Chương 1: Khởi đầu

Bản pdf này chứa ảnh chụp nhanh một lần về các bài học trong Chương 1: Khởi đầu

Bài học trong chương này

Bài học 1: Xây dựng ứng dụng đầu tiên của bạn

1.1: Android Studio và Hello World

1.2A: Giao diện tương tác đầu tiên cuat bạn

1.2B: Chỉnh sửa bố cục

1.3: Chế độ xem văn bản và cuộn

1.4: Nguồn tài nguyên sẵn có

Bài học 2: Các hoạt động

2.1: Các hoạt động và mục tiêu

2.2: Vòng đời và trạng thái của hoạt động

2.3: Mục đích ngầm

Bài học 3: Kiểm tra, gỡ lỗi, và sử dụng thư viện hỗ trợ

3.1: Trình gỡ lỗi

3.2: Kiểm thử

3.3: Thư viện hỗ trợ

Bài học 1.1: Android Studio và Hello World

Giới thiệu

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

Những điều bạn cần biết

Bạn sẽ có thể

Hiểu được quy trình phát triển chung cho các ứng dụng hướng đối tượng bằng cách sử dụng IDE (Môi trường phát triển tích hợp) như là Android Studio.

Chứng minh là bạn phải có ít nhất 1 đến 3 năm kinh nghiệm về lập trình hướng đối tượng, trong đó một số nằm 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

- Máy tính chạy Windows hoặc Linux hoặc máy Mac chạy macOS. Xem trang tải xuống Android Studio để biết các yêu cầu hệ thống mới nhất.

- Có thể kết nối Internet hoặc một cách khác đề tài phiên bản Android Studio và Java mới nhất để vào máy tính của bạn.

Bước 1: Cài đặt Android Studio

Android Studio cung cấp môi trường phát triển tích hợp (IDE) hoàn chỉnh bao gồm một trình soạn thảo văn bản nâng cao và một bộ mẫu ứng dụng. Hơn nữa, nó bao gồm các công cụ cho lập trình, kiểm thử, gỡ lỗi và hiệu suất giúp cho việc phát triển ứng dụng trở nên dễ dàng, nhanh chóng. Bạn có thể thử nghiệm các ứng dụng của mình bằng nhiều trình giả lập được cấu hình sẵn hoặc trên thiết bị di động của riêng bạn, xây dựng các ứng dụng sản xuất và phát hành trên cửa hàng Google Play.

Ghi chú: Android Studio liên tục được cải tiến. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, vui lòng xem Android Studio

ndroid Studio có sẵn cho máy tính chạy Windows hoặc Linux, và cho Mac chạy macOS. Phiên bản mới nhất OpenJDK (Java Development Kit) được tích hợp với Android Studio.

Để bắt đầu và chạy Android Studio, trước tiên hãy kiểm tra các [yêu cầu hệ thống](https://developer.android.com/studio) để đảm bảo hệ thống của bạn đáp ứng các điều kiện. Quá trình cài đặt tương tự trên tất cả các nền tảng. Các điểm khác biệt được ghi chú bên dưới.

Điều hướng đến trang [Android developers](https://developer.android.com/studio) và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.

Chấp nhận cấu hình mặc định cho tất cả các bước và đảm bảo rằng tất cả các thành phần đều được chọn để cài đặt.

Sau khi hoàn tất cài đặt, Setup Wizard sẽ tải xuống và cài đặt thêm một số thành phần bổ sung, bao gồm Android SDK. Hãy kiên nhẫn, quá trình này có thể mất thời gian tùy thuộc vào tốc độ Internet của bạn, và một số bước có thể có vẻ dư thừa.

Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động và bạn đã sẵn sàng để tạo dự án đầu tiên của mình.

Ghi chú: Nếu bạn gặp sự cố khi cài đặt, hãy kiểm tra [Android Studio release notes](https://developer.android.com/studio/releases) hoặc nhờ giảng viên của bạn giúp đỡ.

