

TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



GIÁO TRÌNH

THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG

Hà Nội, 2.2025

MỤC LỤC

CHƯƠNG 1.	Làm quen.....	4
Bài 1)	Tạo ứng dụng đầu tiên	4
1.1)	Android Studio và Hello World	4
1.2)	Giao diện người dùng tương tác đầu tiên	21
1.3)	Trình chỉnh sửa bố cục	21
1.4)	Văn bản và các chế độ cuộn	21
1.5)	Tài nguyên có sẵn.....	21
Bài 2)	Activities	21
2.1)	Activity và Intent	21
2.2)	Vòng đời của Activity và trạng thái	21
2.3)	Intent ngầm định.....	21
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ	21
3.1)	Trình gỡ lỗi	21
3.2)	Kiểm thử đơn vị.....	21
3.3)	Thư viện hỗ trợ.....	21
CHƯƠNG 2.	Trải nghiệm người dùng	22
Bài 1)	Tương tác người dùng	22
1.1)	Hình ảnh có thể chọn	22
1.2)	Các điều khiển nhập liệu	22
1.3)	Menu và bộ chọn	22
1.4)	Điều hướng người dùng	22
1.5)	RecyclerView	22
Bài 2)	Trải nghiệm người dùng thú vị.....	22
2.1)	Hình vẽ, định kiểu và chủ đề	22
2.2)	Thẻ và màu sắc	22

2.3)	Bố cục thích ứng.....	22
Bài 3)	Kiểm thử giao diện người dùng.....	22
3.1)	Espresso cho việc kiểm tra UI	22
CHƯƠNG 3. Làm việc trong nền		22
Bài 1)	Các tác vụ nền.....	22
1.1)	AsyncTask	22
1.2)	AsyncTask và AsyncTaskLoader	22
1.3)	Broadcast receivers	22
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	22
2.1)	Thông báo	22
2.2)	Trình quản lý cảnh báo	22
2.3)	JobScheduler.....	22
CHƯƠNG 4. Lưu dữ liệu người dùng		23
Bài 1)	Tùy chọn và cài đặt.....	23
1.1)	Shared preferences.....	23
1.2)	Cài đặt ứng dụng.....	23
Bài 2)	Lưu trữ dữ liệu với Room	23
2.1)	Room, LiveData và ViewModel.....	23
2.2)	Room, LiveData và ViewModel.....	23
3.1)	Trình gỡ lỗi	

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

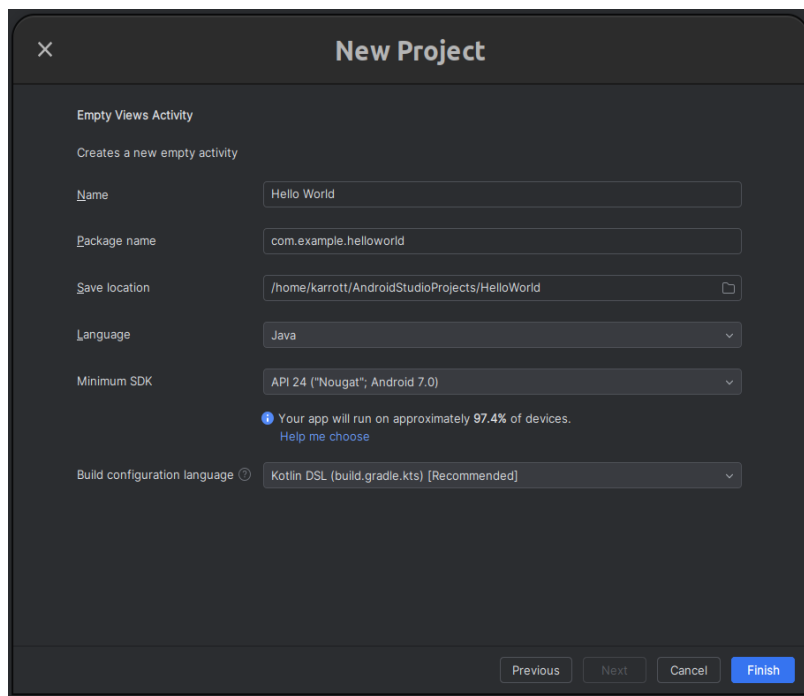
Giới thiệu

Trong bài thực hành này, bạn sẽ học được cách cài đặt Android Studio, môi trường phát triển Android. Bạn cũng sẽ tạo và chạy ứng dụng Android đầu tiên của mình, **Hello World**, trên trình giả lập và thiết bị thực.

Những gì Bạn nên biết

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng hướng đối tượng bằng IDE (integrated development environment - Môi trường phát triển tích hợp) như **Android Studio**.
- Chứng minh rằng bạn cần có 1-3 năm kinh nghiệm trong lập trình hướng đối tượng trong đó có một phần tập trung vào ngôn ngữ lập trình java. (Bài thực hành này sẽ không giải thích về chương trình hướng đối tượng hoặc ngôn ngữ java)



Những gì bạn sẽ cần

- Một máy tính chạy Windows hoặc Linux hoặc Mac chạy macOS. Xem trang tải xuống Android Studio để biết các yêu cầu hệ thống mới nhất.
- Kết nối internet hoặc cách khác để tải và cài đặt phiên bản mới nhất của Android Studio và java lên máy tính của bạn.

Những gì bạn sẽ học

- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng các ứng dụng Android.
- Cách tạo dự án Android từ một mẫu.
- Cách thêm thông điệp log và ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

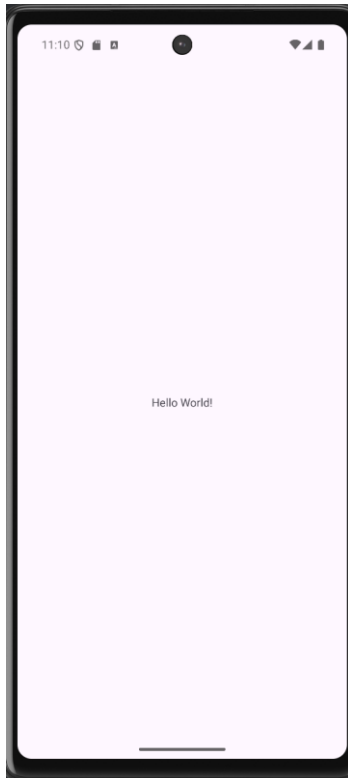
Những gì bạn sẽ làm:

- Cài đặt môi trường phát triển **Android Studio**.
- Tạo một trình giả lập (Thiết bị ảo) để chạy ứng dụng của bạn trên máy tính.
- Tạo và chạy ứng dụng **Hello World** bằng thiết bị ảo hoặc thiết bị thật.
- Khám phá bố cục dự án.
- Tạo và xem các thông điệp trên dự án của bạn.
- Khám phá tệp **AndroidManifest.xml**

Tổng quan về ứng dụng

Sau khi bạn cài đặt thành công Android Studio, bạn sẽ tạo, từ một mẫu, một dự án mới cho ứng dụng Hello World. Ứng dụng đơn giản này hiển thị chuỗi “Hello World” trên màn hình của thiết bị Android ảo hoặc vật lý.

