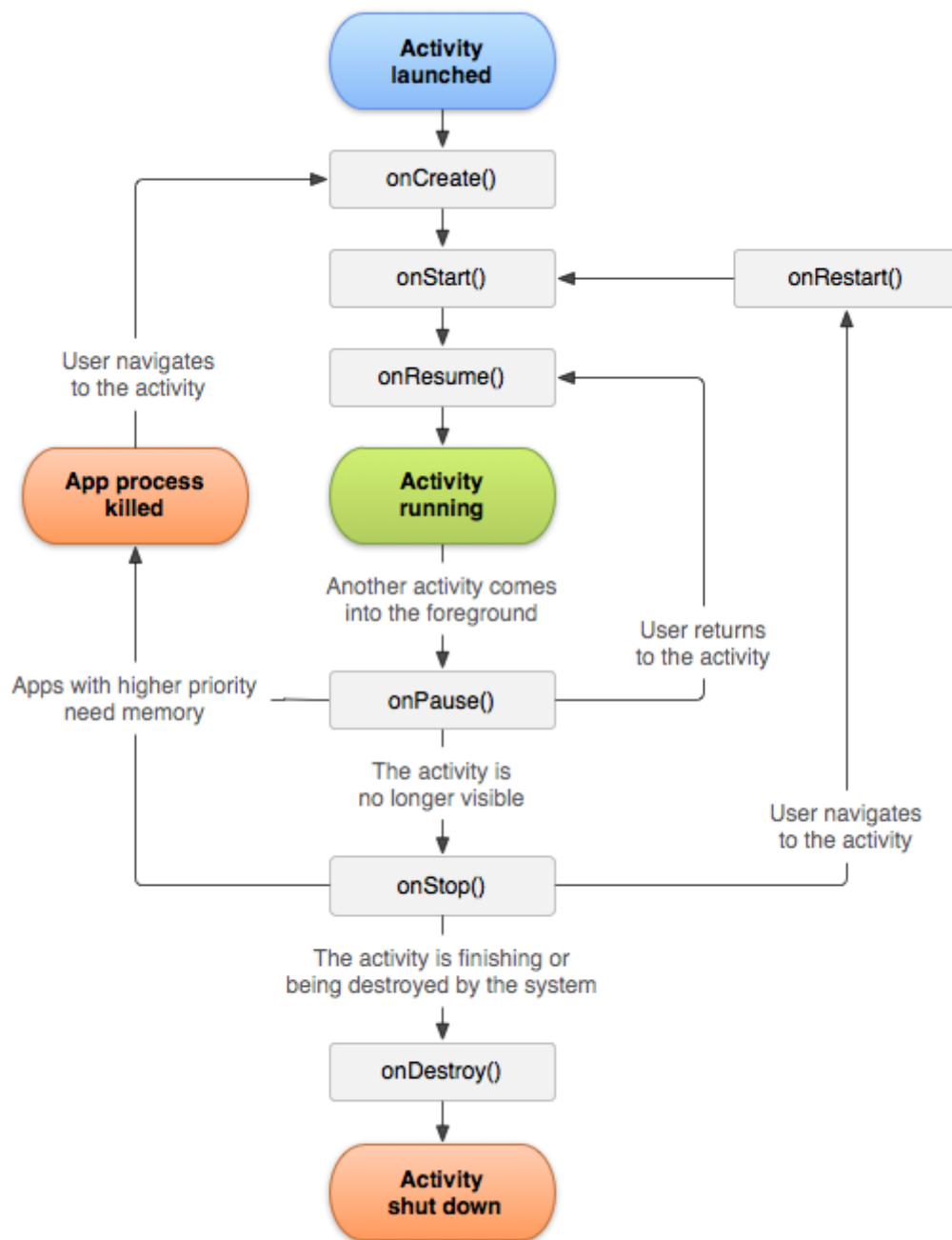
### 4.1. Khái niệm Activity



Lớp Activity là thành phần quan trọng nhất của ứng dụng Android, cách mà chúng hoạt động tạo thành nền tảng cơ bản của mô hình lập trình ứng dụng. Android khởi chạy một ứng dụng thông thường bằng kích hoạt một Activity tương ứng với vòng đời cụ thể của nó trong quá trình hoạt động.