Bước 2: Tạo chương trình Hello World

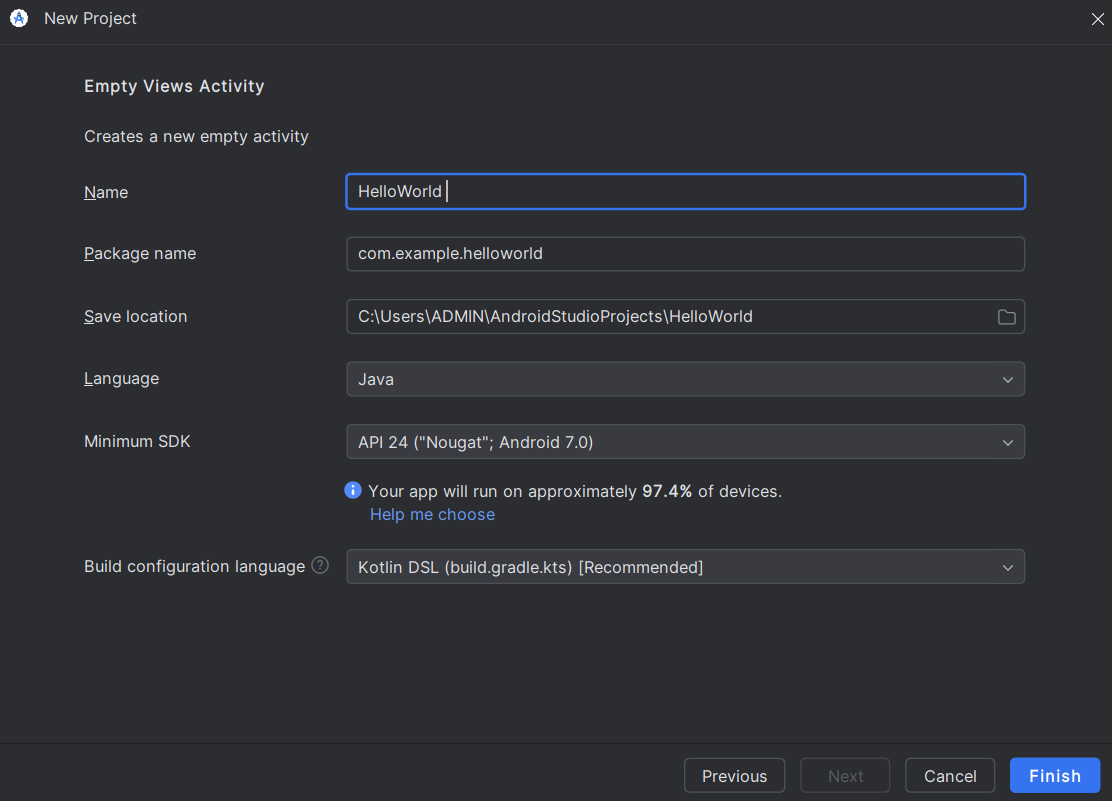
Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị "Hello World" để xác nhận rằng Android Studio đã được cài đặt đúng cách và để tìm hiểu những điều cơ bản về phát triển ứng dụng với Android Studio.

2.1 Create the app project

1. Mở Android Studio nếu nó chưa được mở.

2. Trong cửa sổ Welcome to Android Studio, nhấp vào Start a new Android Studio project.

3. Trong cửa sổ Create Android Project, nhập Hello World cho Application name.

4. trên màn hình thiết bị android target, phone và tablet sẽ được chọn. đảm bảo rằng api 15: android 4.0.3 icecreamsandwich được đặt làm sdk tối thiểu; nếu không, hãy sử dụng menu bật lên để đặt nó.