Đây là ứng dụng hoàn thiện sẽ trông như thế nào:



Nhiệm vụ 1: Tải xuống Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp hoàn chỉnh (IDE) gồm trình chỉnh sửa mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ để phát triển, gỡ lỗi, thử nghiệm và hiệu suất giúp nhà phát triển phát triển ứng dụng nhanh hơn và dễ dàng hơn. Bạn có thể kiểm tra ứng dụng của mình với nhiều trình mô phỏng được cấu hình sẵn hoặc trên thiết bị di động của riêng bạn, Xây dựng các ứng dụng sản xuất và xuất bản trên cửa hàng Google Play.

Lưu ý: Android Studio liên tục được cải thiện. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, xem [Android Studio](#)

Android Studio có sẵn cho máy tính chạy Windows hoặc Linux và cho máy Mac chạy MacOS. OpenJDK (Bộ công cụ phát triển Java) mới nhất đi kèm với Android Studio.

Để thiết lập và chạy Android Studio, trước tiên hãy kiểm tra các yêu cầu hệ thống để đảm bảo rằng hệ thống đáp ứng chúng. Việc cài đặt tương tự cho tất cả các nền tảng. Mọi khác biệt đều được ghi chú bên dưới.

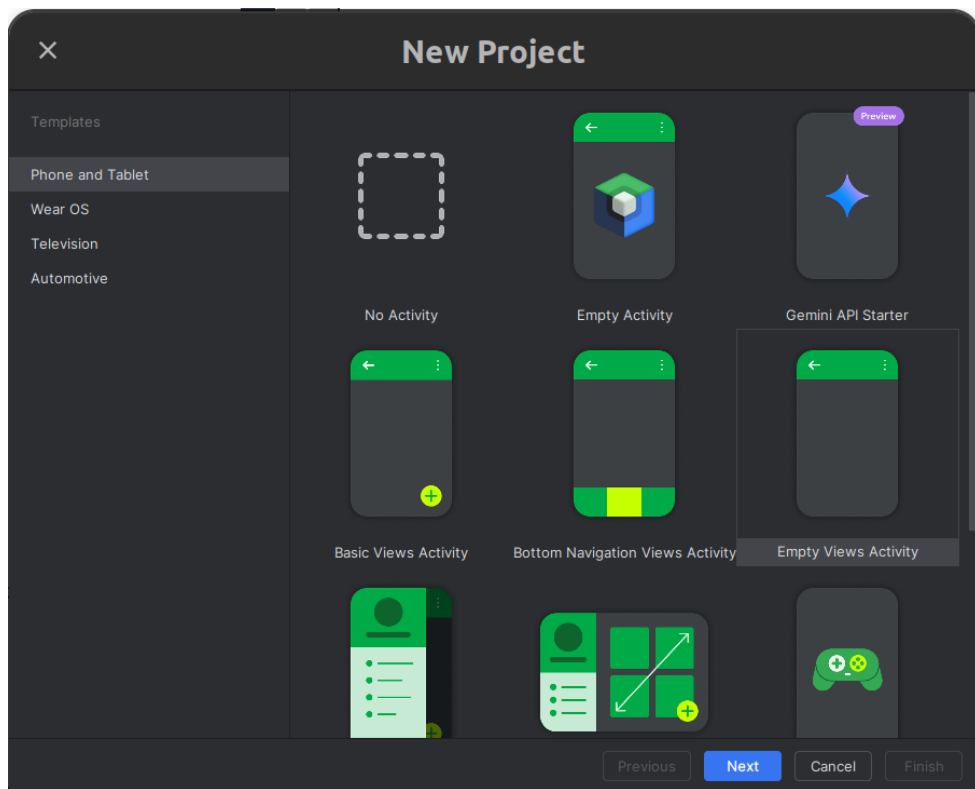
1. Điều hướng đến trang web dành cho [nhà phát triển Android](#) và làm theo hướng dẫn để tải xuống và [cài đặt Android Studio](#).
2. Chấp nhận các cấu hình mặc định cho tất cả các bước và đảm bảo rằng tất cả các thành phần đều được chọn để cài đặt.
3. Sau khi hoàn tất cài đặt, trình hướng dẫn cài đặt sẽ tải xuống và cài đặt thêm một số các thành phần bao gồm SDK Android. Hãy kiên nhẫn, việc này có thể mất một chút thời gian tùy thuộc vào tốc độ internet của bạn và một số bước có thể có vẻ dư thừa.
4. Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động và bạn đã sẵn sàng tạo dự án đầu tiên.

Nhiệm vụ 2: Tạo ứng dụng Hello World

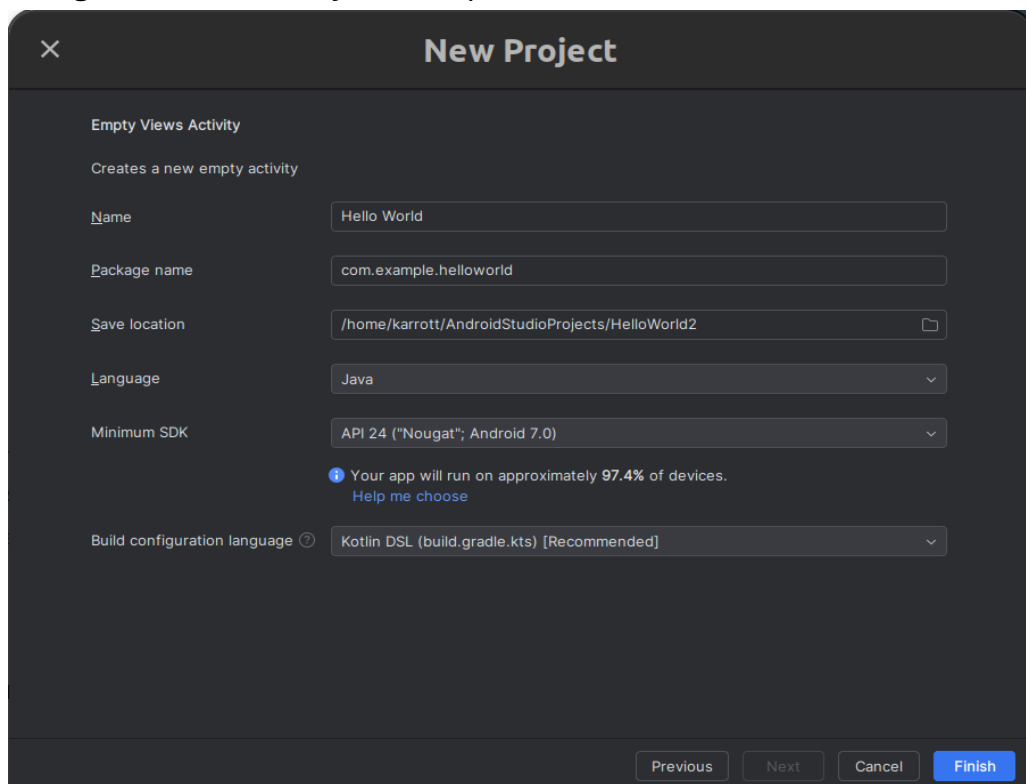
Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị “Hello World” để xác minh rằng Android Studio đang được cài đặt đúng cách và tìm hiểu những kiến thức cơ bản về phát triển ứng dụng với Android Studio.