Thường một Activity cung cấp một cửa sổ, ở đó ứng dụng sẽ dựng các thành phần UI (User Interface - giao diện người dùng). Mặc định cửa sổ này có thể lấp đầy màn hình thiết bị, nhỏ hơn hoặc nổi phía trên các cửa sổ khác. Hầu hết các ứng dụng đều sử dụng nhiều màn hình khác nhau, có nghĩa nó sẽ phải có nhiều Activity khác nhau. Khi một Activity chỉ định là Main Activity, nó sẽ là màn hình đầu tiên khi khởi chạy ứng dụng. Activity này có thể gọi và kích hoạt một Activity khác

## 4.2. Vòng đời Activity



Sơ đồ bắt đầu từ khi Activity launched, tức là khi Activity được kích hoạt, và được hệ thống đẩy vào BackStack. Sau khi kích hoạt, lần lượt các callback `onCreate()`, `onStart()`, `onResume()` sẽ được hệ thống gọi đến.

Sau khi gọi đến các callback trên, thì Activity mới chính thức được xem là đang chạy (Activity running).

Lúc này, nếu có bất kỳ Activity nào khác chiếm quyền hiển thị, thì Activity hiện tại sẽ rơi vào trạng thái `onPause()`. Nếu sự hiển thị của Activity khác làm cho Activity mà chúng ta đang nói đến không còn nhìn thấy nữa thì `onStop()` sẽ được gọi ngay sau

đó Nếu Activity đã vào onPause() rồi, tức là đang bị Activity khác đè lên, mà người dùng sau đó quay về lại Activity cũ, thì onResume() được gọi. Còn nếu Activity đã vào onStop() rồi, mà người dùng quay về lại Activity cũ thì onRestart() được gọi. Trong cả hai trường hợp Activity rơi vào onPause() hoặc onStop(), nó sẽ rất dễ bị hệ thống thu hồi (tức là bị hủy) để giải phóng tài nguyên, khi này nếu quay lại Activity cũ, onCreate() sẽ được gọi chứ không phải onResume() hay onRestart(). Và cuối cùng, nếu một Activity bị hủy một cách có chủ đích, chẳng hạn như người dùng nhấn nút Back ở System Bar, hay hàm finish() được gọi,... thì onDestroy() sẽ được kích hoạt và Activity kết thúc vòng đời của nó.

### **Các trạng thái chính trong vòng đời activity**

- **Running**

Khi Activity được kích hoạt, và được hệ thống đẩy vào BackStack, nó sẽ bước vào trạng thái active. Với trạng thái active, người dùng hoàn toàn có thể nhìn thấy và tương tác với Activity của ứng dụng.

- **Pause**

Trạng thái này khá đặc biệt. Trạng thái tạm dừng. Như bạn đã làm quen trên kia, trạng thái này xảy ra khi mà Activity của bạn vẫn đang chạy, người dùng vẫn nhìn thấy, nhưng Activity khi này lại bị che một phần bởi một thành phần nào đó. Chẳng hạn như khi bị một dialog đè lên. Cái sự che Activity này không phải hoàn toàn. Chính vì vậy mà Activity đó tuy được người dùng nhìn thấy nhưng không tương tác được.

- **Stop**

Trạng thái này khá giống với trạng thái tạm dừng trên kia. Nhưng khi này Activity bị che khuất hoàn toàn bởi một thành phần giao diện nào đó, hoặc bởi một ứng dụng khác. Và tất nhiên lúc này người dùng không thể nhìn thấy Activity của bạn được nữa. Hành động mà khi người dùng nhấn nút Home ở System Bar để đưa ứng dụng của bạn về background, cũng khiến Activity đang hiển thị trong ứng dụng rơi vào trạng thái dừng này.

- **Dead**

Nếu Activity được lấy ra khỏi BackStack, chúng sẽ bị hủy và rơi vào trạng thái này. Trường hợp này xảy ra khi user nhấn nút Back ở System Bar để thoát một Activity. Hoặc lời gọi hàm finish() từ một Activity để “kill chính nó”. Cũng có khi ứng dụng ở trạng thái background quá lâu, hệ thống có thể sẽ thu hồi tài nguyên bằng cách dừng hẳn các Activity trong ứng dụng, làm cho tất cả các Activity đều vào trạng thái này. Khi vào trạng thái dead, Activity sẽ kết thúc vòng đời của nó.

Những ý trên giúp bạn nắm được tổng quan các trạng thái mà một Activity có thể trải qua.