

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

3

| | | |
|--------|---|----|
| Bài 1) | Tạo ứng dụng đầu tiên | 3 |
| 1.1) | Android Studio và Hello World | 3 |
| 1.2) | Giao diện người dùng tương tác đầu tiên..... | 34 |
| 1.3) | Trình chỉnh sửa bố cục | 53 |
| 1.4) | Văn bản và các chế độ cuộn..... | 53 |
| 1.5) | Tài nguyên có sẵn..... | 53 |
| Bài 2) | Activities | 53 |
| 2.1) | Activity và Intent..... | 53 |
| 2.2) | Vòng đời của Activity và trạng thái..... | 53 |
| 2.3) | Intent ngầm định | 53 |
| Bài 3) | Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ | 53 |
| 3.1) | Trình gỡ lỗi..... | 53 |
| 3.2) | Kiểm thử đơn vị | 53 |
| 3.3) | Thư viện hỗ trợ | 53 |

CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG

54

| | | |
|--------|-------------------------------------|----|
| Bài 1) | Tương tác người dùng | 54 |
| 1.1) | Hình ảnh có thể chọn..... | 54 |
| 1.2) | Các điều khiển nhập liệu..... | 54 |
| 1.3) | Menu và bộ chọn | 54 |
| 1.4) | Điều hướng người dùng | 54 |
| 1.5) | RecyclerView..... | 54 |
| Bài 2) | Trải nghiệm người dùng thú vị | 54 |
| 2.1) | Hình vẽ, định kiểu và chủ đề..... | 54 |
| 2.2) | Thẻ và màu sắc | 54 |

| | |
|---|-----------|
| 2.3) Bố cục thích ứng..... | 54 |
| Bài 3) Kiểm thử giao diện người dùng..... | 54 |
| 3.1) Espresso cho việc kiểm tra UI | 54 |
| CHƯƠNG 3. LÀM VIỆC TRONG NỀN | 54 |
| Bài 1) Các tác vụ nền..... | 54 |
| 1.1) AsyncTask..... | 54 |
| 1.2) AsyncTask và AsyncTaskLoader | 54 |
| 1.3) Broadcast receivers | 54 |
| Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền | 54 |
| 2.1) Thông báo | 54 |
| 2.2) Trình quản lý cảnh báo | 54 |
| 2.3) JobScheduler | 54 |
| CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG | 55 |
| Bài 1) Tùy chọn và cài đặt | 55 |
| 1.1) Shared preferences | 55 |
| 1.2) Cài đặt ứng dụng | 55 |
| Bài 2) Lưu trữ dữ liệu với Room | 55 |
| 2.1) Room, LiveData và ViewModel..... | 55 |
| 2.2) Room, LiveData và ViewModel..... | 55 |

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

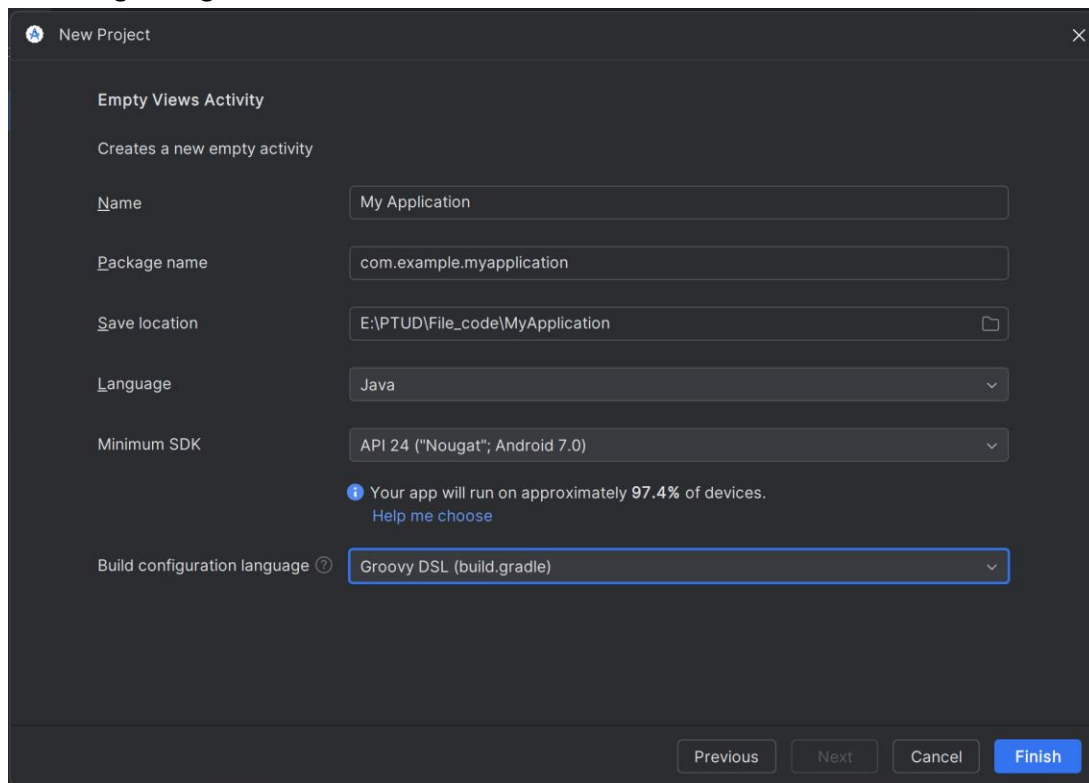
Giới thiệu

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 sẽ có thể:

- 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.

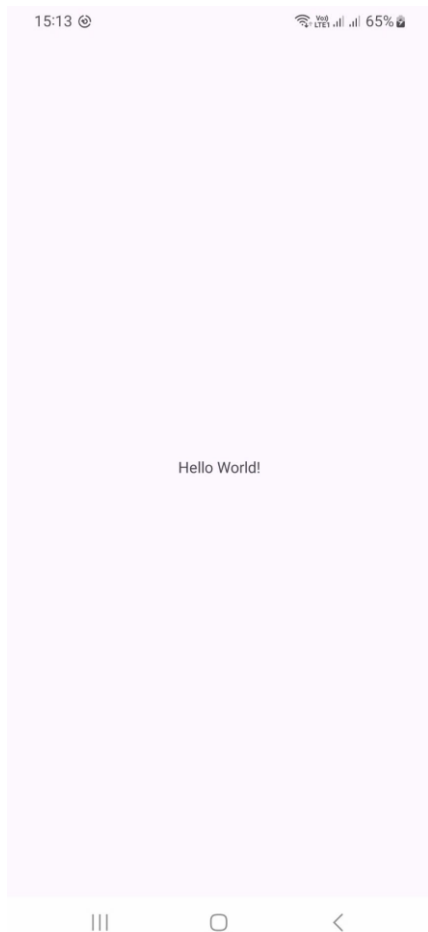
Những gì bạn sẽ làm

- 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, 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 ảo hoặc thiết bị thật Android

Đây là ứng dụng đã hoàn thành sẽ như thế này:



Nhiệm vụ 1: Cài đặt Android Studio

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

Android Studio có sẵn cho máy tính chạy Windows hoặc Linux, và cho máy Mac chạy macOS. Bộ phát triển phần mềm Java (JDK) mới nhất được tích hợp sẵn với Android Studio.

Để bắt đầu sử dụng Android Studio, trước tiên kiểm tra các yêu cầu hệ thống để đảm bảo hệ thống của bạn đáp ứng được chúng. Quá trình cài đặt tương tự nhau trên tất cả các nền tảng. Bất kỳ sự khác biệt nào sẽ được ghi chú bên dưới.

1. Điều hướng đến trang dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio

2. 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

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 cả Android SDK. Hãy kiên nhẫn, điều này có thể mất một chút thời gian tùy thuộc vào tốc độ Internet của bạn và một số bước có thể thừa