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu nó chưa được mở.
2. Trong cửa sổ **Welcome to Android Studio** chính, nhấp vào **New Project**.
3. Trong cửa sổ **New Project** bạn có thể chọn loại dự án bạn muốn tạo từ các danh mục yếu tố hình thức thiết bị, được hiển thị trong ngăn mẫu (Templates) và cửa sổ chọn Activity xuất hiện. Activity là một thành phần tập chung vào một tác vụ duy nhất mà người dùng có thể thực hiện. Đây là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một Activity thường đi kèm với một tập bố cục, xác định cách các thành phần giao diện người dùng hiển thị trên màn hình. Android Studio cung cấp các mẫu Activity để giúp bạn bắt đầu nhanh chóng. Đối với dự án **Hello World** hãy chọn **Empty Views Activity** như hình minh họa bên dưới, sau đó nhấp vào **Next**.



4. Trong cửa sổ **New Project**, nhập **Hello World** vào mục **Name**.



5. Xác minh rằng vị trí dự án (**Save location**) mặc định là nơi bạn muốn lưu trữ ứng dụng **Hello World** của mình và các dự án Android Studio khác hoặc thay đổi nó vào mục ưa thích của bạn.

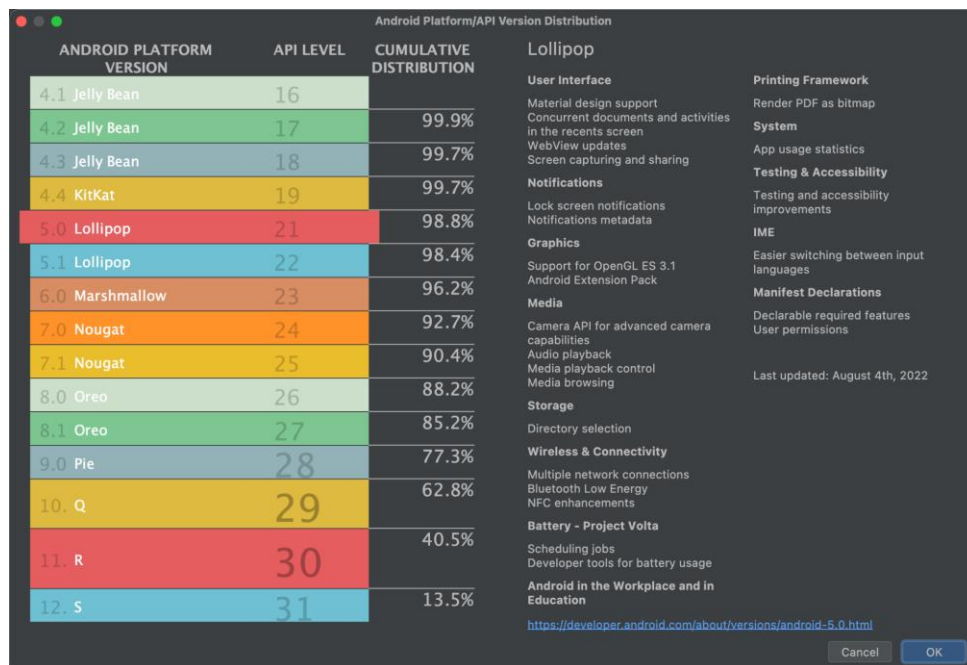
6. Chấp nhận giá trị mặc định **android.example.com** cho **Company domain** hoặc tạo một tên miền duy nhất.

Nếu bạn không có kế hoạch xuất bản ứng dụng của mình, bạn có thể chấp nhận mặc định. Xin lưu ý rằng việc thay đổi tên gói của ứng dụng của bạn sau này là công việc thêm.

7. Ở mục **Language** bạn có thể chọn **Java** hoặc **Kotlin**, Ở dự án này chúng ta chọn ngôn ngữ java.

8. Chọn mức API tối thiểu (**Minimum API level**) mà bạn muốn ứng dụng của bạn hỗ trợ. Khi bạn chọn API thấp hơn, ứng dụng của bạn không thể sử dụng nhiều API Android hiện đại. Tuy nhiên, ứng dụng sẽ chạy được trên nhiều thiết bị Android hơn. Ngược lại, khi chọn API cấp cao hơn, ứng dụng có thể sử dụng nhiều tính năng mới nhưng sẽ chạy trên ít thiết bị hơn.

Nếu bạn muốn xem thêm dữ liệu để hỗ trợ quyết định, hãy nhấp vào Help me choose. Thao tác này sẽ hiển thị một hộp thoại cho thấy phân bố tích lũy của mức API mà bạn đã chọn và giúp bạn đánh giá tác động của việc sử dụng các mức API tối thiểu khác nhau