đây là các cài đặt được sử dụng bởi các ví dụ trong các bài học cho khóa học này. khi viết bài này, các cài đặt này làm cho ứng dụng hello world của bạn tương thích với 97% thiết bị android đang hoạt động trên cửa hàng google play.

bỏ chọn hỗ trợ ứng dụng tức thì và tất cả các tùy chọn khác. sau đó nhấp vào n ext. nếu dự án của bạn yêu cầu các thành phần bổ sung cho sdk mục tiêu bạn đã chọn, android studio sẽ tự động cài đặt các thành phần đó.

thêm hoạt động w indow xuất hiện. activity là một điều duy nhất, tập trung mà người dùng có thể làm. nó là một thành phần quan trọng của bất kỳ ứng dụng android nào. activity thường có một bố cục được liên kết với nó xác định cách các phần tử giao diện người dùng xuất hiện trên màn hình. android studio cung cấp

mẫu hoạt động để giúp bạn bắt đầu. đối với dự án hello world, hãy chọn empty activity như hình dưới đây và nhấp vào next.

A screenshot of a computer

AI-generated content may be incorrect.

Màn hình hoạt động Configure xuất hiện (khác nhau tùy thuộc vào mẫu bạn đã chọn ở bước trước). Theo mặc định, Activity trống được cung cấp bởi mẫu được đặt tên là MainActivity. Bạn có thể thay đổi điều này nếu muốn, nhưng bài học này sử dụng MainActivity.

Đảm bảo rằng **Generate Layout file** tùy chọn được chọn. Tên bố cục theo mặc định là activity main. Bạn có thể thay đổi điều này nếu muốn, nhưng bài học này sử dụng ctivity\_main.

Đảm bảo rằng B**ackwards Compatibility (App Compat)** option được kiểm tra. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước đó.

Nhấp vào 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**: Xem C trên trang nhà phát triển bản dựng của bạn để biết thông tin chi tiết.

Bạn cũng có thể thấy thông báo "Mẹo trong ngày" với các phím tắt và các mẹo hữu ích khác. Nhấp vào Đóng **để** đóng thư.

Trình chỉnh sửa Android Studio sẽ xuất hiện. Làm theo các bước sau:

Nhấp vào **tab activity\_main.xml** để xem trình chỉnh sửa bố cục.

Nhấp vào tab Design **của trình soạn thảo bố cục**, nếu chưa được chọn, để hiển thị hình ảnh của bố cục như hình dưới đây.

Nhấp vào **tab MainActivity.java** để xem trình chỉnh sửa mã như hình bên dưới.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

2.2 Khám phá khung Project > Android

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

Nếu chưa được chọn, hãy nhấp vào tab **P roject** trong cột tab dọc ở phía bên trái của

Cửa sổ Android Studio. Ngăn Dự án xuất hiện.

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 ngăn Dự án, khi được đặt thành **A ndroid**, là ngăn **Dự án > Android**.

A screenshot of a computer

AI-generated content may be incorrect.

2.3 Khám phá thư mục Tập lệnh Gradle

Hệ thống bản dựng Gradle trong Android Studio giúp bạn dễ dàng đưa các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng dưới dạng phần phụ thuộc.

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

A screenshot of a computer

AI-generated content may be incorrect.

Làm theo các bước sau để khám phá hệ thống Gradle:

Nếu thư mục **Tập lệnh Gradle** không được mở rộng, hãy nhấp vào hình tam giác để mở rộng thư mục đó. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng.

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 tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một tệp bản dựng Gradle cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần thực hiện bất kỳ thay đổi nào đối với tệp này, nhưng vẫn hữu ích khi hiểu nội dung của nó.

