

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	2
Bài 1) Tạo ứng dụng đầu tiên	2
1.1) Android Studio và Hello World	2
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 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 tìm hiểu 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 một trình giả lập và trên một thiết bị vật lý.

Những gì Bạn nên biết

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng lập trình hướng đối tượng sử dụng một IDE (môi trường phát triển tích hợp) như Android Studio.
- Chứng minh rằng bạn có ít nhất 1-3 năm kinh nghiệm trong lập trình hướng đối tượng, với một phần trong số đó tập trung vào ngôn ngữ lập trình Java. (Các bài thực hành này sẽ không giải thích về lập trình hướng đối tượng hoặc ngôn ngữ Java.

Những gì Bạn sẽ cần:

- Một máy tính chạy Windows hoặc Linux, hoặc một Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một phương pháp thay thế để tải các cài đặt 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 ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.
- Cách thêm thông điệp ghi lại vào 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** trên các thiết bị ảo và vật lý.
- Khám phá cấu trúc dự án.
- Tạo và xem các thông điệp ghi lại từ ứng dụng của bạn.
- Khám phá tệp **AndroidManifest.xml**

Tổng quan về ứng dụng

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

Task1: Cài đặt Android Studio

- **Android Studio** cung cấp một môi trường phát triển tích hợp (IDE) hoàn chỉnh bao gồm trình soạn thảo mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ hỗ trợ phát triển, gỡ lỗi, kiểm thử và tối ưu hiệu suất, giúp việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn. Bạn có thể kiểm thử ứng dụng của mình trên nhiều trình giả lập được cấu hình sẵn hoặc trên thiết bị di động của chính mình, tạo ứng dụng chính thức và phát hành trên cửa hàng Google Play.
- **Lưu ý : Android Studio** liên tục được cải tiến. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, hãy xem **Android Studio**.
- **Android Studio** có sẵn cho các máy tính chạy Windows hoặc Linux và máy MAC chạy macOS. Phiên bản OpenJDK (Java Development Kit) mới nhất được tích hợp sẵn với Android Studio.
- Để thiết lập và chạy Android Studio, trước tiên hãy kiểm tra yêu cầu hệ thống để đảm bảo hệ thống bạn đáp ứng được. Quá trình cài đặt tương tự trên tất cả các nền tảng. Mọi điểm khác biệt sẽ được ghi chú bên dưới.
 1. Truy cập trang web dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.
 2. Chấp nhận cấu hình đặc biệt 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 thiết lập sẽ tải xuống và cài đặt một số thành phần bổ sung bao gồm Android SDK. Hãy kiên nhẫn, quá trình này có thể mất một chút thời gian tùy thuộc vào tốc độ Internet của bạn và một số bước có thể trông có vẻ lặp lại.
 4. Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động và bạn đã sẵn sàng tạo dự án đầu tiên của mình.
- Khắc phục sự cố : Nếu bạn gặp vấn đề trong quá trình cài đặt , hãy kiểm tra ghi chú phát hành của Android Studio hoặc nhờ sự trợ giúp từ giảng viên của bạn.