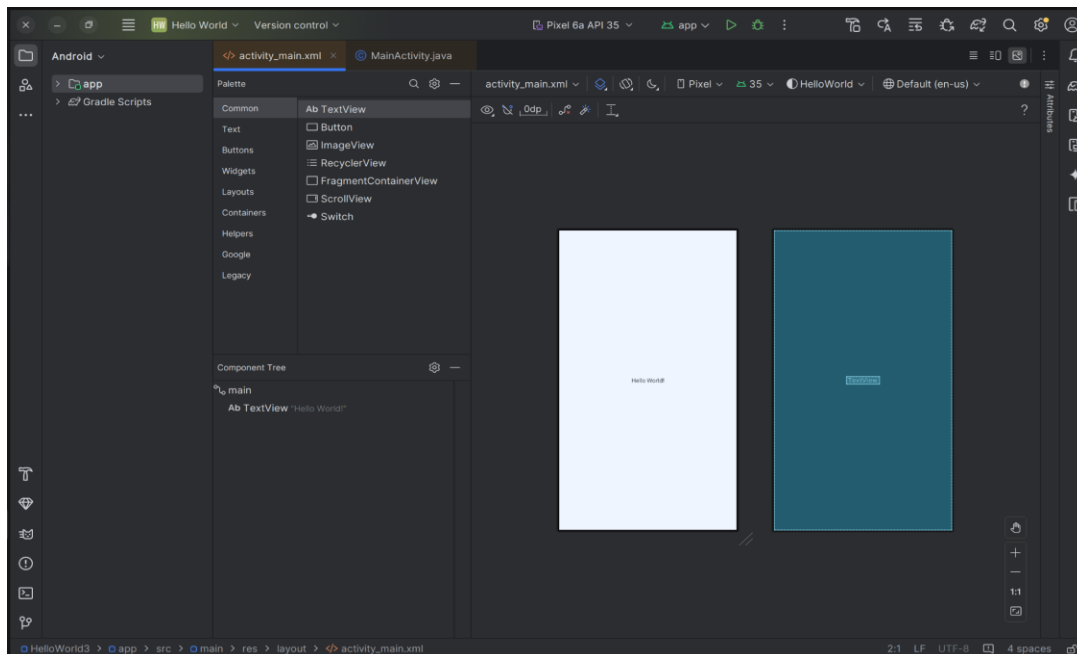
- Ở mục **Build configuration language** có thể chọn ngôn ngữ **Kotlin DSL** hoặc **Groovy DSL**. Chúng ta nên chọn **Kotlin DSL** vì Kotlin được ưu tiên hơn Groovy để viết các tập lệnh Gradle vì Kotlin dễ đọc hơn và cung cấp hỗ trợ IDE và kiểm tra thời gian biên dịch tốt hơn.
- nhấn vào **Finish** để hoàn thành tạo dự án.

Android Studio tạo một thư mục cho các dự án của bạn và xây dựng dự án bằng Gradle (quá trình này có thể mất vài phút).

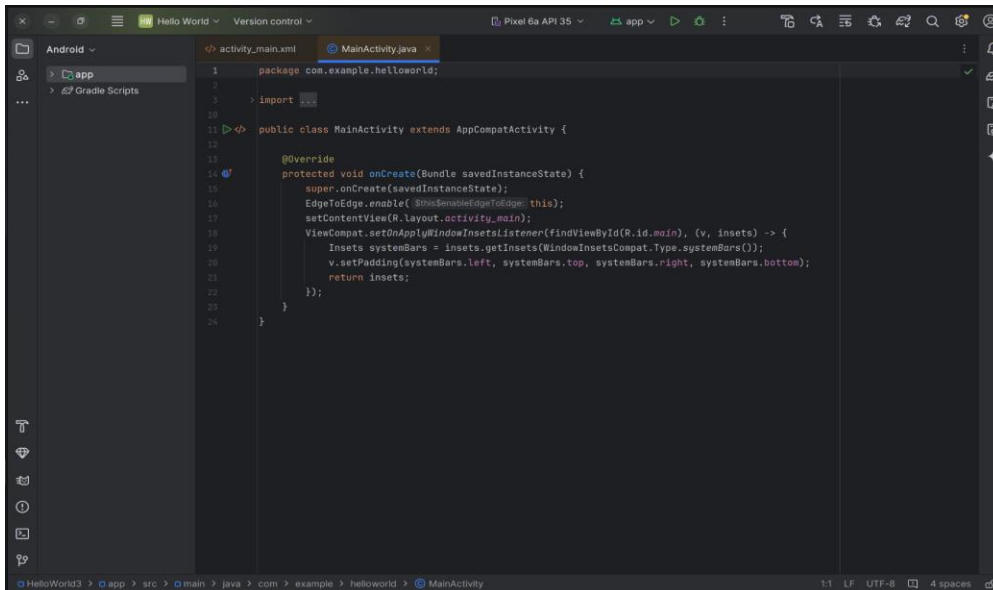
Mẹo: xem trang “[Cấu hình xây dựng của bạn](#)” trên trang web dành cho nhà phát triển để biết thông tin chi tiết.

Trình chỉnh sửa Android Studio xuất hiện. Hãy làm theo các bước sau:

- Nhấn vào tab **activity_main.xml** để xem trình chỉnh sửa bố cục.
- Nhấp vào tab **Design** của trình chỉnh sửa bố cục, nếu chưa chọn, để hiển thị bố cục của giao diện như minh họa bên dưới.



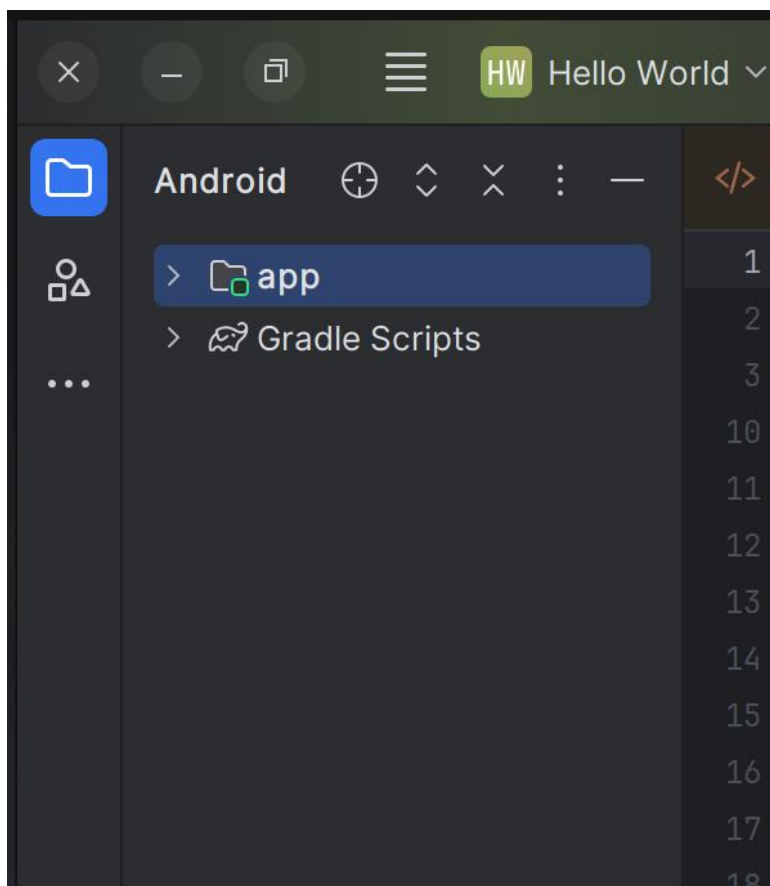
- Nhấp vào tab **MainActivity.java** để xem trình soạn thảo mã như hiển thị bên dưới.