Theo mặc định, tệp bản dựng cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ Gradle và các phần phụ thuộc chung cho tất cả các mô-đun trong dự án. Khi phần 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 kho lưu trữ của tệp này. Theo mặc định, các dự án Android Studio mới khai báo JCenter và Google (bao gồm kho lưu trữ G oogle Maven) là vị trí kho lưu trữ:

allprojects {repositories {

google)

jcenter)

}

}

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

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

Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phần phụ thuộc trong phần d ependencies. Bạn có thể khai báo phần phụ thuộc thư viện bằng cách sử dụng một trong một số cấu hình phần phụ thuộc khác nhau. Mỗi cấu hình phần phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ: câu lệnh i mplementation fileTree(dir: 'libs', include: ['\*.jar']) thêm một

sự phụ thuộc của tất cả các tệp ".jar" bên trong thư mục l ibs.

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

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

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

Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phần phụ thuộc trong phần d ependencies. Bạn có thể khai báo phần phụ thuộc thư viện bằng cách sử dụng một trong một số cấu hình phần phụ thuộc khác nhau. Mỗi cấu hình phần 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

sự phụ thuộc của tất cả các tệp ".jar" bên trong thư mục l ibs.

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

A screen shot of a computer

AI-generated content may be incorrect.

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

2.4 Khám phá ứng dụng và thư mục res

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

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 tệp đó trong trình soạn thảo mã.

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

Mở rộng **thư mục** res và thư mục **layout**, và nhấp đúp vào tệp **a ctivity\_main.xml** để mở nó trong trình chỉnh sửa bố cục.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Thư **mục res** chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh. Hoạt động thường được liên kết với bố cục của các chế độ xem giao diện người dùng được xác định dưới dạng tệp XML. Tệp này thường được đặt tên theo Hoạt động của nó.

2.5 Khám phá thư mục tệp kê khai

Thư mục manifests chứa các tệp cung cấp thông tin cần thiết về ứng dụng của bạn cho

Hệ thống Android, hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng.

Mở rộng thư mục **manifests**.

Mở tệp **AndroidManifest.xml**.

Tệp AndroidManifest.xml mô tả tất cả các thành phần của ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như mỗi thành phần A, phải được khai báo trong tệp XML này. Trong các bài học khóa học khác, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để biết phần giới thiệu, hãy xem Tổng quan về bản kê khai pp.

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

Trong tác vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo (AVD) để tạo một thiết bị ảo (còn được gọi là trình mô phỏng) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Xin lưu ý rằng Trình mô phỏng Android có các yêu cầu riêng ngoài các yêu cầu hệ thống cơ bản cho Android Studio.

Khi sử dụng Trình quản lý AVD, bạn xác định các đặc điểm phần cứng của thiết bị, cấp độ API, bộ nhớ, giao diện và các thuộc tính khác và lưu thiết bị đó dưới dạng thiết bị ảo. Với thiết bị ảo, bạn có thể kiểm tra ứng dụng trên các cấu hình thiết bị khác nhau (chẳng hạn như máy tính bảng và điện thoại) với các cấp độ API khác nhau mà không cần phải sử dụng thiết bị thực.

1. Trong Android Studio, hãy chọn T**ools > Trình quản lý AVD > Android** hoặc nhấp vào biểu tượng Trình quản lý AVD

 trên thanh công cụ. Y **thiết bị ảo của chúng tôi** xuất hiện. Nếu bạn đã tạo

thiết bị ảo, màn hình hiển thị chúng (như trong hình bên dưới); nếu không, bạn sẽ thấy một danh sách trống.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Nhấp vào nút + **Tạo thiết bị ảo**. Cửa sổ **Select Hardware** xuất hiện hiển thị danh sách các thiết bị phần cứng được định 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**).
2. Chọn một thiết bị như **Nexus 5x** hoặc **Pixel XL,** và nhấp vào **Next**. S**ystem Image** screen xuất hiện.
3. Nhấp vào **tab Đề xuất** nếu nó chưa được chọn và chọn phiên bản hệ thống Android để chạy trên thiết bị ảo (chẳng hạn như **Oreo**).

Có nhiều phiên bản có sẵn hơn so với hiển thị trong tab **R khuyến nghị**. Nhìn vào các tab **x 86 Images** và **Other Images** để xem chúng.