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

Nhiệm vụ 2: Tạo ứng dụng Hello World

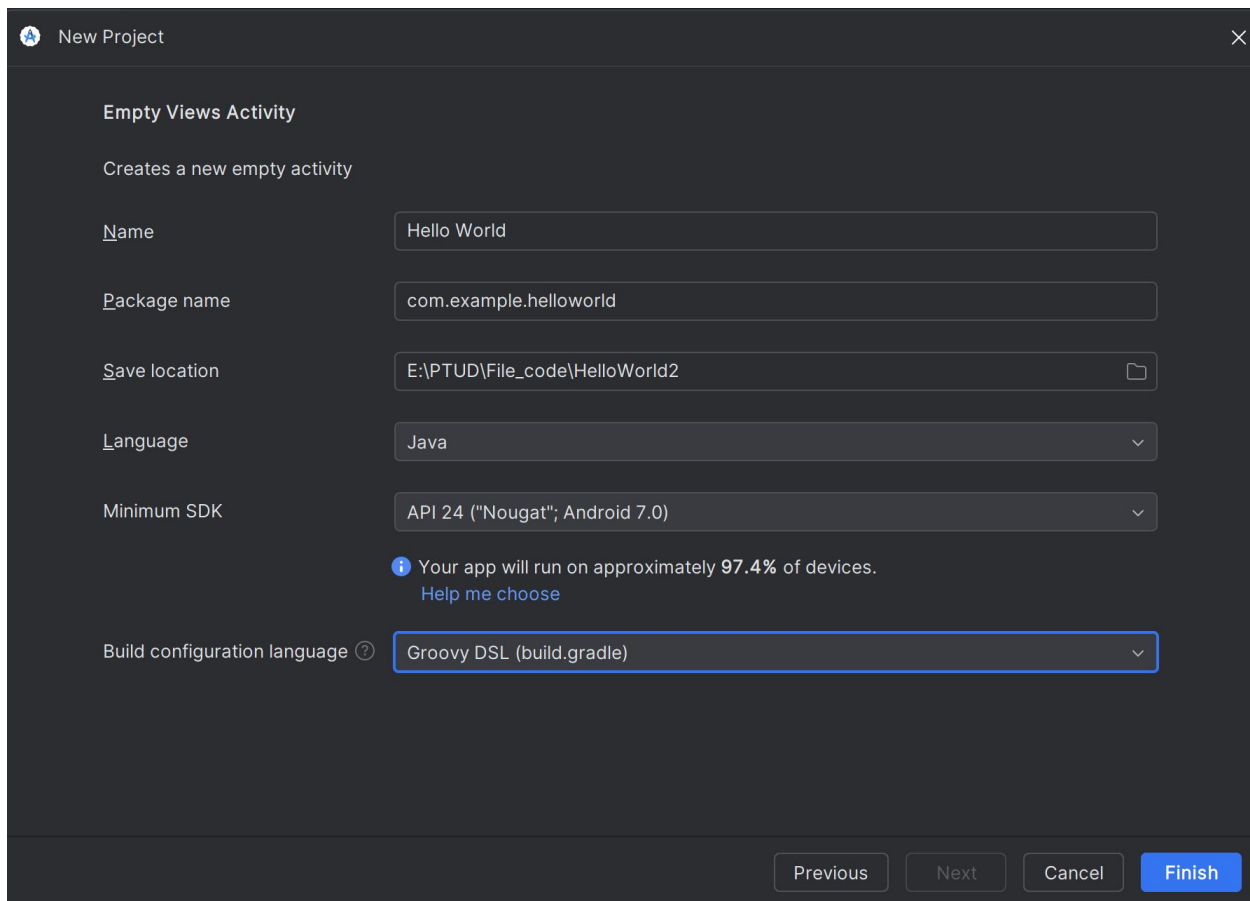
Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị “Hello World” để xác nhận rằng Android Studio đã được cài đặt chính xác và để tìm hiểu những điều cơ bản của việc phát triển với Android Studio

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu chưa mở

2. Trong cửa sổ chính chào mừng đến với Android Studio, nhấp vào bắt đầu dự án Android Studio mới

3. Trong cửa sổ tạo dự án Android, nhập Hello World cho tên ứng dụng



4. Xác nhận vị trí dự án mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác hoặc thay đổi nó thành thư mục ưa thích của mình

5. Chấp nhận `Android.example.com` cho miền công ty hoặc tạo một miền công ty duy nhất.

Nếu bạn không có kế hoạch xuất 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.

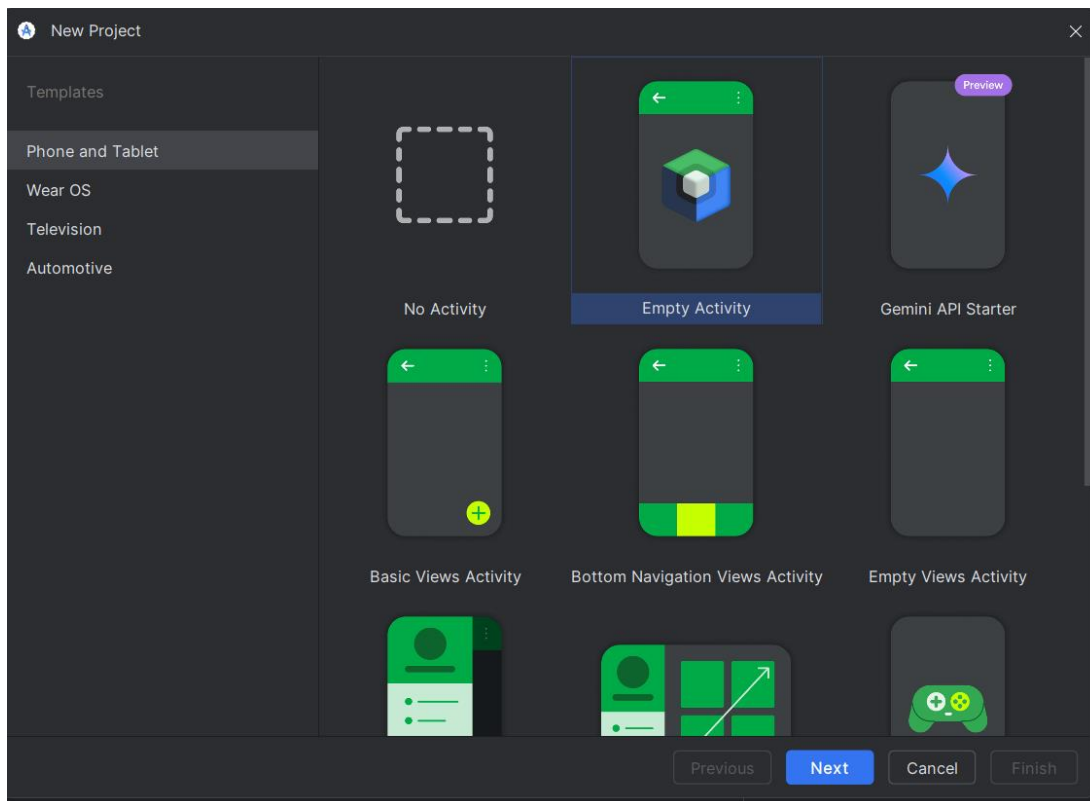
6. Bỏ chọn các tùy chọn bao gồm hỗ trợ C++ và bao gồm hỗ trợ Kotlin, và nhấp vào Tiếp theo

7. Trên màn hình Target Android thiết bị, điện thoại và máy tính bảng nên 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

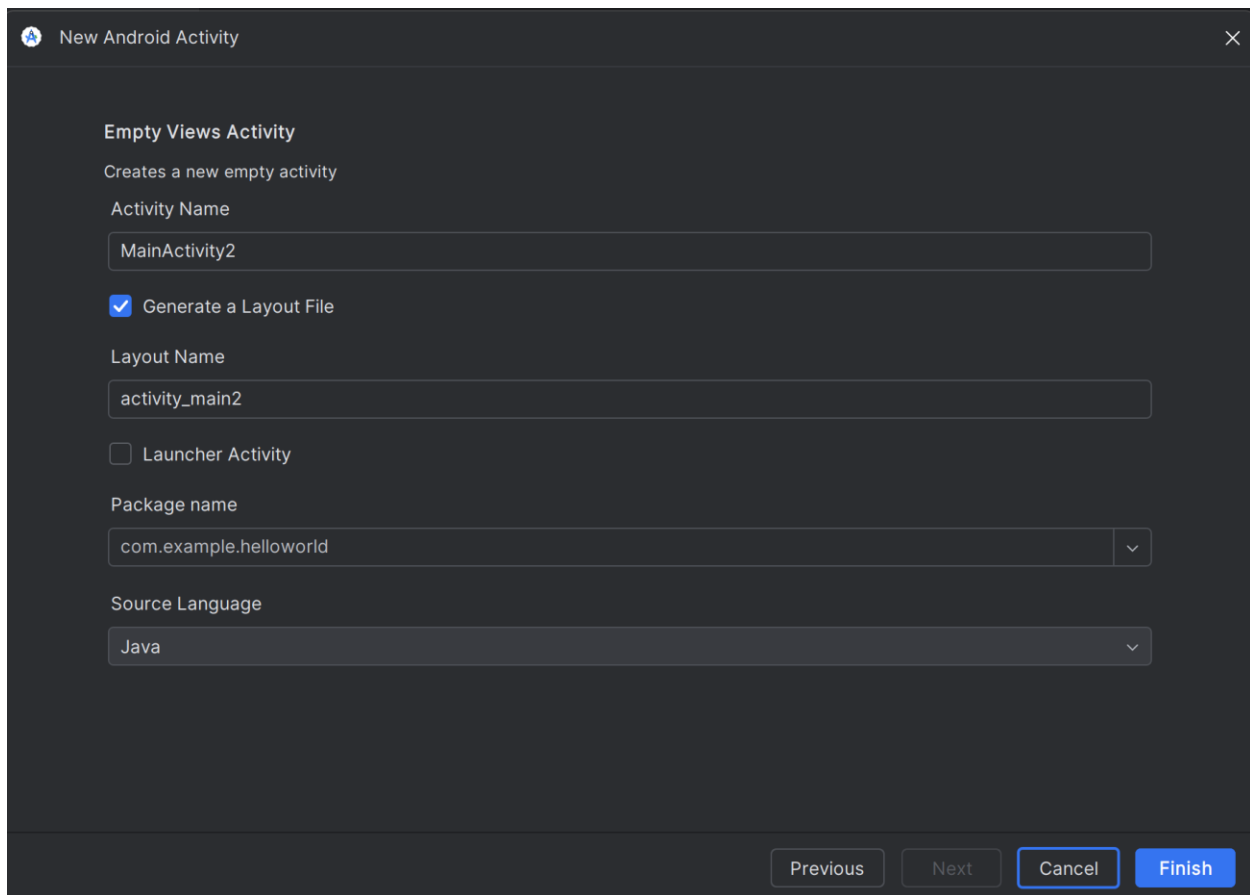
Đây là các cài đặt được sử dụng bởi các ví dụ trong các bài học cho khóa học này. Theo bài này, các cài đặt này làm cho ứng dụng Hello World của bạn tương thích với 97% thiết bị Android hoạt động trên Google Play Store.