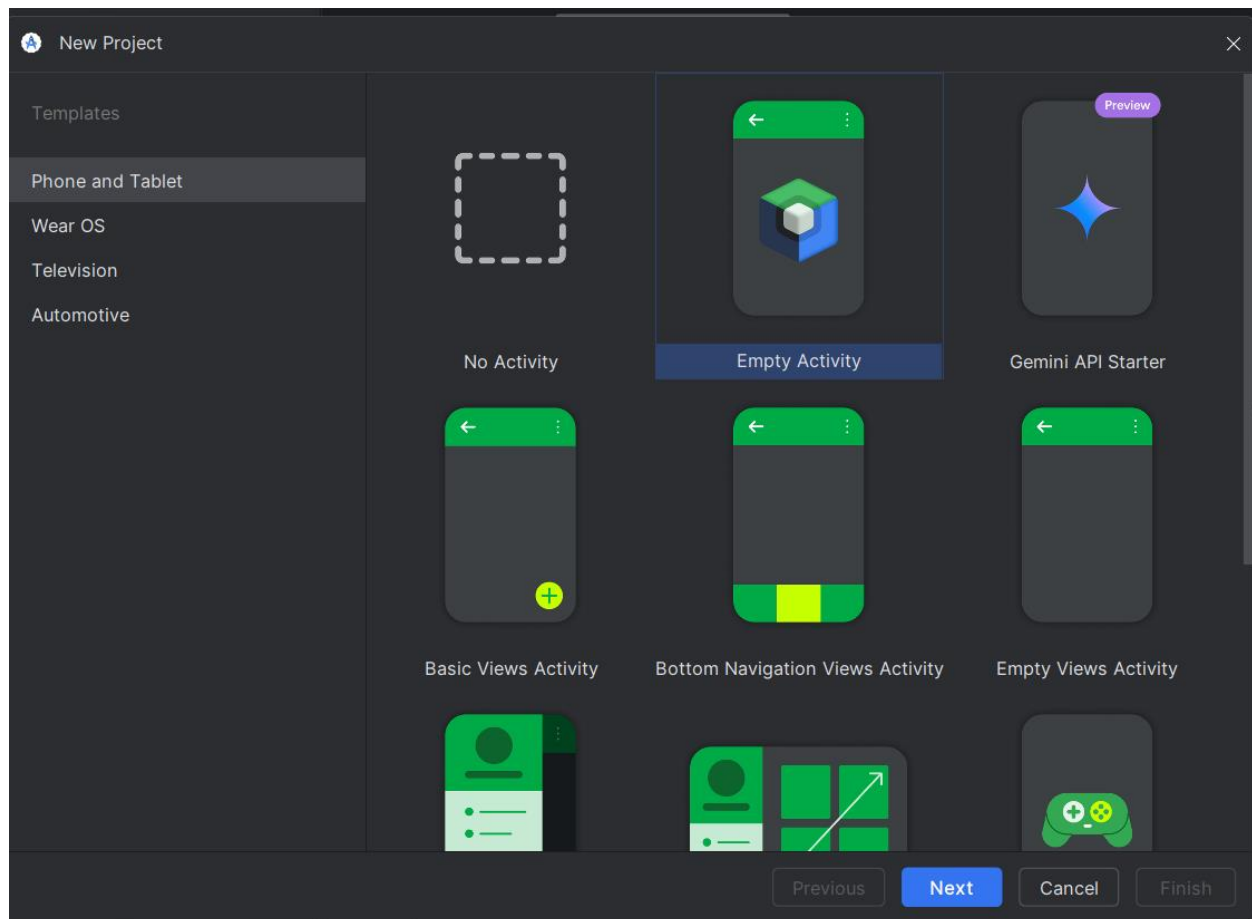
Task 2 : Tạo ứng dụng Hello World

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

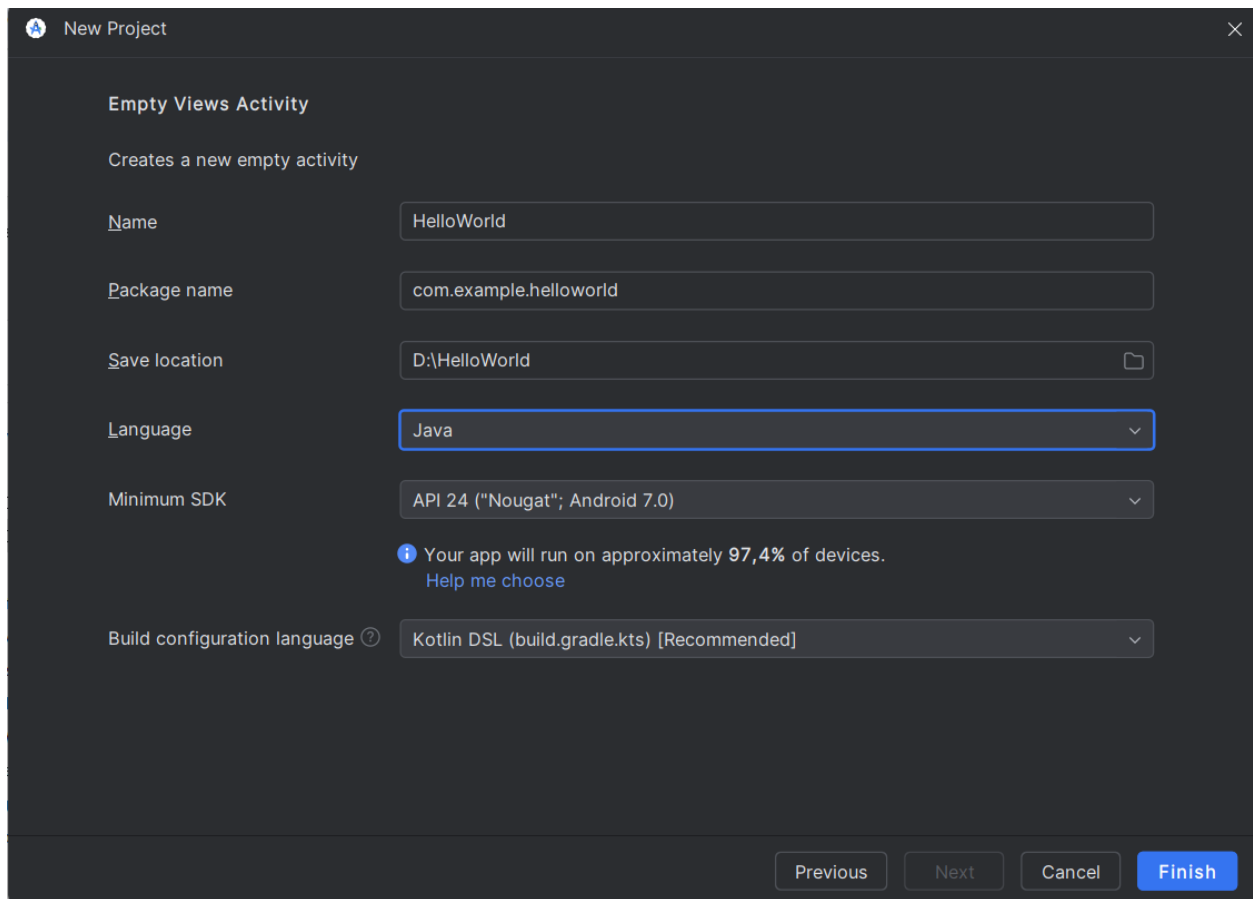
2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu nó chưa được mở
2. Trong cửa sổ Welcome to Android Studio , nhấn vào Start a new Android Studio Project.
3. Trong cửa sổ Create Android Studio, nhập Hello World cho Application name.
4. Xác minh rằng Project location mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác hoặc thay đổi nó sang thư mục mà bạn ưu tiên.
5. Chấp nhận android.example.com mặc định cho Company Domain, hoặc tạo một tên miền công ty riêng.
Nếu bạn không có kế hoạch phát hành ứng dụng của mình, bạn có thể giữ nguyên mặc định. Lưu ý rằng việc thay đổi tên gói của ứng dụng sau này sẽ tốn thêm công sức.
6. Để nguyên không chọn các tùy chọn Include C++ support và Include Kotlin support và nhấn Next
7. Trên màn hình Target Android Devices, đảm bảo rằng Phone and Table được chọn. Đảm bảo API 15 : Android 4.0.3 IceCreamSandwich được làm bằng MiniumSDK; Nếu không, hãy sử dụng popup menu để thiết lập nó.
Đây là các cài đặt được sử dụng cho các ví dụ trong các bài học của khóa học này. Tính tới thời điểm hiện tại, các cài đặt này giúp ứng dụng Hello World của bạn tương thích với 97% thiết bị Android đang hoạt động trên Google Play Store.
8. Để nguyên không chọn Include Instant App support và tất cả các tùy chọn khác. Sau đó, nhấn Next. Nếu dự án của bạn yêu cầu các thành phần bổ sung cho target SDK đã chọn, Android Studio sẽ tự động cài đặt chúng.
9. Cửa sổ Add an Activity xuất hiện. Activity là một thành phần quan trọng trong bất kỳ ứng dụng Android nào, đại diện cho một tác vụ đơn lẻ mà người dùng có thể thực hiện. Thông thường một Activity sẽ đi kèm với một bố cục(layout) xác định cách các thành phần giao diện người dùng (UI) hiện thị trên màn hình. Android Studio cung cấp các mẫu Activity để