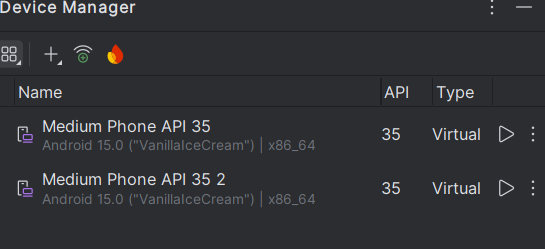
Nếu liên **kết Download** hiển thị bên cạnh hình ảnh hệ thống bạn muốn sử dụng, nó vẫn chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào **Finish** khi hoàn tất.

1. Sau khi chọn hình ảnh hệ thống, hãy nhấp vào **Next**. **Cửa sổ Thiết bị ảo ndroid (AVD)** xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào **Finish**.

3.2 Chạy ứng dụng trên thiết bị ảo

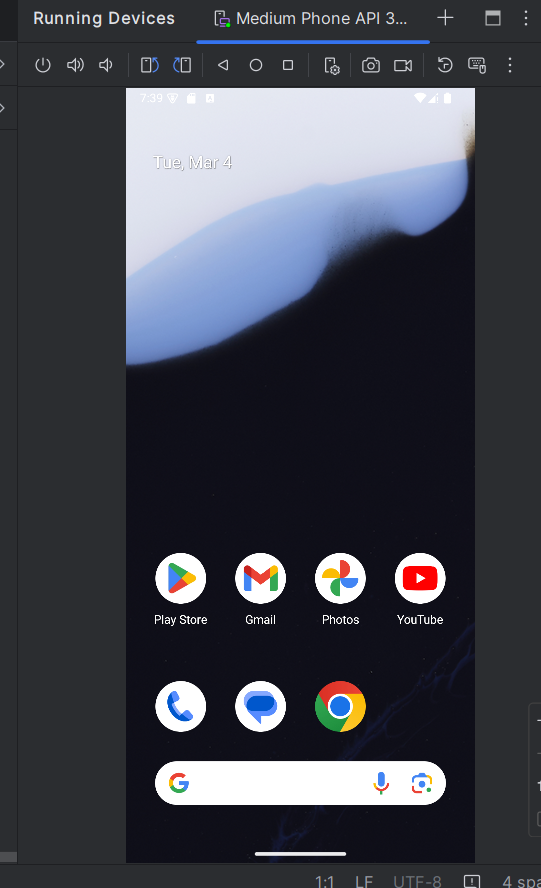
Trong nhiệm vụ này, cuối cùng bạn sẽ chạy ứng dụng Hello World của mình.

1. Trong Android Studio, hãy chọn **ứng dụng Run > Run** hoặc nhấp vào **biểu tượng** Run trên thanh công cụ.
2. Cửa **sổ Select Deployment Target**, trong **Available Virtual Devices**, chọn thiết bị ảo mà bạn vừa tạo và nhấp vào **OK**.



Trình giả lập khởi động và khởi động giống như một thiết bị vật lý. Tùy thuộc vào tốc độ máy tính của bạn, quá trình này có thể mất một lúc. Ứng dụng của bạn sẽ được xây dựng và sau khi trình mô phỏng đã sẵn sàng, Android Studio sẽ tải ứng dụng lên trình mô phỏng và chạy ứng dụng đó.

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



**Mẹo**: Khi thử nghiệm trên thiết bị ảo, bạn nên khởi động thiết bị một lần, ngay từ đầu phiên của bạn. Bạn không nên đóng ứng dụng cho đến khi hoàn tất việc kiểm tra ứng dụng của mình để ứng dụng của bạn không phải trải qua quá trình khởi động thiết bị một lần nữa. Để đóng thiết bị ảo, hãy nhấp vào nút **X** tại

trên đầu trình giả lập, chọn **Quit** từ menu hoặc nhấn **Control-Q** trong Windows hoặc **Command-Q** in macOS.