2.2 Khám phá dự án > Ngăn Android

Trong phần thực hành này, bạn sẽ khám phá cách dự án được tổ chức trong Android Studio.

1. Nếu chưa được chọn, hãy nhấp vào tab **Project** trong cột tab dọc ở phía bên trái của cửa sổ Android Studio. Ngăn dự án xuất hiện.
2. Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, hãy chọn **Android** từ menu bật lên ở đầu ngăn dự án, như được hiển thị bên dưới.

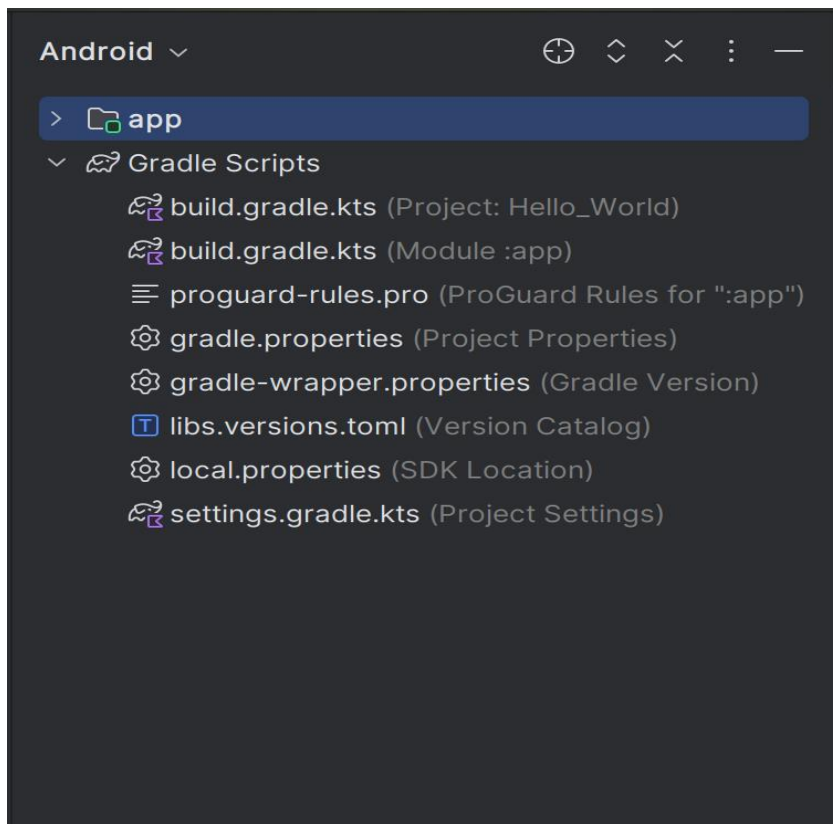


Lưu ý: Chương này và các chương khác đề cập đến khung dự án, khi được đặt thành **Android**, là khung **project > Android**

2.3 Khám phá thư mục Gradle Scripts

Hệ thống xây dựng Gradle trong Android Studio giúp dễ dàng thêm các tệp nhị phân bên ngoài hoặc các module thư viện khác vào bản dựng của mình dưới dạng các phần phụ thuộc.

Khi bạn lần đầu tiên tạo một dự án ứng dụng, ngăn Project > Android sẽ xuất hiện với thư mục Gradle Scripts được mở rộng như hình dưới đây.



Thực hiện các bước sau để khám phá hệ thống Gradle:

1. Nếu thư mục Gradle Scripts chưa được mở rộng, hãy nhấp vào biểu tượng tam giác để mở rộng nó. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống dựng.

2. Tìm tệp **build.gradle.kts** (Project: Hello_World)

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung tất cả các module tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một Gradle cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần chỉnh sửa tệp này nhưng việc hiểu nội dung của nó vẫn rất hữu ích.

Theo mặc định, tệp xây dựng cấp cao nhất sử dụng khối buildscript để xác định các kho lưu trữ Gradle vào các phụ thuộc chung cho tất cả các module trong dự án. Khi phụ thuộc của bạn không phải là thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong bất kỳ kho lưu trữ trực tuyến nào được chỉ định trong khối repositories của tệp này. Theo mặc định, các dự án mới tạo trong Android Studio khai báo JCenter và Google (bao gồm kho lưu trữ Google Maven) là các vị trí kho lưu trữ

```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2 plugins {
3     alias(libs.plugins.android.application) apply false
4 }
```

3. Tìm tệp **build.gradle** (Module:app)

Ngoài tệp **build.gradle** cấp dự án, mỗi module cũng có một tệp **build.gradle** riêng, cho phép bạn cấu hình các cài đặt cho từng module cụ thể (ứng dụng HelloWorld chỉ có một module). Việc cấu hình các cài đặt build này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như các loại build bổ sung và các biến thể sản phẩm. Bạn cũng có thể ghi đè các cài đặt trong tệp **AndroidManifest.xml** hoặc tệp **build.gradle** cấp cao.

Tệp này thường là tệp được chỉnh sửa nhiều nhất khi thay đổi cấu hình ở cấp độ ứng dụng, chẳng hạn như khai báo các phụ thuộc trong phần **dependencies**. Bạn có thể khai báo một phụ thuộc thư viện bằng cách sử dụng một trong nhiều cấu hình phụ thuộc khác nhau. Mỗi cấu hình phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ, câu lệnh `implementation fileTree(dir: 'libs', include: ['*.jar'])` thêm một phụ thuộc gồm tất cả các tệp “.jar” bên trong thư mục **libs**.