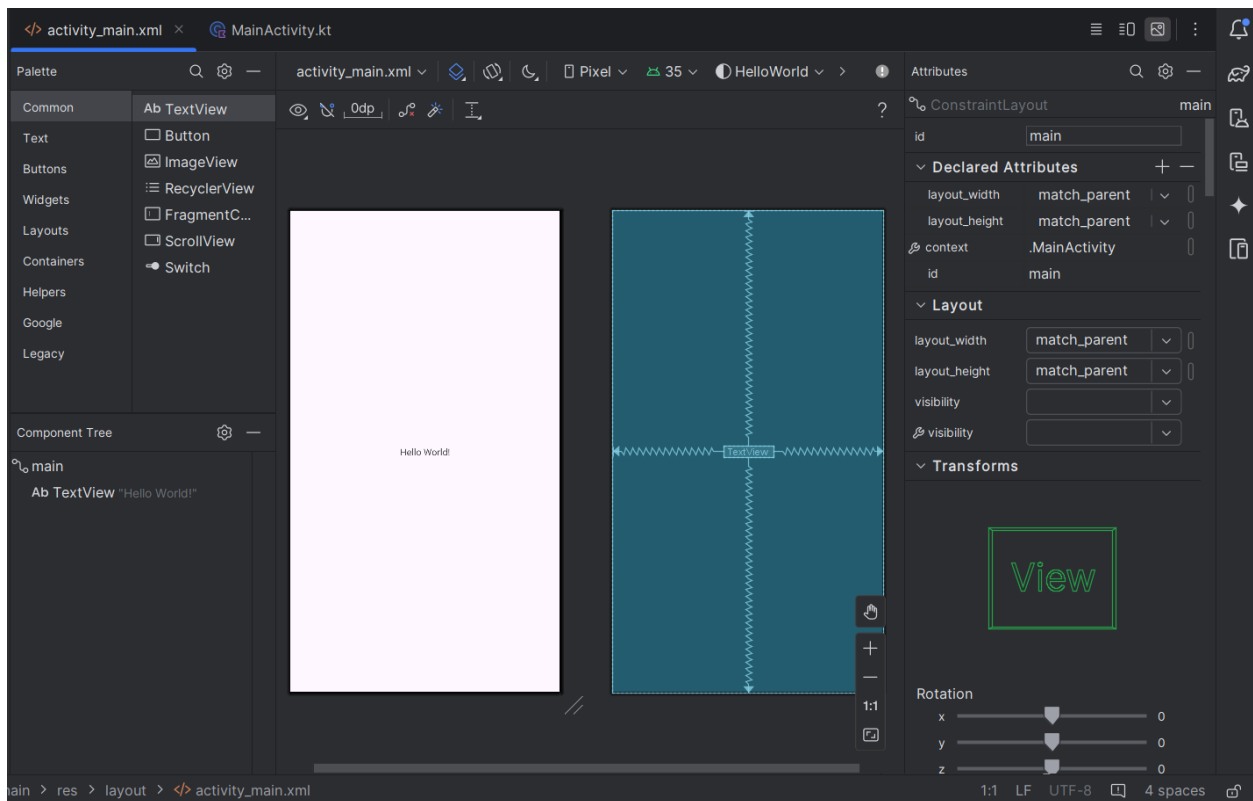
giúp bạn bắt đầu. Đối với dự án Hello World, hãy chọn Empty Activity như bên hình dưới và nhấn Next.



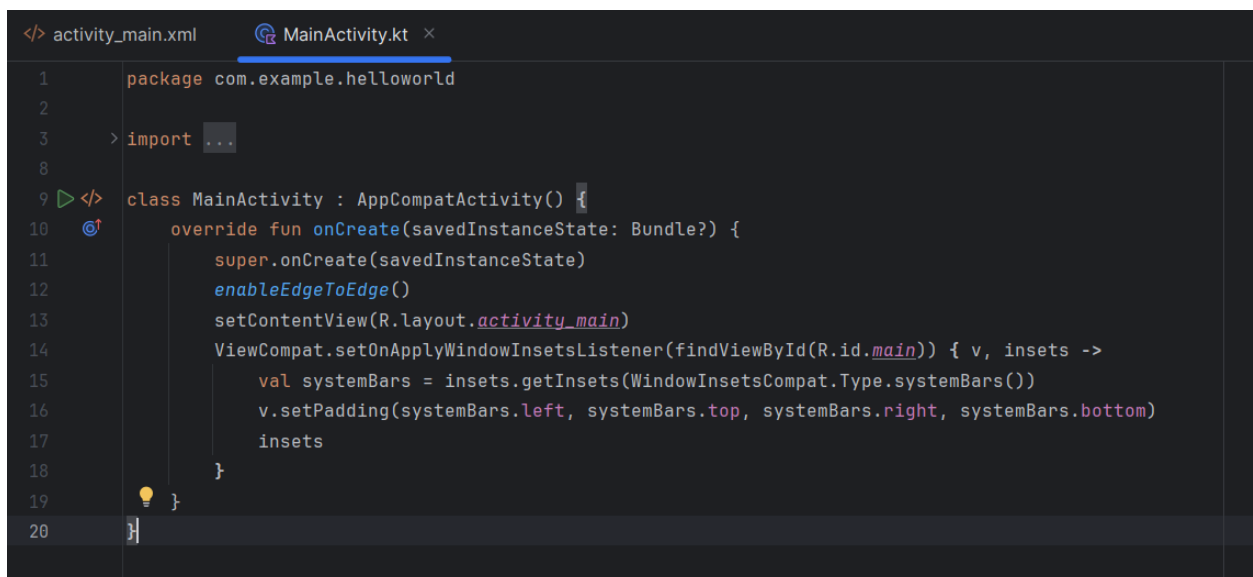
10. Màn hình Configure Activity xuất hiện (giao diện này có thể khác nhau tùy thuộc vào mẫu mà bạn đã chọn ở bước trước). Theo mặc định, Empty Activity do mẫu cung cấp được đặt tên là MainActivity. Bạn có thể thay đổi tên này nếu muốn, nhưng trong bài học này chúng ta sẽ sử dụng MainActivity.
11. Đảm bảo rằng tùy chọn Generate Layout file được chọn. Tên bố cục mặc định là activity_main. Bạn có thể thay đổi tên này nếu muốn, nhưng trong bài học này chúng ta sẽ sử dụng activity_main.
12. Đảm bảo rằng tùy chọn Backwards Compatibility(App Compat) được chọn. Tùy chọn này giúp ứng dụng của bạn tương thích ngược với các phiên bản Android trước đó.
13. Nhấn Finish.



- Android Studio sẽ 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 : Tham khảo trang dành cho nhà phát triển Configure your build để biết thông tin chi tiết.
- Bạn cũng có thể thấy thông báo “Tip of the day” với các phím tắt và mẹo hữu ích khác. Nhấp Close để tắt thông báo này
- Trình soạn thảo của Android Studio xuất hiện. Hãy làm theo các bước sau :
 1. Nhấp vào activiti_main.xml để xem trình chỉnh sửa bố cục.
 2. Nhấp vào tab Design trong trình chỉnh sửa bố cục (nếu chưa được chọn) để hiển thị bản xem trước giao diện đồ họa của bố cục như hình dưới đây.