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

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

Thiết bị Android như điện thoại hoặc máy tính bảng.

Cáp dữ liệu để kết nối thiết bị Android với máy tính qua cổng USB.

Nếu bạn đang sử dụng hệ thống Linux hoặc Windows, bạn có thể cần thực hiện các bước bổ sung để chạy trên thiết bị phần cứng. Kiểm tra tài liệu về Thiết bị phần cứng U sing. Bạn cũng có thể cần cài đặt trình điều khiển USB thích hợp cho thiết bị của mình. Đối với trình điều khiển USB dựa trên Windows, hãy xem Trình điều khiển USB OEM.

4.1 Bật gỡ lỗi USB

Để cho phép Android Studio giao tiếp với thiết bị của bạn, bạn phải bật tính năng Gỡ lỗi USB trên thiết bị Android của mình. Tính năng này được bật trong **cài đặt tùy chọn Developer** trên thiết bị của bạn.

Trên Android 4.2 trở lên, màn hình **tùy chọn Developer** bị ẩn theo mặc định. Để hiển thị các tùy chọn dành cho nhà phát triển và bật Gỡ lỗi USB:

* 1. Trên thiết bị của bạn, hãy mở **S ettings**, tìm kiếm Giới **thiệu phone**,

nhấp vào **About phone** và nhấn vào **Số xây dựng** bảy lần.

* 1. Quay lại màn hình trước đó (**Settings / System**). **Các tùy chọn Developer** xuất hiện trong danh sách. Nhấn vào **Tùy chọn Developer**.

1. Chọn **gỡ lỗi USB**.

4.2 Chạy ứng dụng của bạn trên thiết bị

Giờ đây, bạn có thể kết nối thiết bị của mình và chạy ứng dụng từ Android Studio.

Kết nối thiết bị của bạn với máy phát triển bằng cáp USB.

Nhấp vào  **nút** Run trên thanh công cụ. Cửa sổ **Select Deployment Target** mở ra với danh sách các trình giả lập có sẵn và các thiết bị được kết nối.

Chọn thiết bị của bạn và nhấp vào **OK**.

Android Studio cài đặt và chạy ứng dụng trên thiết bị của bạn.

Troubleshooting

Nếu Android Studio không nhận dạng thiết bị của bạn, hãy thử các cách sau:

Rút phích cắm và cắm lại thiết bị của bạn.

Khởi động lại Android Studio.

Nếu máy tính của bạn vẫn không tìm thấy thiết bị hoặc tuyên bố thiết bị là "không được phép", hãy làm theo các bước sau:

Rút phích cắm của thiết bị.

Trên thiết bị, mở **Tùy chọn Developer trong ứng dụng Cài đặt**.

Nhấn vào Thu hồi **ủy quyền** gỡ lỗi USB.

Kết nối lại thiết bị với máy tính của bạn.

Khi được nhắc, hãy cấp ủy quyền.

Nhiệm vụ 5: Thay đổi cấu hình Gradle của ứng dụng

Trong nhiệm vụ này, bạn sẽ thay đổi điều gì đó về cấu hình ứng dụng trong

build.gradle(Module:app) để tìm hiểu cách thực hiện các thay đổi và đồng bộ hóa chúng với dự án Android Studio của bạn.