Dưới đây là tệp **build.gradle**(Module:app) cho ứng dụng HelloWorld.

```

1  plugins {
2      alias(libs.plugins.android.application)
3  }
4
5  android {
6      namespace = "com.example.helloworld"
7      compileSdk = 35
8
9      defaultConfig {
10         applicationId = "com.example.helloworld"
11         minSdk = 24
12         targetSdk = 35
13         versionCode = 1
14         versionName = "1.0"
15
16         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes {
20         release {
21             isMinifyEnabled = false
22             proguardFiles(
23                 getDefaultProguardFile("proguard-android-optimize.txt"),
24                 "proguard-rules.pro"
25             )
26         }
27     }
28     compileOptions {
29         sourceCompatibility = JavaVersion.VERSION_11
30         targetCompatibility = JavaVersion.VERSION_11
31     }
32 }
33
34 dependencies {
35
36     implementation(libs.appcompat)
37     implementation(libs.material)
38     implementation(libs.activity)
39     implementation(libs.constraintlayout)
40     testImplementation(libs.junit)
41     androidTestImplementation(libs.ext.junit)
42     androidTestImplementation(libs.espresso.core)
43 }

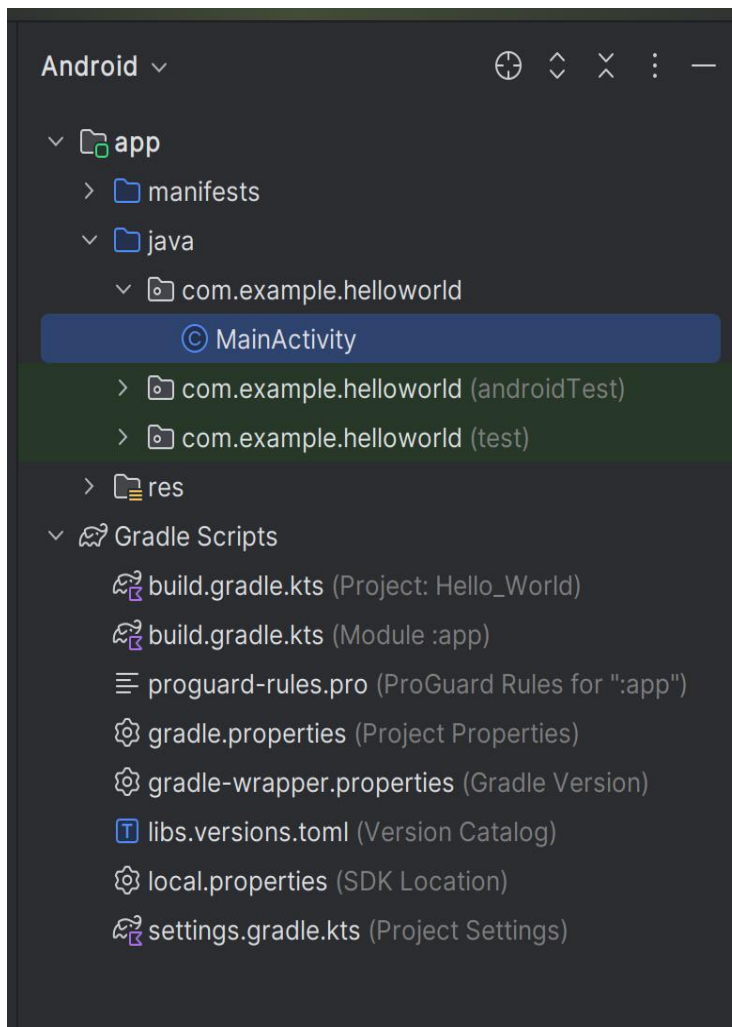
```

4. Nhấp vào hình tam giác để đóng **Gradle Scripts**.

2.4 Khám phá thư mục app và res

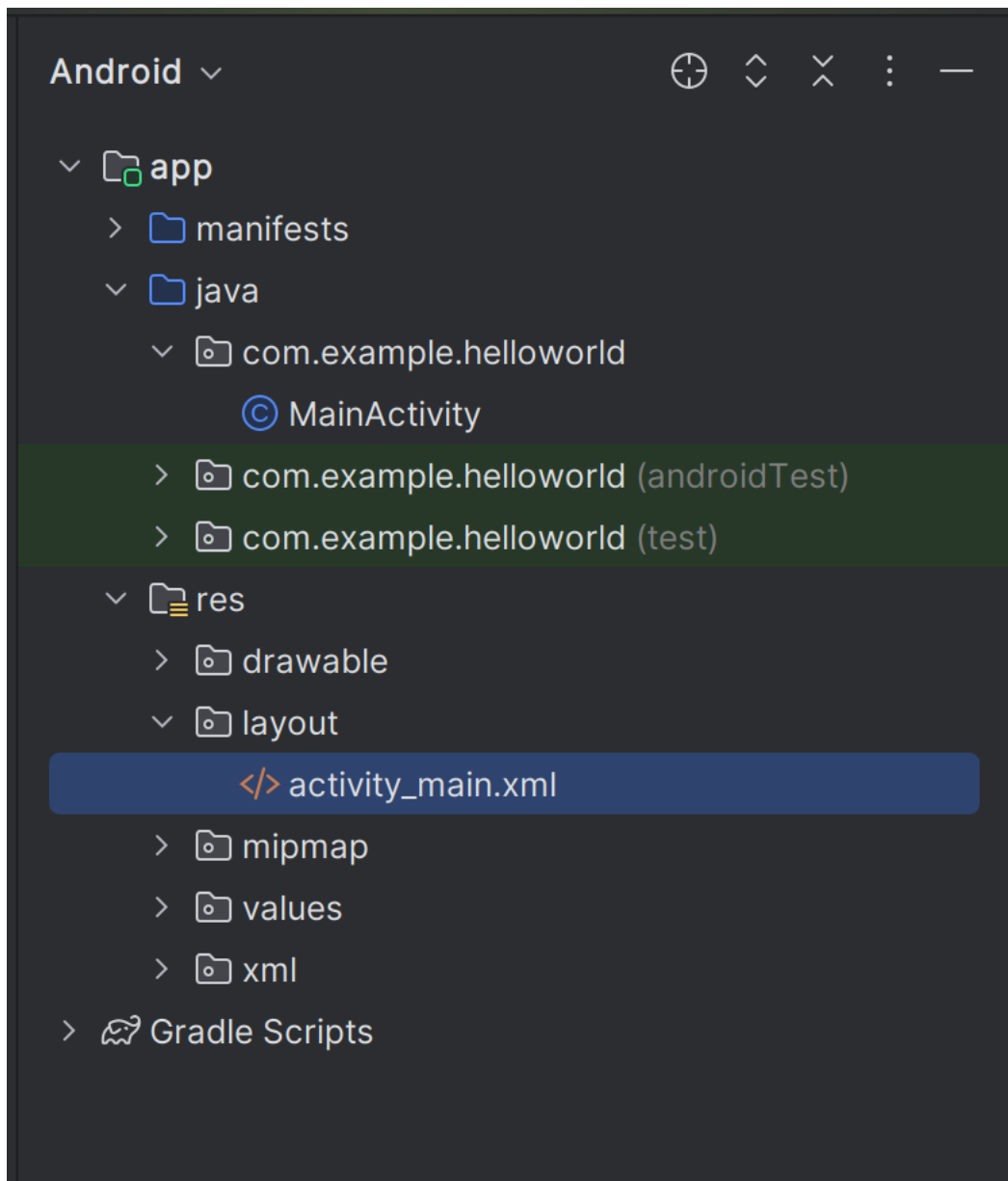
Tất cả mã nguồn và tài nguyên cho ứng dụng đều nằm trong các thư mục app và res.