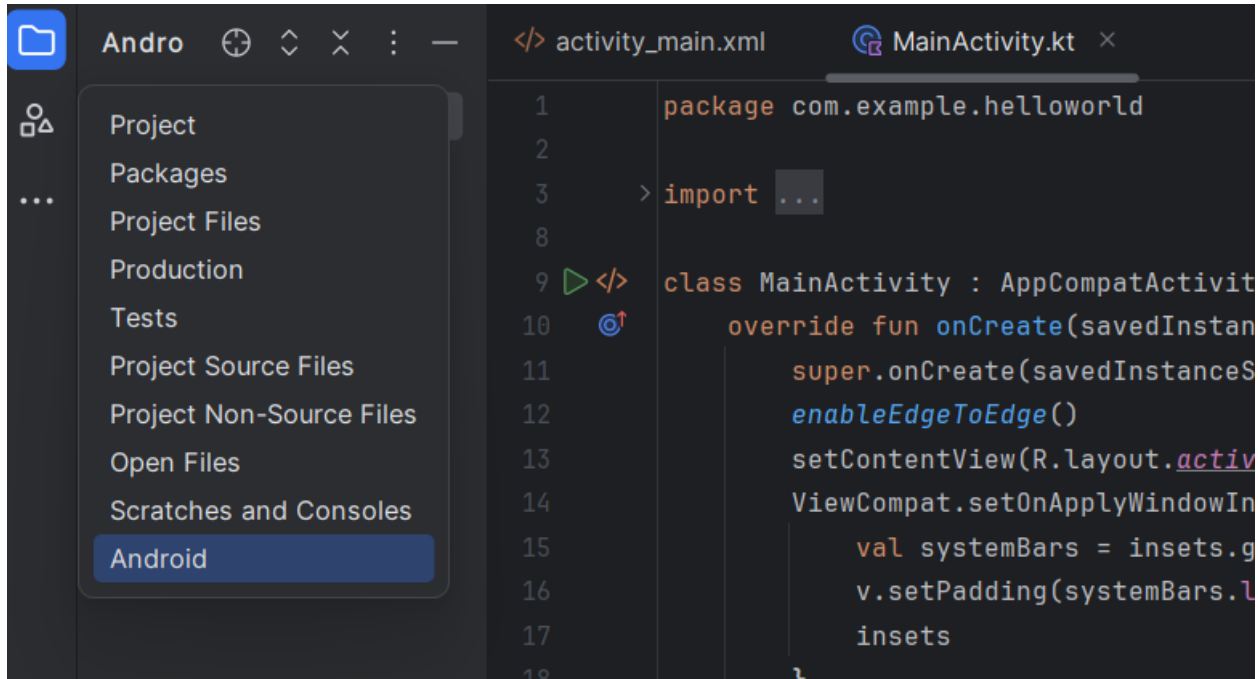
3. Nhấp vào tab MainActivity.java để xem trình chỉnh sửa mã nguồn như hình dưới



2.2 Khám phá Project > Android pane

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

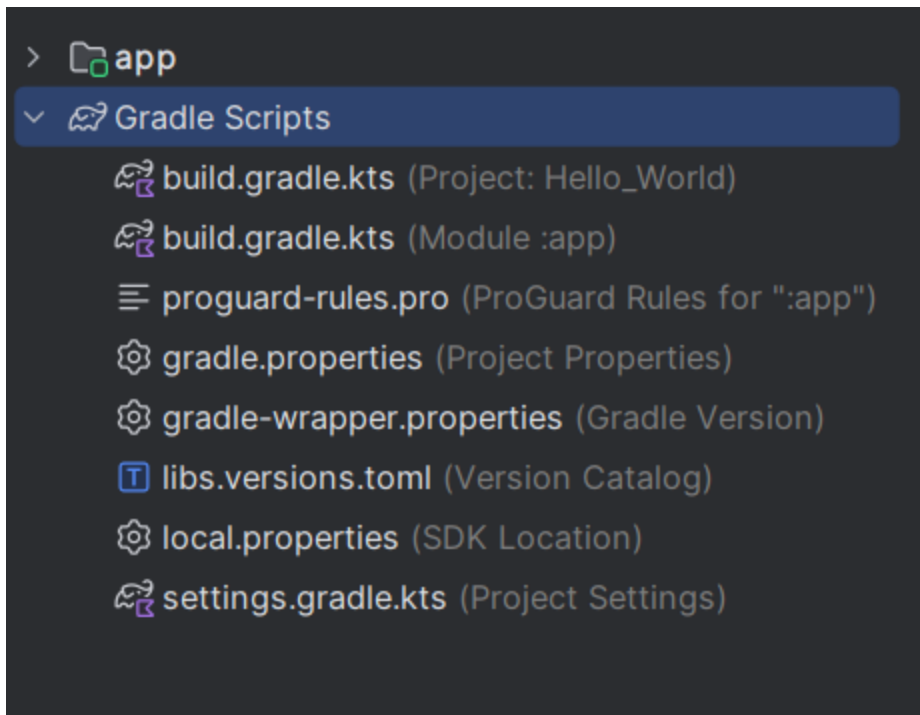
1. Nếu chưa được chọn, hãy nhấp vào tab Project trong cột tab dọc ở phía bên trái cửa sổ Android Studio. Project pane sẽ xuất hiện.
2. Để xem Project theo tiêu chuẩn của Android Project , hãy chọn Android từ popup menu ở đầu Project pane, như hình bên dưới



Lưu ý : Chương này và các chương khác đề cập đến Project khi được đặt thành là Android là Project -> Android pane

2.3 Khám phá thư mục Gradle Scripts

- Hệ thống dựng Gradle trong Android Studio giúp bạn dễ dàng thêm các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng dưới dạng phụ thuộc.
- Khi bạn lần đầu tạo một dự án ứng dụng, Project -> Android pane sẽ hiện với thư mục Gradle Scripts được mở rộng như hình bên 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 (Project: HelloWorld).
 - Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô -đun trong Project của mình. Mỗi Project trong Android Studio đều chứa một tệp Gradle build cấp cao nhất. Hầu hết thời gian, bạn không cần chỉnh sửa tệp này, nhưng việc hiểu nội dung của nó vẫn rất hữu ích.
 - Theo mặc định, tệp build cấp cao nhất sử dụng khối buildscripts để xác định các kho lưu trữ Gradle và các phần phụ thuộc chung cho tất cả các mô-đun trong Project. Khi phần phụ thuộc của bạn không phải là thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm các tệp trong các kho lưu trữ trực tuyến được chỉ định trong khối repositories của tệp này. Theo mặc định, các Project mới trong Android Studio khai báo JCenter và Google (bao gồm Google Maven repository) làm vị trí kho lưu trữ:

```
> activity_main.xml    MainActivity.kt    build.gradle.kts (Hello World) ×    build.gradle.kts (:app)
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     alias(libs.plugins.kotlin.android) apply false
5 }
```