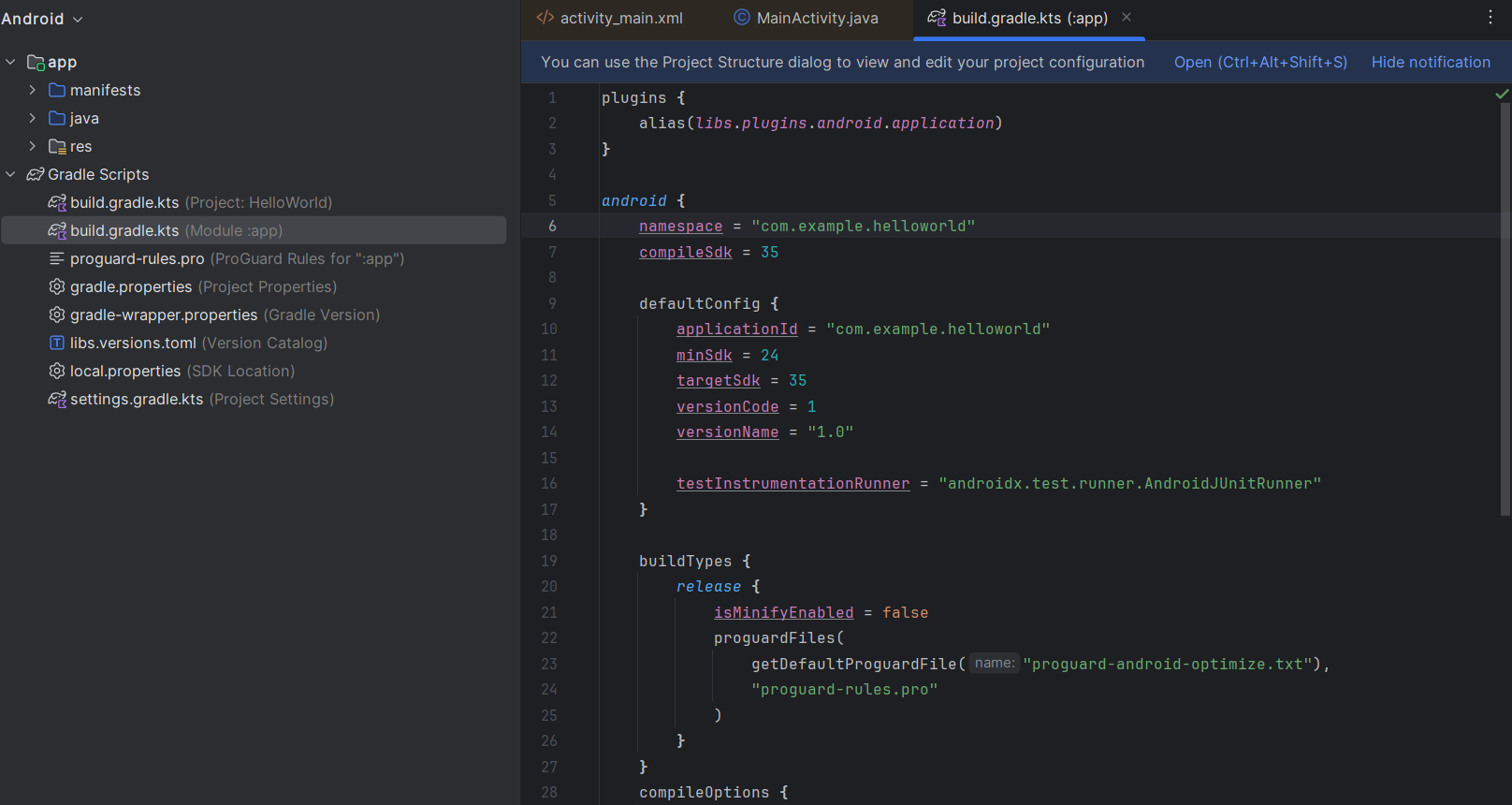
5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng

Làm theo các bước sau:

Mở rộng thư mục **Gradle Scripts** nếu nó chưa được mở và nhấp đúp vào tập tin **build.gradle(Module:app).**

Nội dung của tệp xuất hiện trong trình soạn thảo mã.

Trong khối d efaultConfig , thay đổi giá trị của m inSdkVersion thành 1 7 như hình bên dưới (ban đầu nó được đặt thành 1 5).



Trình soạn thảo mã hiển thị thanh thông báo ở trên cùng với  **liên kết** Sync Now.