8. Bỏ chọn hỗ trợ bao gồm ứng dụng tức thì và tất cả các tùy chọn khác. Sau đó nhấp vào Tiếp theo. Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK mục tiêu đã chọn, Android Studio sẽ tự động cài đặt chúng

9. Cửa sổ Thêm hoạt động xuất hiện. Hoạt động là một việc tập trung duy nhất mà người dùng có thể làm. Nó là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một hoạt động thường có bố cục được liên kết với nó xác định cách các thành phần giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp các mẫu Hoạt động để giúp bạn bắt đầu. Đối với dự án Hello World, chọn Empty Activity như hình dưới đây và nhấp vào Next



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



11. Đảm bảo rằng tùy chọn **Generate Layout** file được chọn. Tên layout mặc định là `activity_main`. Bạn có thể thay đổi nếu muốn, nhưng bài học này sử dụng `activity_main`.

12. Đảm bảo rằng tùy chọn **Tương thích ngược (Tương thích ứng dụng)** được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước đó.

13. Nhấp vào Kết thúc.

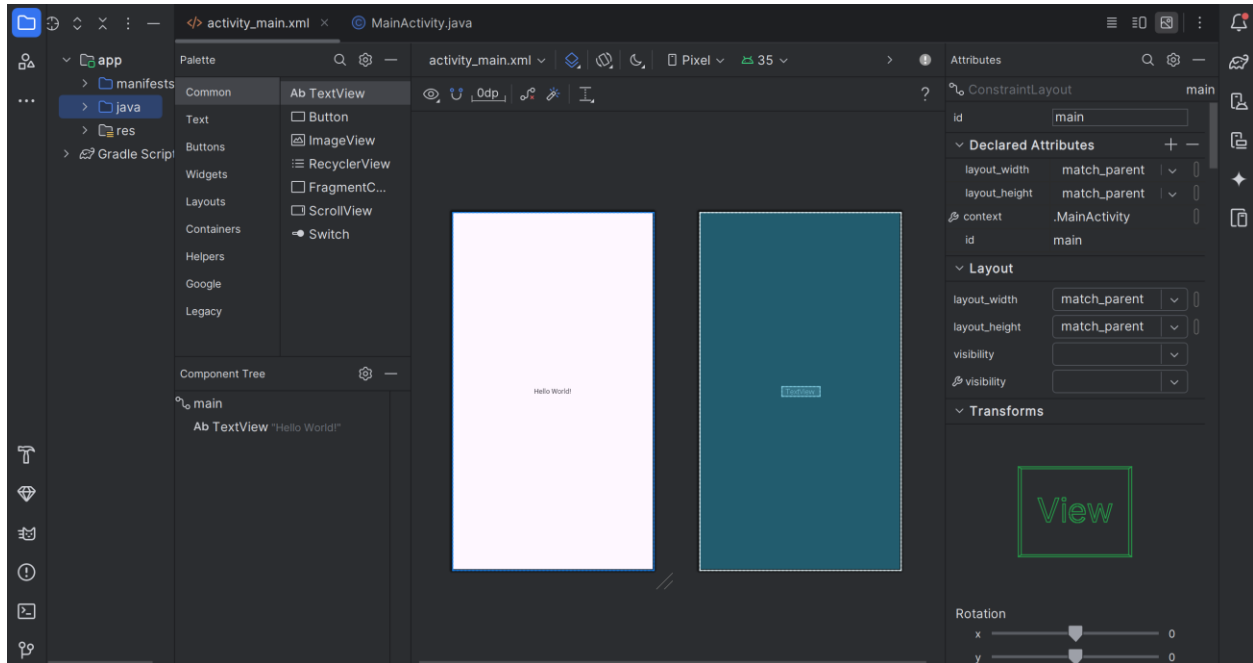
Android Studio tạo một thư mục cho các dự án của bạn và xây dựng dự án bằng Gradle (có thể mất vài phút).

Mẹo: Xem trang Cấu hình nhà phát triển bản dựng của bạn để biết thông tin chi tiết.

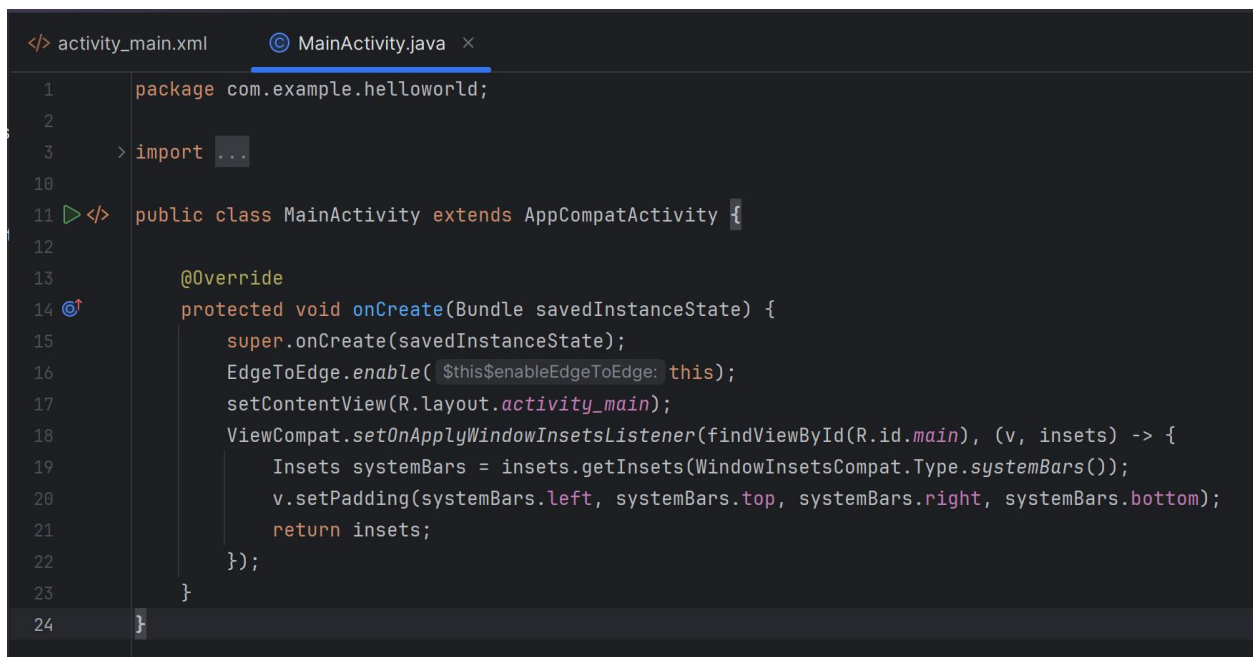
Bạn cũng có thể thấy thông báo “Mẹo trong ngày” với các phím tắt và các mẹo hữu ích khác. Nhấp vào Đóng để đóng thông báo.

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

1. Nhấp vào tab **activity_main.xml** để xem trình chỉnh sửa bố cục
2. Nhấp vào tab **Thiết kế** của trình chỉnh sửa bố cục , nếu chưa được chọn, để hiển thị bản đồ họa của bố cục như hình dưới đây



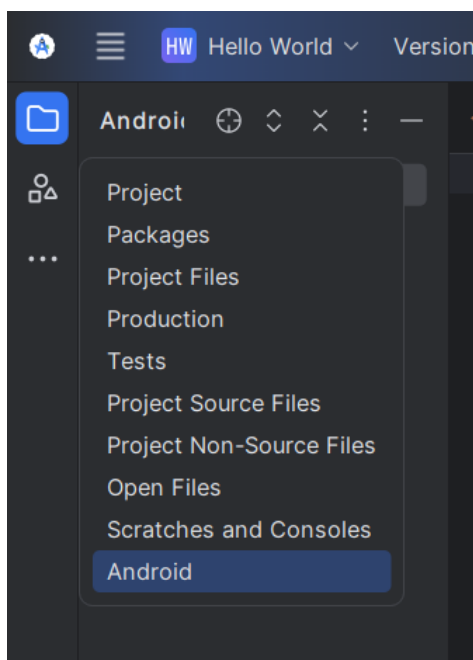
3. Nhấp vào tab **MainActivity.java** để xem trình soạn thảo mã như hiển thị bên dưới



2.2 Khám phá ngăn Project > Android

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

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

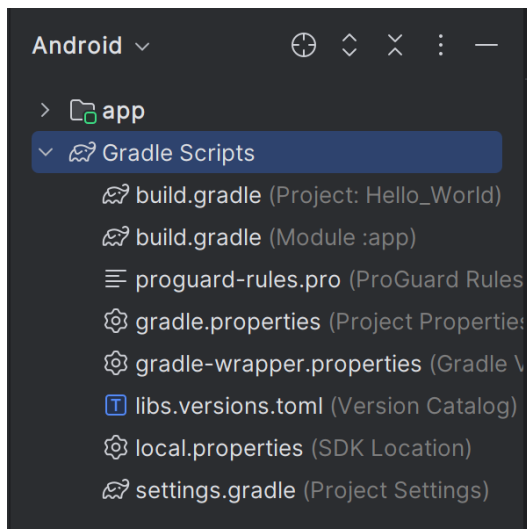


Lưu ý: Chương này và các chương khác đề cập đến ngăn Dự án khi được đặt thành Android là ngăn Dự án > Android

2.3 Khám phá thư mục Gradle Scripts

Hệ thống xây 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 phụ thuộc

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



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

1. Nếu thư mục **Gradle Scripts** không được mở rộng, hãy nhấp vào hình tam giác để mở rộng. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây 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 tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một tệp 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 tệp.

Theo mặc định, tệp dựng cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ Gradle và các phụ thuộc chung cho tất cả các mô-đun 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 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 Google Maven](#)) là các vị trí kho lưu trữ:

```
plugins {  
    alias(libs.plugins.android.application) apply false  
}
```

3. Tìm tệp build.gradle (Module:app)

Ngoài tệp cấp dự án build.gradle, mỗi mô-đun đều có build.gradle tệp 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 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 cài đặt trong AndroidManifest.xml tệp hoặc cấp cao nhất build.gradle file.

Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phụ thuộc trong dependencies phần. Bạn có thể khai báo phụ thuộc thư viện bằng một trong một số 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 thêm phụ thuộc của tất cả các tệp “.jar” bên trong thư mục libs. implementation fileTree(dir: 'libs', include: ['*.jar'])

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

```

plugins {
    alias(libs.plugins.android.application)
}
android {
    namespace 'com.example.helloworld'
    compileSdk 35
    defaultConfig { DefaultConfig it ->
        applicationId "com.example.helloworld"
        minSdk 24
        targetSdk 35
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes { NamedDomainObjectContainer<BuildType> it ->
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions { CompileOptions it ->
        sourceCompatibility JavaVersion.VERSION_11
        targetCompatibility JavaVersion.VERSION_11
    }
}
dependencies {
    implementation libs.appcompat
    implementation libs.material
    implementation libs.activity
    implementation libs.constraintlayout
    testImplementation libs.junit
    androidTestImplementation libs.ext.junit
    androidTestImplementation libs.espresso.core
}

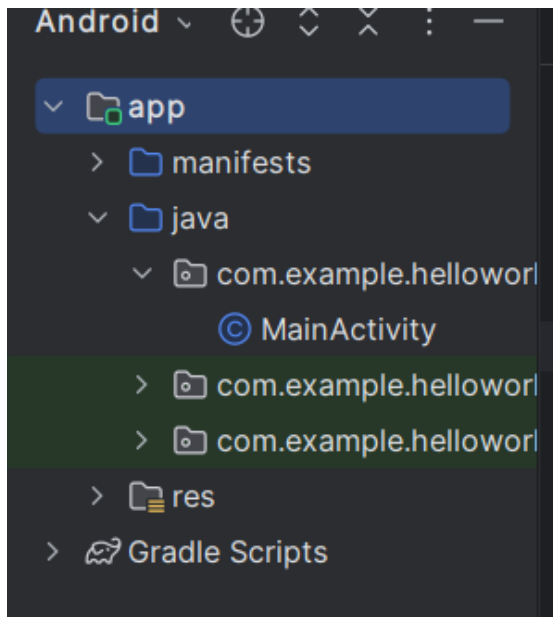
```

4. Nhấp vào hình tam giác để đóng Gradle Scripts

2.4 Khám phá các thư mục ứng dụng và res

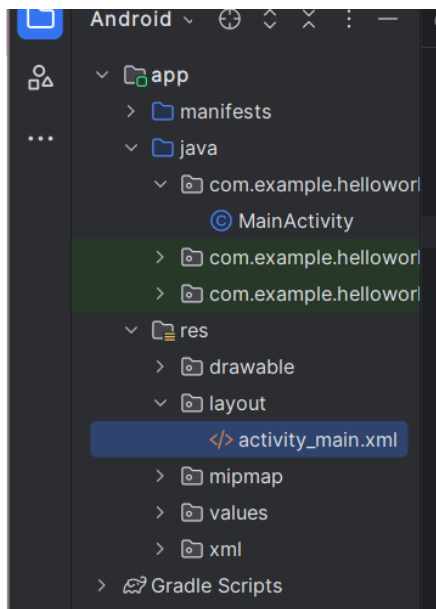
Tất cả mã và tài nguyên cho ứng dụng đều nằm trong các thư mục ứng dụng và res

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 để mở tệp trong trình soạn thảo mã



Thư mục **java** bao gồm các tệp lớp Java trong ba thư mục con, như thể hiện 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 khác được sử dụng để thử nghiệm và được mô tả trong bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa MainActivity.java. Tên của Activity màn hình đầu tiên mà người dùng nhìn thấy, cũng khởi tạo các tài nguyên trên toàn ứng dụng, thường được gọi là MainActivity (phần mở rộng tệp bị bỏ qua 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 tệp trong trình chỉnh sửa layout



Thư mục res chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh. Mỗi hoạt động thường được liên kết với bố cục của chế độ xem UI được định nghĩa là tệp XML. Tệp này thường được đặt tên theo Activity

2.5 Tìm hiểu thư mục manifests

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 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 của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như mỗi hoạt động, phải được khai báo trong tệp XML này. Trong các bài học khác của khóa học, 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 App Manifest Overview

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

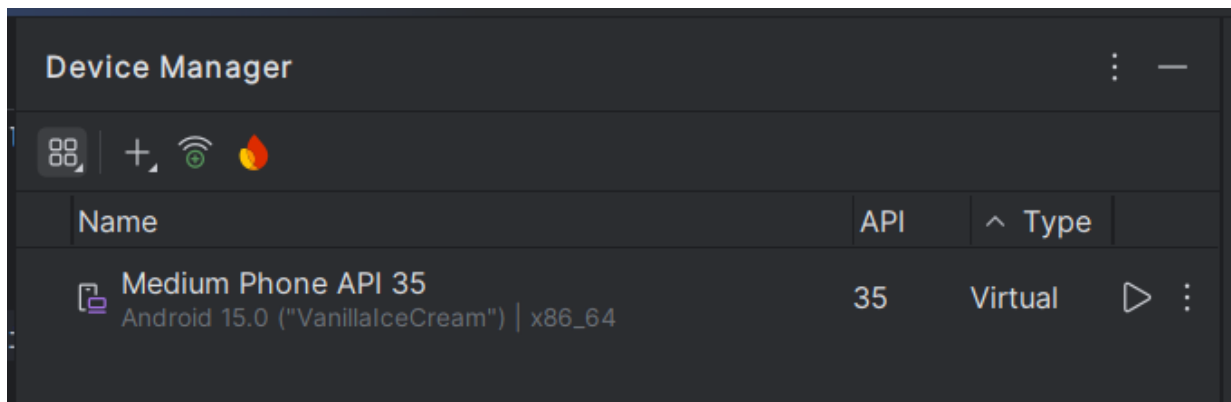
Trong tác vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo một thiết bị ảo (còn được gọi là trình giả lập) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng Trình giả lập Android có các yêu cầu bổ sung ngoài các yêu cầu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

Sử dụng AVD Manager, bạn xác định các đặc điểm phần cứng của thiết bị, mức API, bộ nhớ, giao diện và các thuộc tính khác và lưu dưới dạng thiết bị ảo. Với 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 mức API khác nhau mà không cần phải sử dụng thiết bị vật lý.

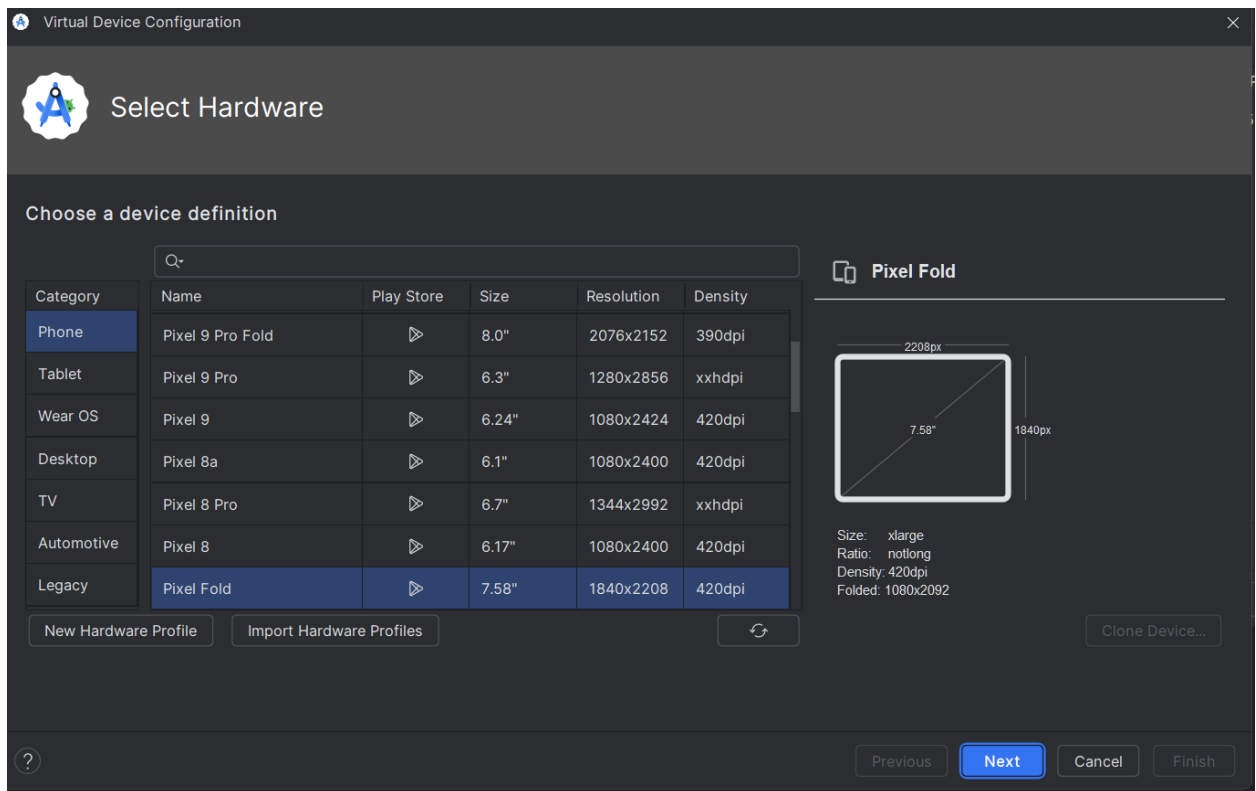
3.1 Tạo một thiết bị ảo Android (AVD)

Để chạy trình giả lập trên máy tính, bạn phải tạo 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 xuất hiện. Nếu bạn đã tạo thiết bị ảo, màn hình sẽ hiển thị chúng; nếu không, bạn sẽ thấy màn hình sau:

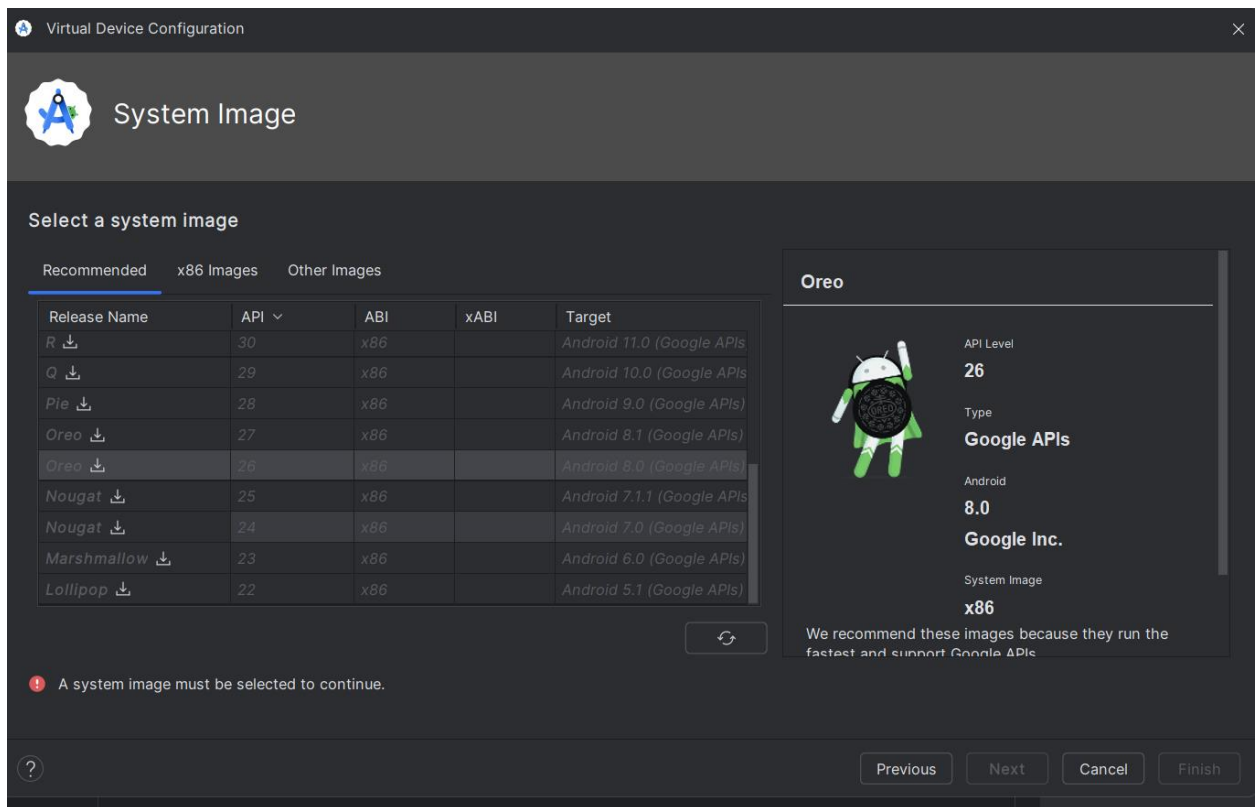


2. Nhấp vào +Create Virtual Device. Cửa sổ Select Hardware sẽ xuất hiện, hiển thị danh sách các thiết bị phần cứng được cấu hình trước. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo (size), độ phân giải màn hình theo pixel (Resolution), và mật độ pixel (Density)



3. Chọn một thiết bị như Nexus 5X hoặc Pixel XL, và nhấp vào Tiếp theo. Màn hình ảnh hệ thống sẽ xuất hiện

4. Nhấp vào tab Recommended nếu chưa được chọn và chọn phiên bản hệ thống Android nào để chạy trên thiết bị ảo (ví dụ như Oreo)



Có nhiều phiên bản khả dụng hơn những phiên bản được hiển thị trong tab Recommended. Nhìn vào hình ảnh x86 và các hình ảnh khác để xem chúng

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

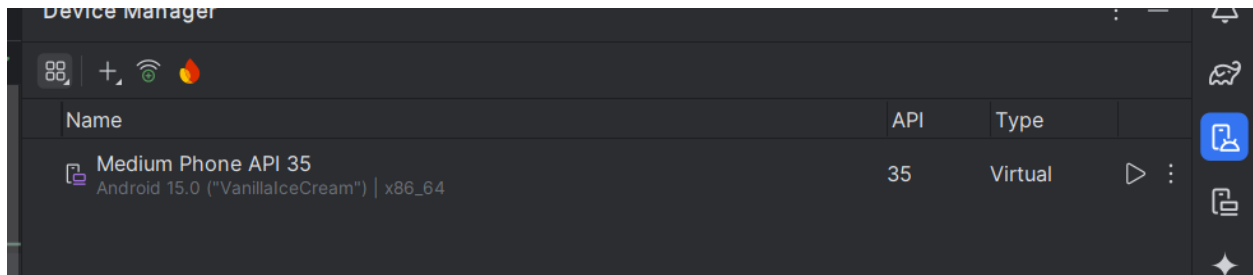
5. Sau khi chọn một hình ảnh hệ thống, hãy nhấp vào Tiếp theo . Cửa sổ Thiết bị ảo Android (AVD) xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào Kết thúc

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, chọn Run > Run app hoặc nhấp vào biểu tượng chạy 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 chạy giống như một thiết bị vật lý. Tùy thuộc vào tốc độ máy tính của bạn, việc này có thể mất một lúc. Ứng dụng của bạn được xây dựng và khi trình giả lập đã sẵn sàng, Android Studio sẽ tải ứng dụng lên trình giả lập và chạy ứng dụng đó

Bạn sẽ thấy ứng dụng Hello World như hình dưới đây



Mẹo : Khi thử nghiệm trên thiết bị ảo, bạn nên khởi động thiết bị một lần, ngay khi bắt đầu phiên làm việc. Bạn không nên đóng thiết bị cho đến khi hoàn tất thử nghiệm ứng dụng, để ứng dụng không phải trải qua quá trình khởi động thiết bị một lần nữa. Để đóng thiết bị ảo, hãy nhấp vào nút **X** ở đầu trình giả lập, chọn **Quit** từ menu hoặc nhấn **Control-Q** trong Windows hoặc **Command-Q** trong 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 nên luôn kiểm tra ứng dụng của mình trên cả thiết bị ảo và vật lý

Những gì bạn cần:

- Một thiết bị Android như điện thoại hoặc máy tính bảng
- Cáp dữ liệu để kết nối thiết bị Android của bạn với máy tính qua cổng USB
- Nếu bạn đang sử dụng hệ thống Linux hoặc Windows, bạn có thể cần thực hiện các bước để bổ sung để chạy trên thiết bị phần cứng. Kiểm tra tài liệu **Sử dụng thiết bị phần cứng**. Bạn cũng có thể cài đặt trình điều khiển USB phù 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 Android Studio giao tiếp với thiết bị của bạn, bạn phải bật USB Debugging trên thiết bị Android của mình. Tính năng này được bật trong cài đặt **tùy chọn nhà phát triển** của thiết bị

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

1. Trên thiết bị của bạn, mở **cài đặt**, tìm **Giới thiệu về điện thoại**, nhấp vào **Giới thiệu về điện thoại** và chạm vào **Số bản dựng** bảy lần.
2. Quay lại màn hình trước đó (**Cài đặt/Hệ thống**). **Tùy chọn nhà phát triển** xuất hiện trong danh sách. Chạm vào **Tùy chọn nhà phát triển** .
3. Chọn gỡ lỗi USB

4.2 Chạy ứng dụng trên thiết bị của bạn

Bây giờ bạn có thể kết nối thiết bị và chạy ứng dụng từ Android Studio.

1. Kết nối thiết bị của bạn với máy phát triển bằng cáp USB.
2. Bấm vào nút Run trên thanh công cụ. Cửa sổ Select Deployment Target mở ra với danh sách các trình giả lập có sẵn và thiết bị được kết nối
3. Chọn thiết bị của bạn và nhấp OK

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

Xử lý sự cố

Nếu Android Studio không nhận ra thiết bị của bạn, thử cách sau:

1. Rút dây cáp và cắm lại thiết bị
2. Khởi động lại Android Studio

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

1. Rút dây cáp của thiết bị
2. Trên thiết bị, mở **Tùy chọn nhà phát triển** trong **Ứng dụng cài đặt**
3. Nhấn vào thu hồi quyền gỡ lỗi USB
4. Kết nối lại thiết bị với máy tính
5. Khi được nhắc, hãy cấp quyền

Bạn có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Xem tài liệu Sử dụng thiết bị phần cứng .

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

Trong nhiệm vụ này, bạn sẽ thay đổi một số thông tin về cấu hình ứng dụng trong tệp build.gradle(Module:app) để tìm hiểu cách thực hiện thay đổi và đồng bộ hóa chúng với dự án Android Studio của bạn.