3. Tìm tệp build.gradle(Module:app).

- Ngoài tệp build.gradle cấp dự án, mỗi mô-đun đều có tệp build.gradle riêng, tệp này cho phép bạn cấu hình cài đặt build riêng cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Việc cấu hình các cài đặt bản dựng cho phép bạn cung cấp các tùy chọn gói tùy chỉnh, chẳng hạn như các kiểu build và product flavors bổ sung. Bạn cũng có thể ghi đè các cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp cao nhất.
- Đây là tệp thường được chỉnh sửa nhất khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phần phụ thuộc trong phần dependencies. Bạn có thể khai báo một phụ thuộc thư viện bằng một trong số các 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 cho tất cả các tệp ".jar" bên trong thư mục libs.
- Dưới đây là tệp build.gradle(Module:app) dành cho ứng dụng HelloWorld:

```
o plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.helloworld"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.helloworld"
        minSdk = 24
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner =
```

```

"androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-
optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
    kotlinOptions {
        jvmTarget = "11"
    }
}

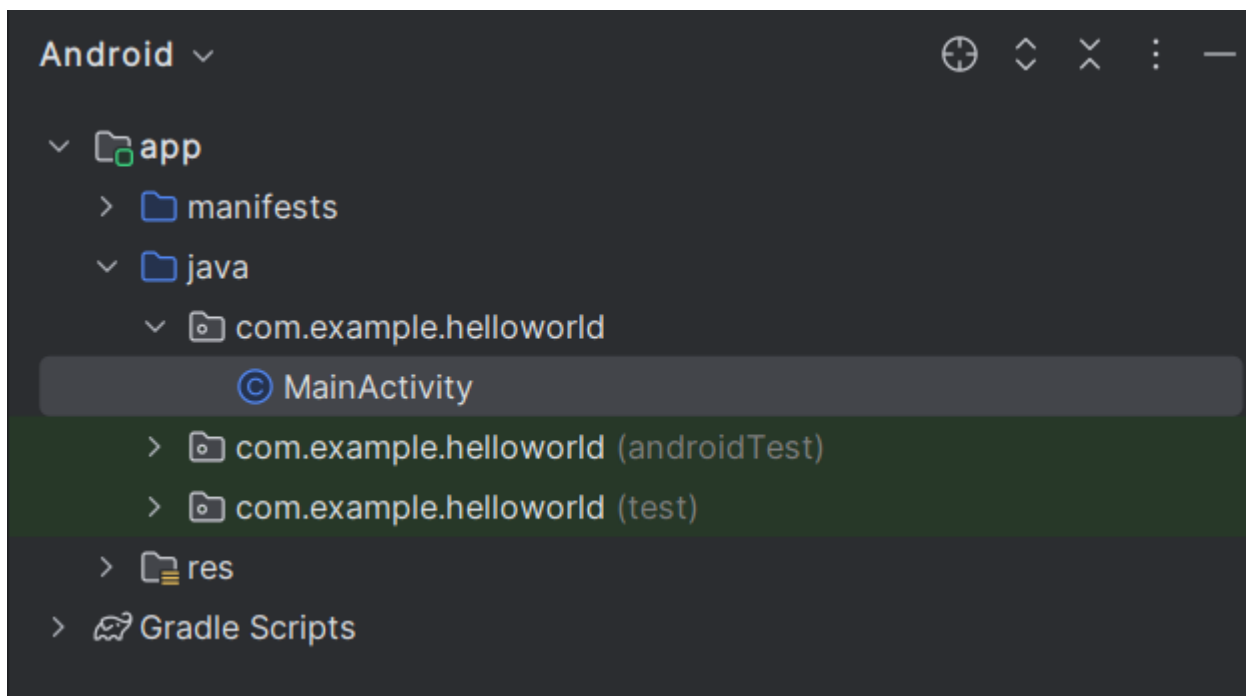
dependencies {
    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}