1. Mở rộng thư mục **app**, thư mục **java**, và thư mục **com.example.android.helloworld** để xem tệp Java **MainActivity**. Nhấp đúp vào tệp sẽ mở nó trong trình chỉnh sửa mã.



Thư mục **java** bao gồm các tệp lớp Java trong ba thư mục con, như được hiển thị trong hình trên. Thư mục **com.example.hello.helloworld** (hoặc tên miền mà bạn đã chỉ định) chứa tất cả các tệp cho một gói ứng dụng. Hai thư mục còn lại được sử dụng để kiểm thử và sẽ được mô tả trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một gói duy nhất và nó chứa tệp **MainActivity.java**. Tên của **Activity** đầu tiên (màn hình) mà người dùng nhìn thấy, đồng thời cũng khởi tạo các tài nguyên toàn ứng dụng, thường được gọi là **MainActivity** (phần mở rộng tệp bị lược bỏ trong ngăn **Project > Android**).

2. Mở rộng thư mục **res** và thư mục **layout**, sau đó nhấp đúp vào tệp **activity_main.xml** để mở nó trong trình chỉnh sửa bố cục.



Thư mục **res** chứa các tài nguyên như bố cục, chuỗi và hình ảnh. Một Activity thường được liên kết với một bố cục của các khung nhìn giao diện người dùng, được định nghĩa dưới dạng tệp XML. Tệp này thường được đặt tên theo Activity của nó.

2.5 Khám phá thư mục manifests.

Thư mục **manifests** chứa các tệp cung cấp thông tin quan trọng về ứng dụng của bạn cho hệ thống Android. Hệ thống cần có những thông tin này trước khi có thể chạy bất kỳ mã nào của ứng dụng.

1. Mở rộng thư mục **manifests**.
2. Mở tệp **AndroidManifest.xml**

Tệp **AndroidManifest.xml** mô tả tất cả các thành phần của ứng dụng Android. Mọi thành phần trong ứng dụng, chẳng hạn như từng **Activity**, phải được khai báo trong tệp XML này. Trong các bài học tiếp theo, bạn sẽ chỉnh sửa tệp này để thêm các tính năng và quyền truy cập tính năng. Để tìm hiểu thêm, hãy xem Tổng quan về App Manifest.

Nhiệm vụ 3: Sử dụng thiết bị ảo (trình giả lập).

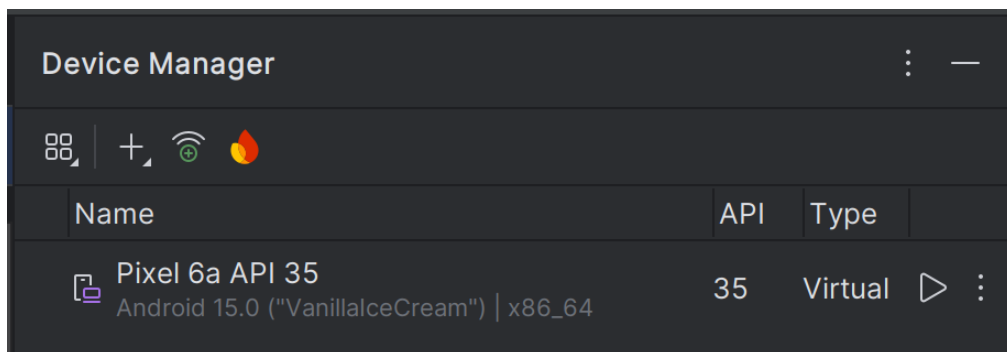
Trong nhiệm vụ này, bạn sẽ sử dụng **Android Virtual Device (AVD) Manager** để tạo một thiết bị ảo (còn được gọi là trình giả lập) mô phỏng cấu hình của một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng Android Emulator có các yêu cầu bổ sung so với yêu cầu hệ thống cơ bản của Android Studio.

Sử dụng AVD Manager, bạn có thể xác định các đặc điểm phần cứng của một thiết bị, mức API của nó, dung lượng lưu trữ, giao diện và các thuộc tính khác, rồi lưu nó dưới dạng một thiết bị ảo. Với các thiết bị ảo, bạn có thể kiểm thử ứng dụng trên các cấu hình thiết bị khác nhau (như máy tính bảng và điện thoại) với các mức API khác nhau, mà không cần sử dụng thiết bị thực tế.

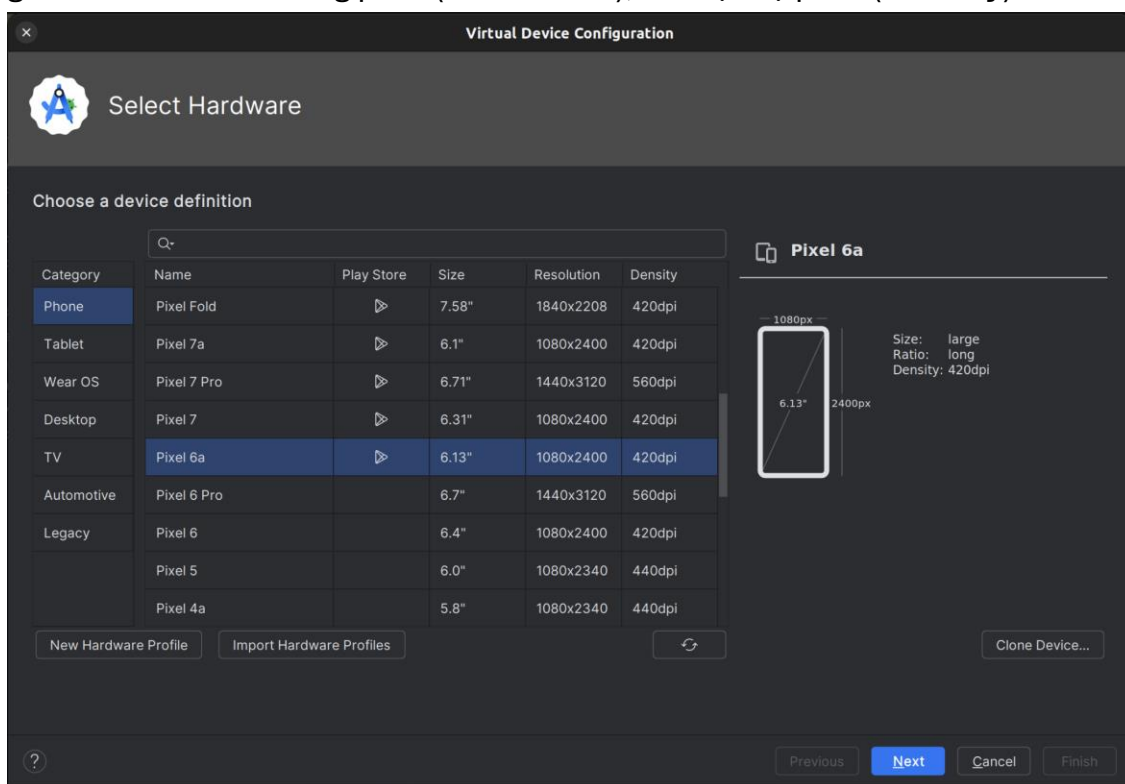
3.1 Tạo một thiết bị ảo Android (AVD).