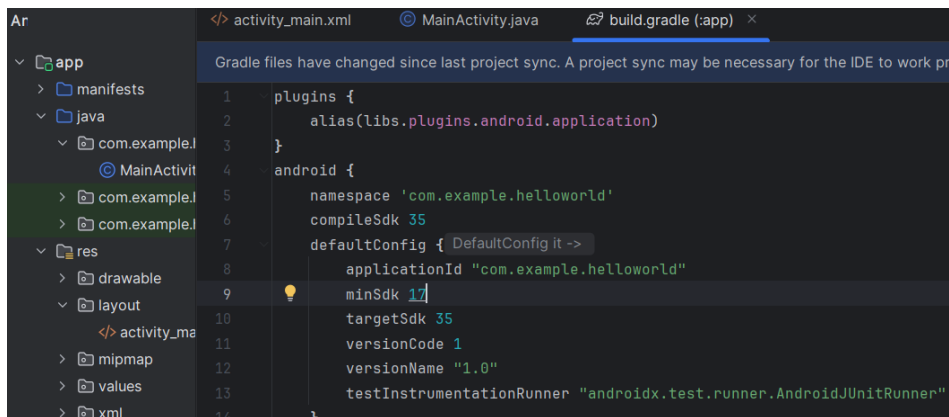
5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng

Thực hiện theo các bước sau:

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

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


2. Trong khối defaultConfig, thay đổi giá trị của minSdkVersion thành 17 như hình dưới đây (ban đầu giá trị này được đặt thành 15)



Trình chỉnh sửa mã hiển thị thanh thông báo ở trên cùng với liên kết **Đồng bộ hóa ngay**

5.2 Đồng bộ cấu hình Gradle mới

Khi bạn thực hiện thay đổi đối với tệp cấu hình bản dựng trong một dự án, Android Studio yêu cầu bạn đồng bộ hóa các tệp dự án để có thể nhập các thay đổi cấu hình bản dựng và chạy một số kiểm tra để đảm bảo cấu hình sẽ không tạo ra lỗi bản dựng.

Để đồng bộ các tệp dự án, hãy nhấp vào **Đồng bộ ngay** trên thanh thông báo xuất hiện khi thực hiện thay đổi (như hiển thị trong hình trước) hoặc nhấp vào biểu tượng **Đồng bộ dự án với tệp Gradle**  trên thanh công cụ

Khi quá trình đồng bộ hóa Gradle hoàn tất, thông báo Gradle build finished sẽ xuất hiện ở góc dưới bên trái của cửa sổ Android Studio

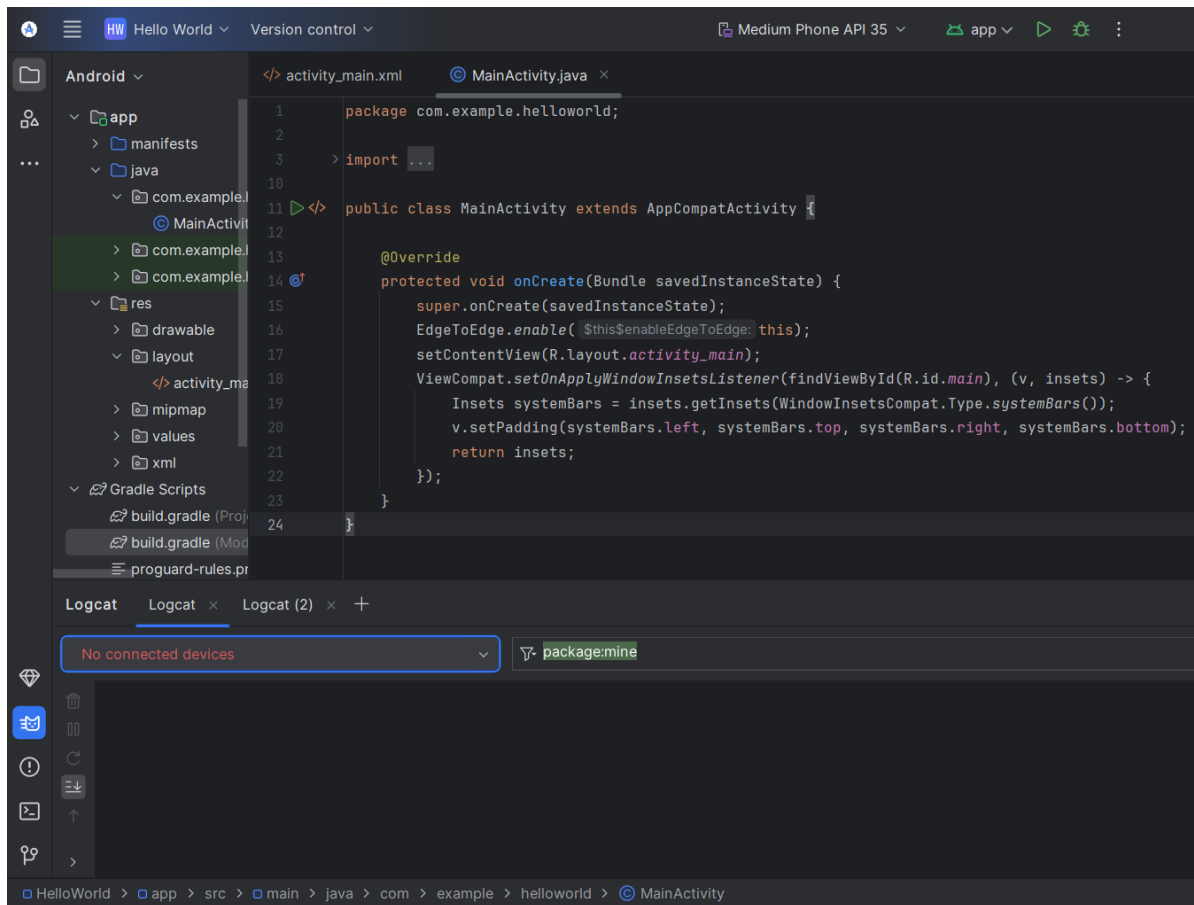
Để hiểu sâu hơn về Gradle, hãy tham khảo tài liệu Tổng quan về hệ thống xây dựng và Cấu hình bản dựng Gradle

Nhiệm vụ 6: Thêm câu lệnh Log vào ứng dụng của bạn

Trong tác vụ này, bạn sẽ thêm các câu lệnh **Log** vào ứng dụng của mình, hiển thị các thông báo trong ngăn **Logcat**. Thông báo **Log** là một công cụ gỡ lỗi mạnh mẽ mà bạn có thể sử dụng để kiểm tra các giá trị, đường dẫn thực thi và báo cáo các trường hợp ngoại lệ

6.1 Xem ngăn Logcat

Để xem ngăn **Logcat**, hãy nhấp vào tab **Logcat** ở cuối cửa sổ Android Studio như minh họa trong hình bên dưới



Trong hình trên:

1. Tab Logcat để mở và đóng ngán Logcat, hiển thị thông tin về ứng dụng của bạn khi ứng dụng đang chạy. Nếu bạn thêm câu lệnh Log vào ứng dụng, thông báo Log sẽ xuất hiện ở đây.

2. Menu cấp độ Log được đặt thành Verbose (mặc định), hiển thị tất cả các thông báo Log. Các thiết lập khác bao gồm Debug , Error , Info và Warn

6.2 Thêm câu lệnh log vào ứng dụng của bạn

Các câu lệnh Log trong mã ứng dụng của bạn hiển thị thông báo trong ngăn Logcat. Ví dụ:

```
Log.d("MainActivity", "Hello World");
```

Các phần của tin nhắn bao gồm:

- Log: Lớp Log để gửi tin nhắn nhật ký đến ngăn Logcat

- d: Cài đặt mức gỡ lỗi (Debug) Log để lọc thông báo Log hiển thị trong ngăn Logcat. Các cấp độ Log khác là e cho lỗi (Error), w cho cảnh báo (Warn) và i cho thông tin (Info).
- "MainActivity": Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong ngăn Logcat. Đây thường là tên của Activity mà thông điệp bắt đầu. Tuy nhiên, bạn có thể làm cho điều này bất cứ thứ gì hữu ích cho bạn để gỡ lỗi.

Theo quy ước, thẻ log được định nghĩa là hằng số cho Activity:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

- Hello world": Đối số thứ hai là thông điệp thực tế được hiển thị.

Thực hiện các bước sau:

1. Mở ứng dụng Hello World của bạn trong Android Studio và mở MainActivity
2. Để tự động thêm các mục nhập rõ ràng vào dự án của bạn (chẳng hạn như mục android.util.Log nhập bắt buộc khi sử dụng Log), hãy chọn **File > Settings** trong Windows hoặc Android Studio > Preferences trong macOS
3. Chọn **Editor > General > Auto Import** . Chọn tất cả các hộp kiểm và đặt **Insert imports on paste** thành **All** .
4. Nhấp vào Apply và nhấp vào OK
5. Trong phương thức onCreate() của MainActivity, thêm câu lệnh sau:

```
Log.d("MainActivity", "Hello World");
```