```

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

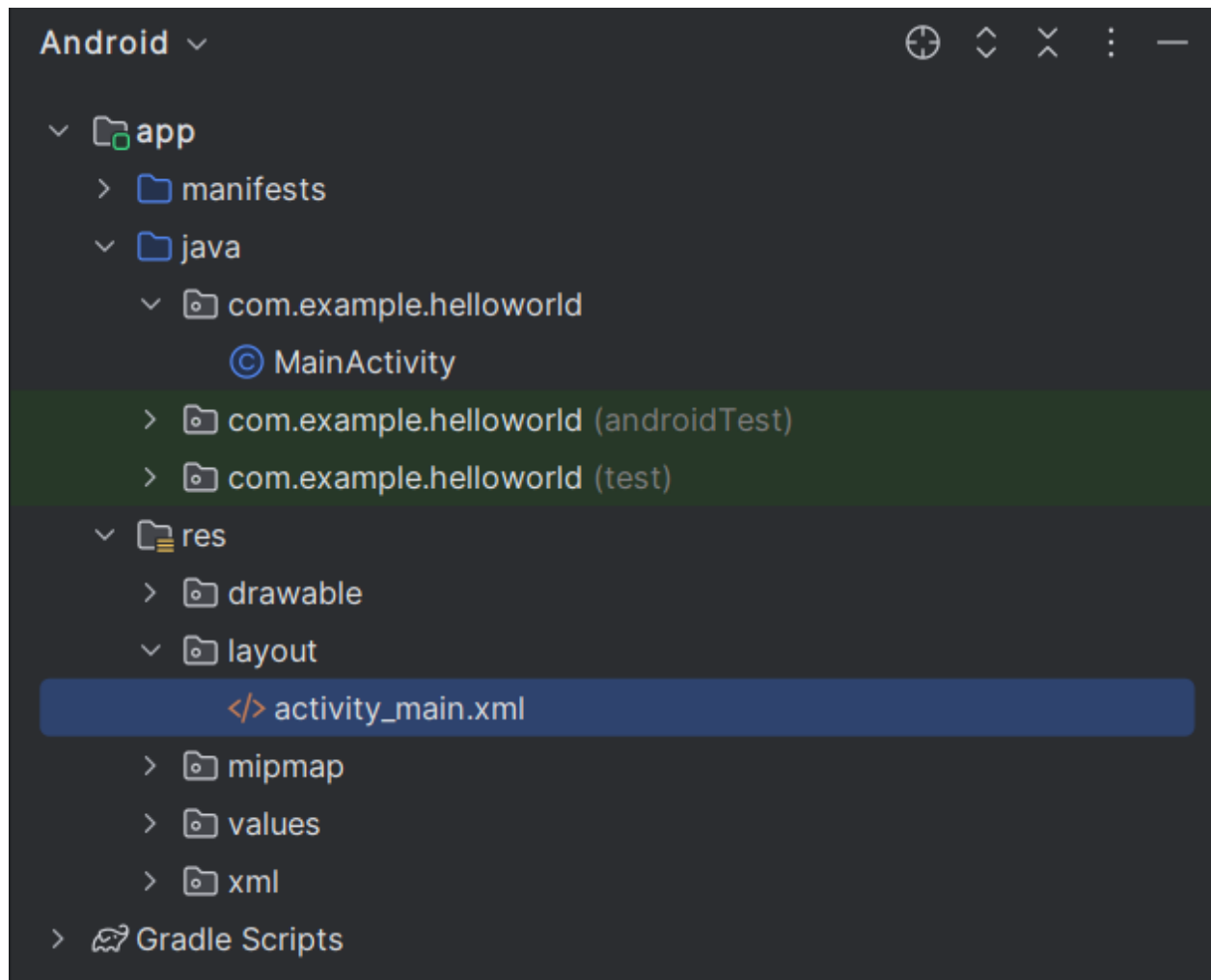
2.4 Khám phá ứng dụng và thư mục res

- 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 MainActivity. Nhấp đúp vào tệp để mở nó trong trình soạn thảo mã.



Thư mục Java bao gồm các tệp lớp Java trong ba thư mục con, như hình minh họa trên. Thư mục `com.example.android.helloworld` (hoặc tên miền bạn đã chỉ định) chứa tất cả các tệp trong một gói ứng dụng. Hai thư mục còn lại được sử dụng cho việc kiểm thử và sẽ được giải thích trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa tệp `MainActivity.java`. `MainActivity` là tên thông dụng cho Activity đầu tiên mà người dùng nhìn thấy (trong Project->Android pane phần mở rộng tệp thường bị ẩn đi).

2. Mở rộng thư mục `res` và thư mục `layout` và nhấp đúp vào tệp `activity_main.xml` để mở nó trong trình chỉnh sửa layout.



- Thư mục res chứa các tài nguyên như layouts, strings và images. Một Activity thường được liên kết với một bố cục giao diện người dùng được định nghĩa dưới tệp XML. Tệp này thường được đặt tên theo tên của Activity tương ứng.

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 bắt buộc phải có những thông tin này trước khi có thể chạy bất kì 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 của bạn. Mọi thành phần trong ứng dụng, chẳng hạn như mỗi **Activity**, đều phải được khai báo trong tệp **XML** này. Trong các bài học khác, bạn sẽ chỉnh sửa tệp

này để thêm các tính năng và quyền cho ứng dụng. Để tìm hiểu tổng quan, hãy xem **App Manifest Overview**.


TASK 3 : Sử dụng thiết bị ảo (trình giả lập)

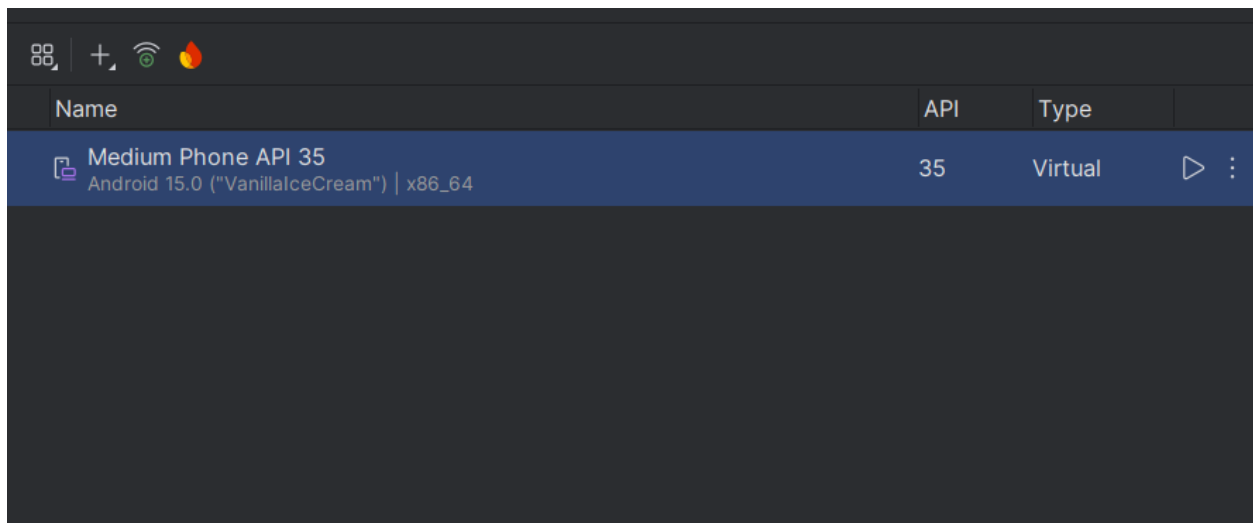
- Trong nhiệm vụ này, bạn sẽ sử dụng Android Virtual Device(ADV) manager để tạo một thiết bị ảo (còn gọi là **trình giả lập**) mô phỏng cấu hình của một loại thiết bị Android cụ thể, sau đó sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý trình giả lập Android yêu cầu thêm một số yêu cầu hệ thống ngoài các yêu cầu cơ bản cho Android Studio.
- Khi sử dụng AVD Manager bạn có thể xác định các đặc điểm phần cứng của thiết bị, chọn mức độ API, bộ nhớ , giao diện và các thuộc tính khác và lưu cấu hình đó dưới dạng một thiết ảo. Với các thiết bị ảo, bạn có thể kiểm tra ứng dụng trên các cấu hình thiết bị khác nhau (như máy tính bảng và điện thoại) với các cấp độ API khác nhau mà không cần sử dụng các thiết bị vật lý.