Để chạy một trình giả lập trên máy tính của bạn, bạn cần tạo một cấu hình mô tả thiết bị ảo.

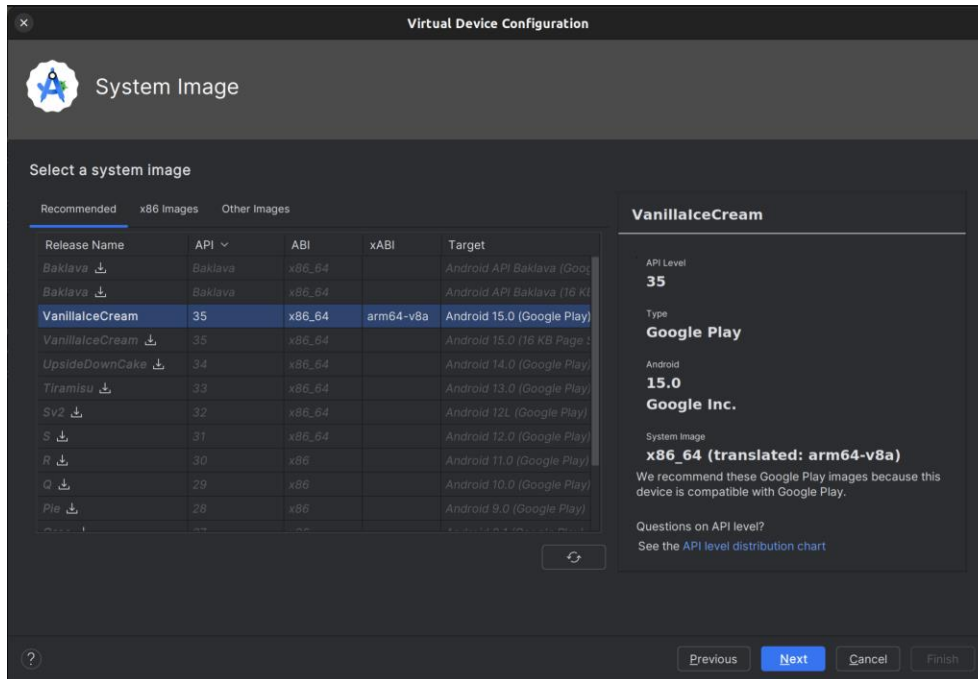
1. Trong Android Studio, chọn **Tools > Android > AVD Manager**, hoặc nhấp vào biểu tượng AVD Manager trên thanh công cụ. Màn hình **Your Virtual Devices** sẽ xuất hiện. Nếu bạn đã tạo các thiết bị ảo trước đó, màn hình sẽ hiển thị chúng (như trong hình dưới đây); nếu không, bạn sẽ thấy một danh sách trống.



2. Nhấp vào **+Create Virtual Device**. Cửa sổ **Select Hardware** sẽ xuất hiện, hiển thị danh sách các thiết bị phần cứng được cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo (**Size**), độ phân giải màn hình tính bằng pixel (**Resolution**), và mật độ pixel (**Density**).



3. Chọn một thiết bị như **Pixel 6a** sau đó nhấp vào **Next**. Màn hình **System Image** sẽ xuất hiện.
4. Nhấp vào tab **Recommended** nếu nó chưa được chọn, và chọn phiên bản của hệ điều hành Android để chạy trên thiết bị ảo (ví dụ như **VanillalceCream**).



Có nhiều phiên bản khác ngoài những phiên bản được hiển thị trong tab **Recommended**. Hãy xem các tab **x86 Images** và **Other Images** để thấy chúng.

Nếu một liên kết **Download** xuất hiện bên cạnh hình ảnh hệ thống mà bạn muốn sử dụng, điều đó có nghĩa là nó chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống, và nhấp vào **Finish** khi quá trình hoàn tất.

- Sau khi chọn một hình ảnh hệ thống, nhấp vào **Next**. Cửa sổ **Android Virtual Device (AVD)** sẽ xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào **Finish**.

1.2) Giao diện người dùng tương tác đầu tiên

1.3) Trình chỉnh sửa bố cục

1.4) Văn bản và các chế độ cuộn

1.5) Tài nguyên có sẵn

Bài 2) Activities

2.1) Activity và Intent

2.2) Vòng đời của Activity và trạng thái

2.3) Intent ngầm định

Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ

3.1) Trình gỡ lỗi

3.2) Kiểm thử đơn vị

3.3) Thư viện hỗ trợ

CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG

Bài 1) Tương tác người dùng

- 1.1) Hình ảnh có thể chọn
- 1.2) Các điều khiển nhập liệu
- 1.3) Menu và bộ chọn
- 1.4) Điều hướng người dùng
- 1.5) RecyclerView

Bài 2) Trải nghiệm người dùng thú vị

- 2.1) Hình vẽ, định kiểu và chủ đề
- 2.2) Thẻ và màu sắc
- 2.3) Bố cục thích ứng

Bài 3) Kiểm thử giao diện người dùng

- 3.1) Espresso cho việc kiểm tra UI

CHƯƠNG 3. LÀM VIỆC TRONG NỀN

Bài 1) Các tác vụ nền

- 1.1) AsyncTask
- 1.2) AsyncTask và AsyncTaskLoader
- 1.3) Broadcast receivers

Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền

- 2.1) Thông báo
- 2.2) Trình quản lý cảnh báo
- 2.3) JobScheduler

CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG

Bài 1) Tùy chọn và cài đặt

1.1) Shared preferences

1.2) Cài đặt ứng dụng

Bài 2) Lưu trữ dữ liệu với Room

2.1) Room, LiveData và ViewModel

2.2) Room, LiveData và ViewModel