Phương thức onCreate() bây giờ sẽ giống như mã sau:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d("MainActivity", "Hello World");
}
```

6. Nếu ngăn Logcat chưa mở, hãy nhấp vào tab Logcat ở cuối Android Studio để mở
7. Kiểm tra xem tên của mục tiêu và tên gói của ứng dụng có chính xác không
8. Thay đổi mức Log trong ngăn Logcat thành Debug (hoặc giữ nguyên mức Verbose vì có rất ít thông báo Log)

9. Chạy ứng dụng của bạn

Thông báo sẽ xuất hiện trong ngăn Logcat:

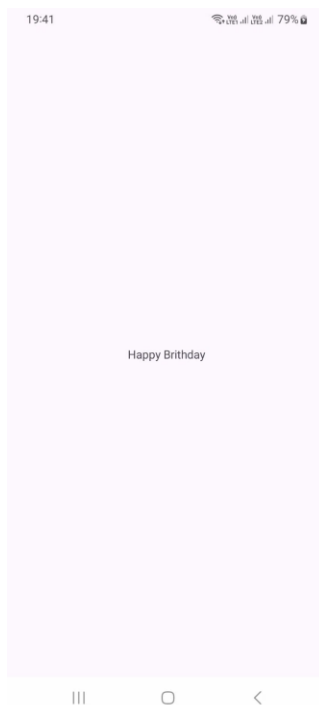
```
2025-02-24 16:52:11.493 4453-4453 MainActivity com.example.helloworld D Hello World
```

Coding challenge

Lưu ý: Tất cả các thử thách lập trình đều là tùy chọn và không phải là điều kiện tiên quyết cho các bài học sau

Thách thức: Bây giờ bạn đã thiết lập xong và quen thuộc với quy trình phát triển cơ bản, hãy thực hiện như sau:

1. Tạo một dự án mới trong Android Studio
2. Thay đổi “Hello World” thành “Happy Brithday” và tên của người có sinh nhật gần đây.
3. (Tùy chọn) Chụp ảnh màn hình ứng dụng đã hoàn thành của bạn và gửi email cho người mà bạn quên ngày sinh.



4. Một cách sử dụng phổ biến của lớp Log là ghi nhật ký các ngoại lệ Java khi chúng xảy ra trong chương trình của bạn. Có một số phương thức hữu ích, chẳng hạn như Log.e(), mà bạn có thể sử dụng cho mục đích này. Khám phá các phương thức bạn có thể sử dụng để

bao gồm ngoại lệ với thông báo Nhật ký. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi nhật ký ngoại lệ

Tổng kết

- Để cài đặt Android Studio, hãy vào Android Studio và làm theo hướng dẫn để tải xuống và cài đặt.
- Khi tạo ứng dụng mới, hãy đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu.
- Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy nhấp vào tab Dự án trong cột tab dọc, sau đó chọn Android trong menu bật lên ở trên cùng.
- Chỉnh sửa build.gradle(Module:app)tệp khi bạn cần thêm thư viện mới vào dự án hoặc thay đổi phiên bản thư viện.