3.1 Tạo Android virtual device (ADV)

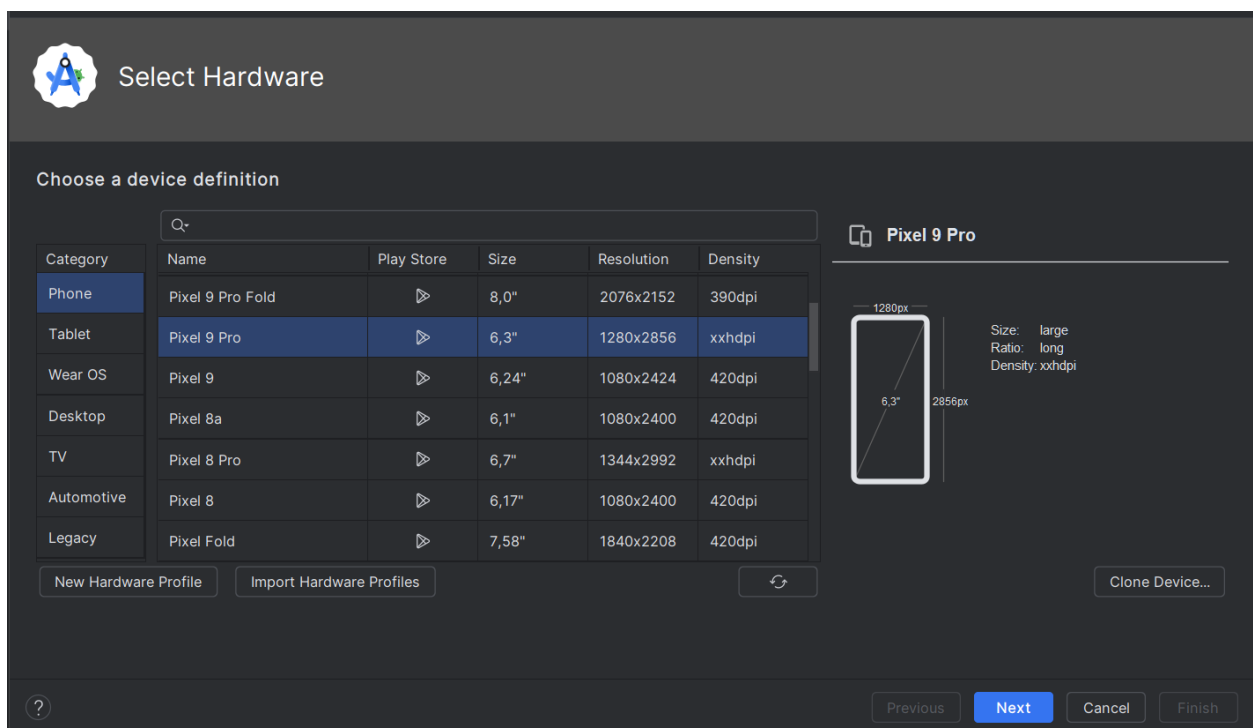
Để chạy trình giả lập trên máy tính của bạn, bạn cần tạo một cấu hình mô tả thiết bị ảo đó.

1. Trong **Android Studio**, chọn **Tools > Android > AVD Manager** hoặc nhấp

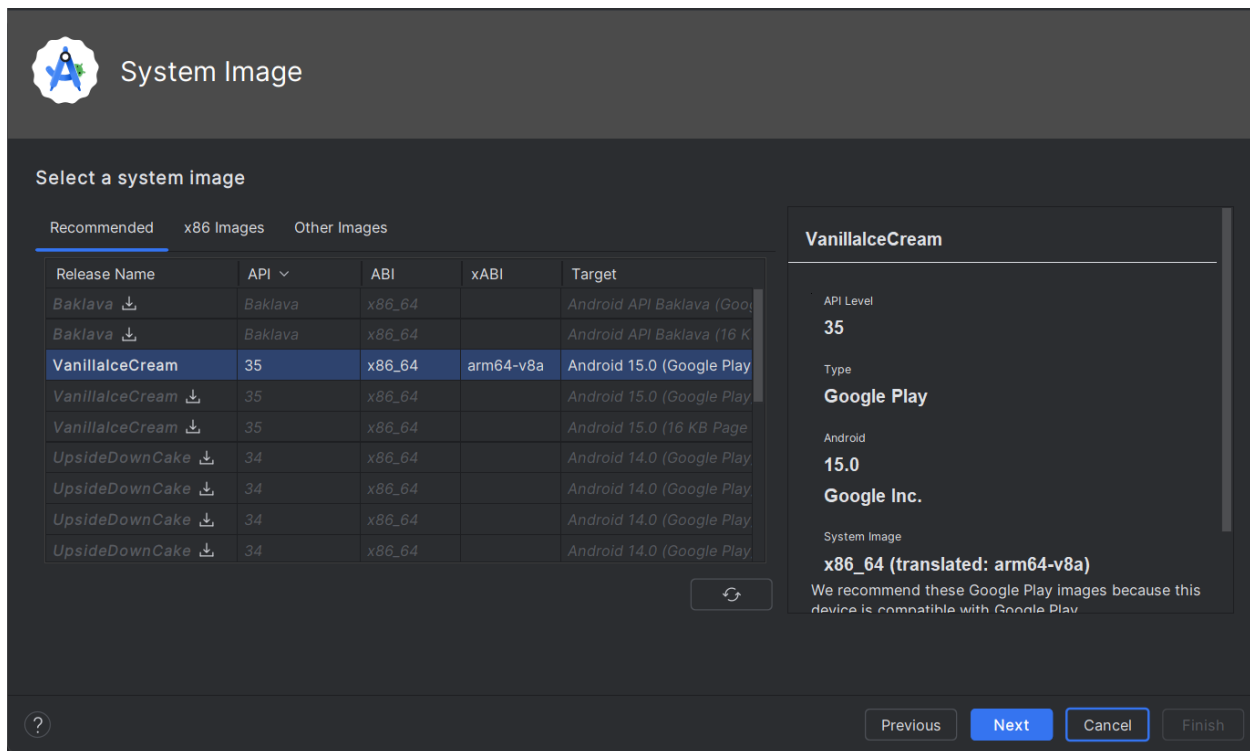
vào biểu tượng **AVD Manager**  trên thanh công cụ. Màn hình **Your Virtual Devices** xuất hiện. Nếu bạn đã tạo các thiết bị ảo trước đó, màn hình sẽ hiển thị chúng (như hình minh họa bên dưới); nếu chưa có thiết bị ảo nào, danh sách sẽ trống.



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



3. Chọn một thiết bị như Pixel 9 Pro hoặc Pixel 9 và nhấp vào Next. Màn hình System Image xuất hiện.
4. Nhấp vào tab Recommended nếu nó chưa được chọn, rồi chọn phiên bản hệ điều hành Android mà bạn muốn chạy trên thiết bị ảo (như Oreo).




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

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

- 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. Hãy kiểm tra cấu hình của bạn và nhấp **Finish**.

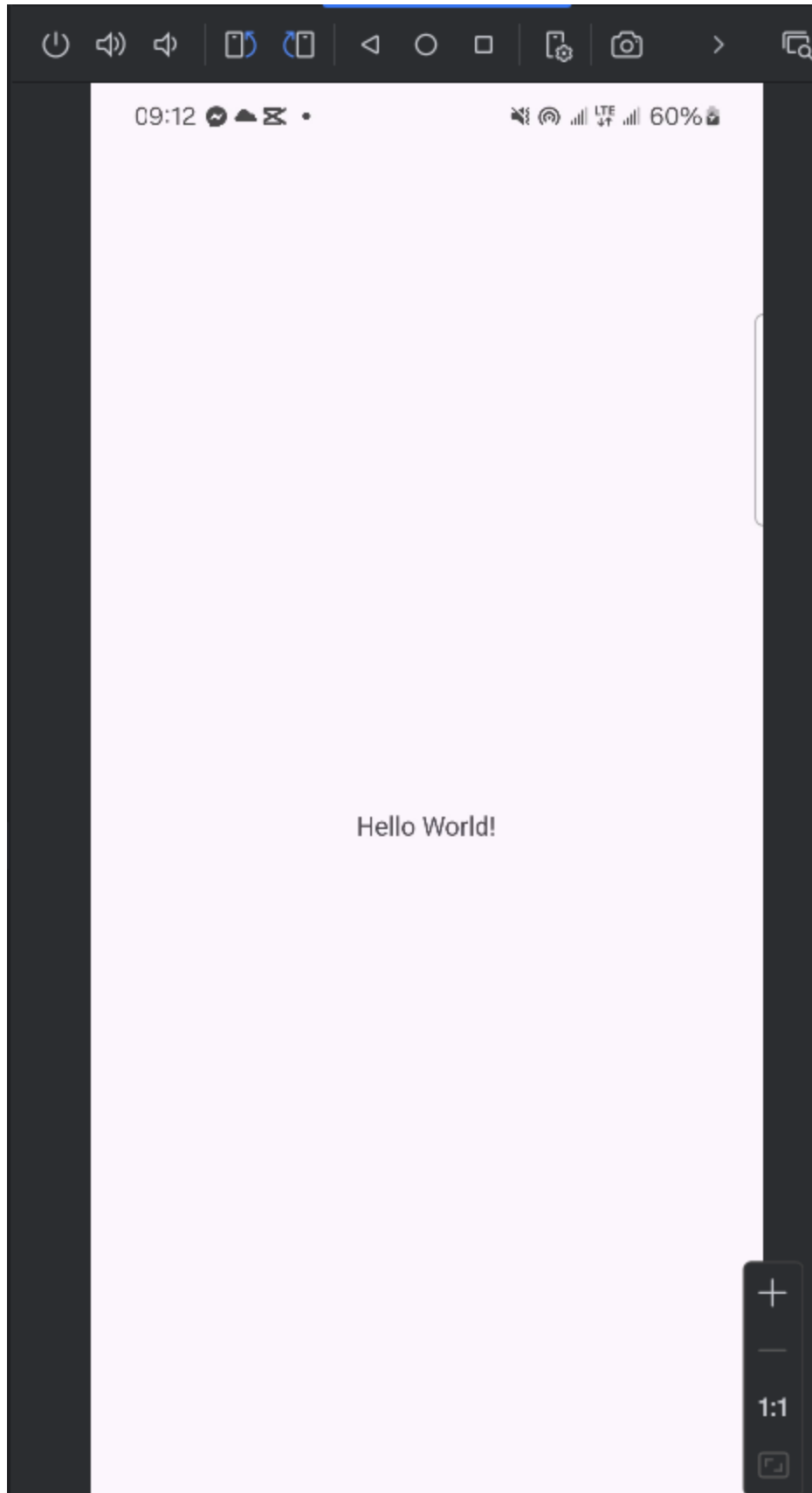
3.2 Chạy ứng dụng trên thiết bị ảo

Trong nhiệm vụ này, bạn sẽ chạy ứng dụng Hello World của mình.

- Tong **Android Studio**, chọn **Run > Run** hoặc nhấp vào biểu tượng **Run**  trên thanh công cụ.
- Trong cửa sổ **Select Deployment Target**, dưới phần **Available Virtual Devices**, chọn thiết bị ảo mà bạn vừa tạo và nhấp **OK**.

Trình giả lập sẽ khởi động và hoạt động giống như một thiết bị vật lý. Tùy thuộc vào tốc độ của máy tính, quá trình này có thể mất một chút thời gian. Ứng dụng của bạn sẽ được xây dựng và khi trình giả lập sẵn sàng, **Android Studio** sẽ tải ứng dụng lên trình giả lập và chạy nó.

Bạn sẽ thấy ứng dụng **Hello World** hiển thị như trong hình sau.



Mẹo: Khi kiểm tra trên thiết bị ảo, bạn nên khởi động nó một lần ngay từ đầu phiên làm việc. Bạn không nên đóng thiết bị ảo cho đến khi hoàn tất việc kiểm tra ứng dụng, để tránh phải chờ quá trình khởi động lại thiết bị. Để đóng thiết bị ảo, hãy nhấp vào nút **X** ở đầu trình giả lập, chọn **Thoát** từ menu hoặc nhấn **Control-Q** trên Windows hoặc **Command-Q** trên macOS.

TASK 4 : (Tùy chọn) Sử dụng thiết bị vật lý

Trong nhiệm vụ cuối cùng này, bạn sẽ chạy ứng dụng của mình trên một thiết bị di động vật lý như điện thoại hoặc máy tính bảng. Bạn nên luôn kiểm tra ứng dụng trên cả thiết bị ảo và thiết bị vật lý.

Những gì bạn cần :

- Một thiết bị Android như điện thoại hoặc máy tính bảng.
- Một cáp dữ liệu để kết nối thiết bị Android của bạn với máy tính qua cổng USB.
- Nếu bạn đang sử dụng hệ điều hành **Linux** hoặc **Windows**, bạn có thể cần thực hiện thêm một số bước để chạy ứng dụng trên thiết bị phần cứng. Hãy kiểm tra tài liệu Using Hardware Devices. Bạn cũng có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Đối với trình điều khiển USB trên Windows, hãy tham khảo mục OEM USB Drivers .

4.1 Bật gỡ lỗi USB

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