- Tất cả mã và tài nguyên cho ứng dụng đều nằm trong các thư mục app và res. javaThư mục bao gồm các hoạt động, thử nghiệm và các thành phần khác trong mã nguồn Java. resThư mục chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh.
- Chỉnh sửa AndroidManifest.xmltệp để thêm các thành phần tính năng và quyền vào ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML này.
- Sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo thiết bị ảo (còn gọi là trình giả lập) để chạy ứng dụng của bạn.
- Thêm Logcác câu lệnh vào ứng dụng của bạn để hiển thị thông báo trong ngăn Logcat như một công cụ cơ bản để gỡ lỗi.
- Để chạy ứng dụng của bạn trên thiết bị Android vật lý bằng Android Studio, hãy bật Gỡ lỗi USB trên thiết bị. Mở Cài đặt > Giới thiệu về điện thoại và chạm vào Số bản dựng bảy lần. Quay lại màn hình trước đó (Cài đặt) và chạm vào Tùy chọn nhà phát triển . Chọn Gỡ lỗi USB

Các khái niệm liên quan

Tài liệu về khái niệm liên quan có trong 1.0: Giới thiệu về Android và 1.1 Ứng dụng Android đầu tiên của bạn

Tìm hiểu thêm

Tài liệu Android Studio

- Trang tải xuống Android Studio
- Ghi chú phát hành Android Studio
- Làm quen Android Studio
- Công cụ dòng lệnh Logcat
- Thiết bị ảo (AVD)
- Tổng quan về ứng dụng
- Cấu hình bản dựng của bạn
- Lớp Log
- Tạo và quản lý các thiết bị ảo

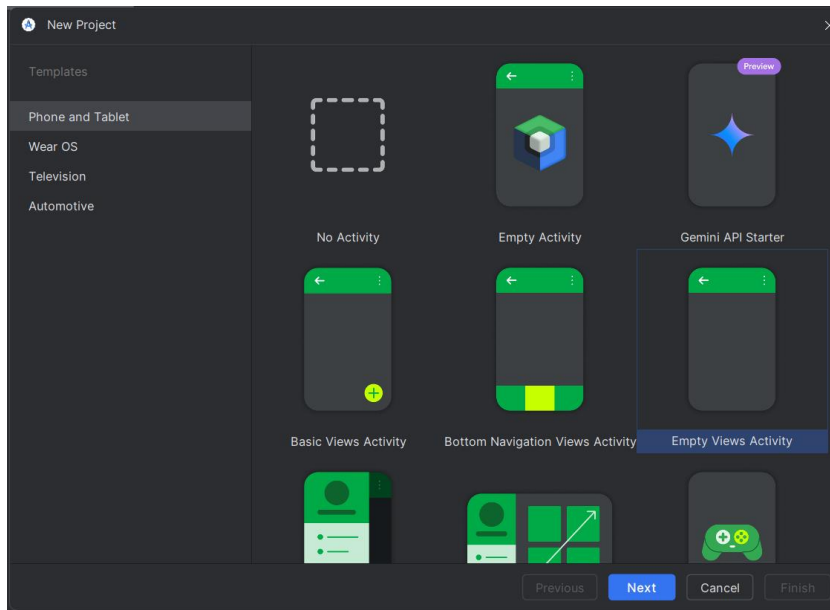
Khác:

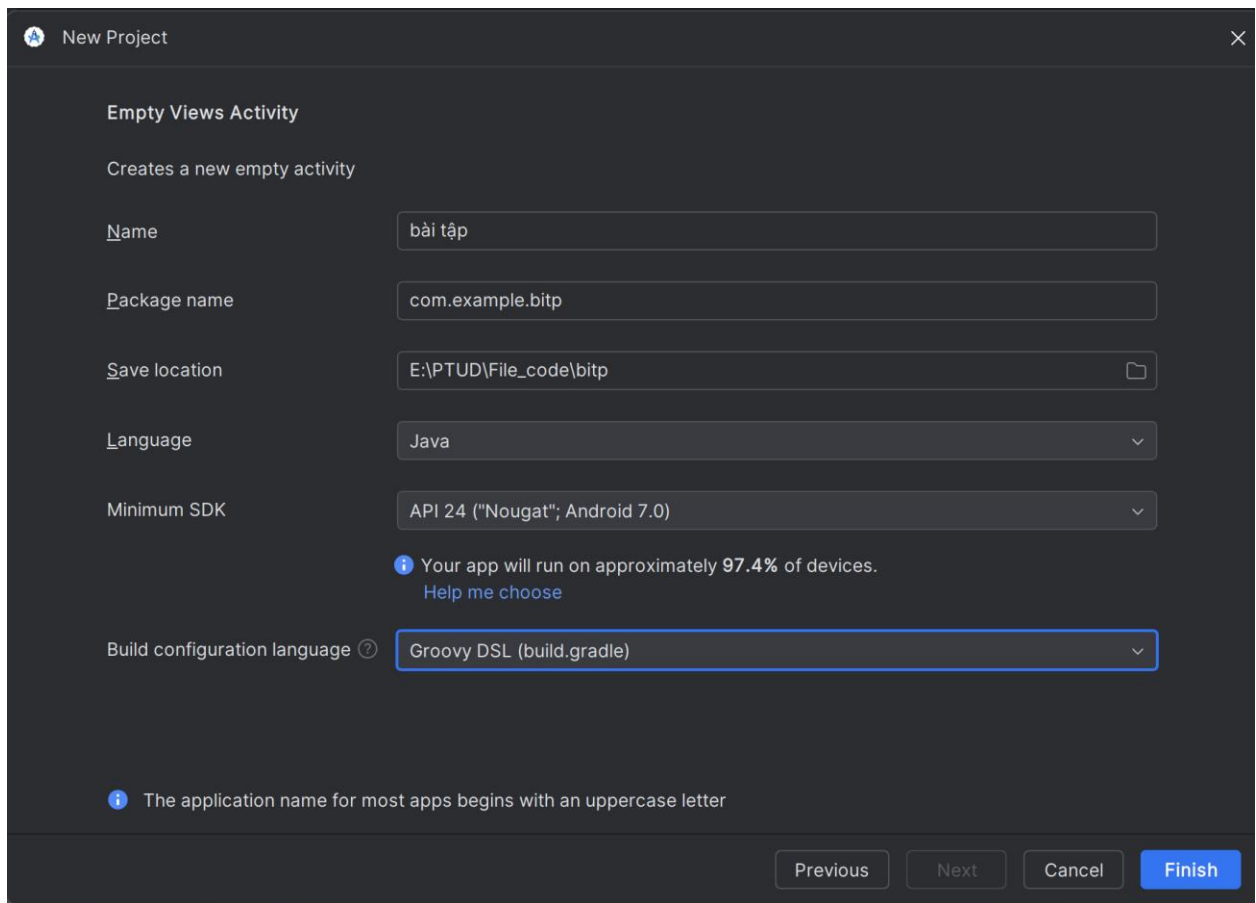
- Làm cách nào cài đặt Java?
- Cài đặt phần mềm JDK và cài đặt JAVA_HOME
- Trang web Gradle
- Cú pháp Apache Groovy
- Trang Wikipedia Gradle

Bài tập

Tạo và chạy một ứng dụng

- Tạo một dự án Android mới từ mẫu

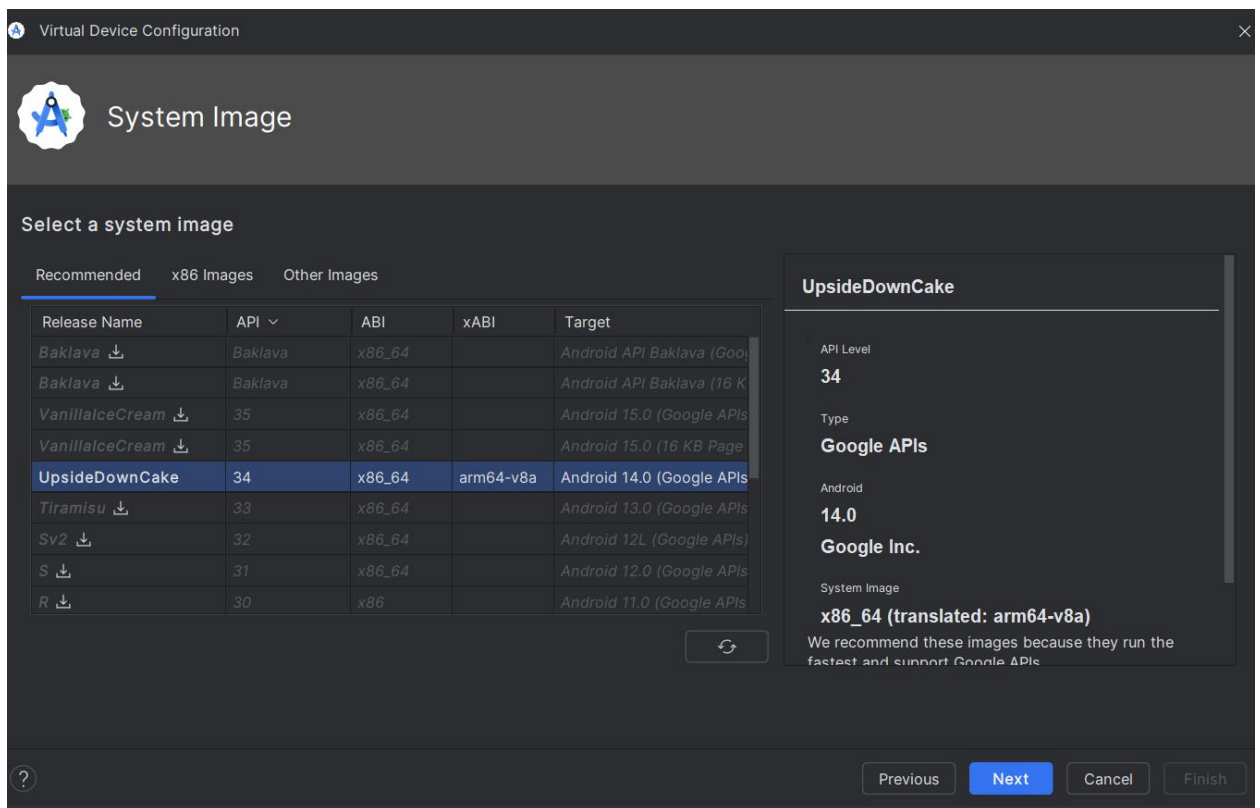
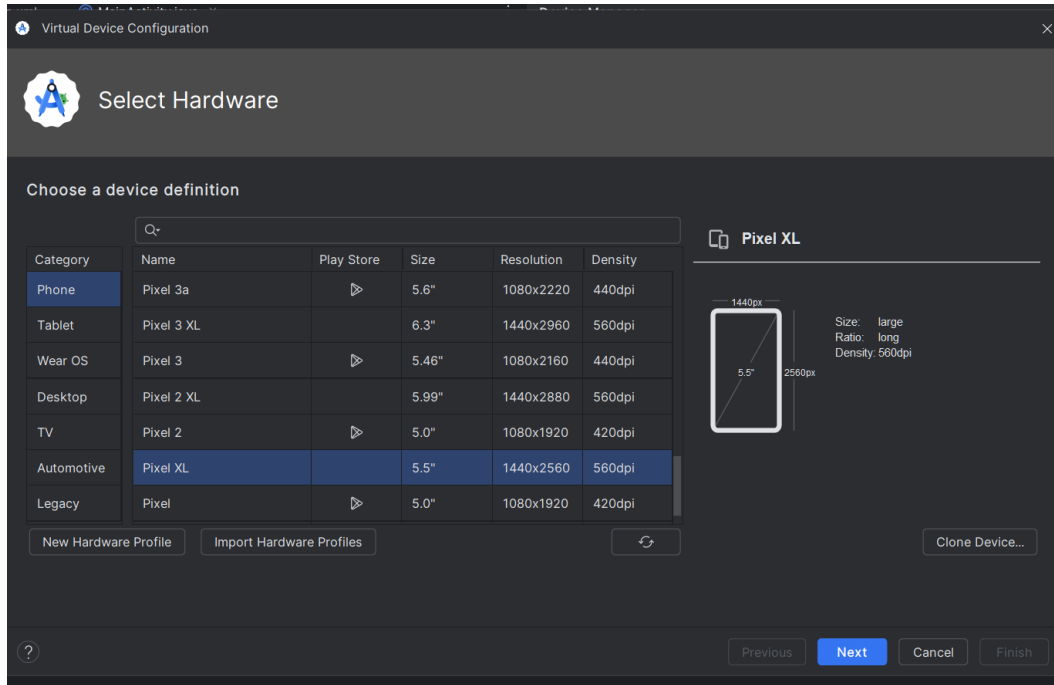


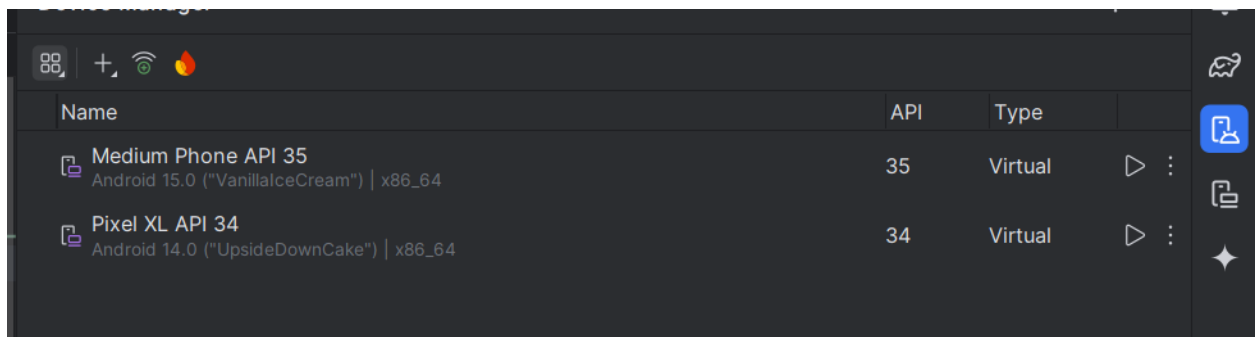
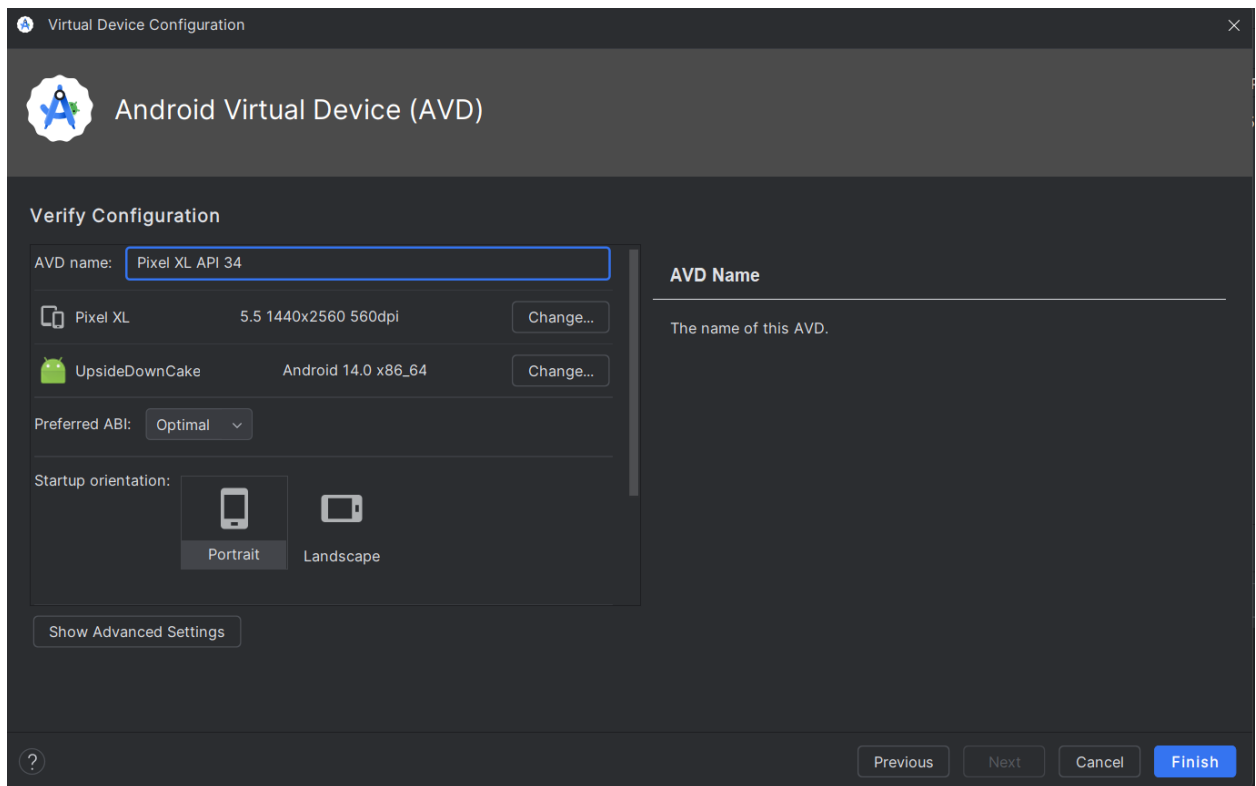


- Thêm câu lệnh Log cho các mức Log khác nhau trong onCreate() trong hoạt động chính



- Tạo trình giả lập cho thiết bị, chọn bất kỳ phiên bản Android nào bạn thích và chạy ứng dụng





- Sử dụng bộ lọc trong Logcat để tìm các câu lệnh Log của bạn và điều chỉnh các cấp độ để chỉ hiển thị các câu lệnh gỡ lỗi hoặc ghi lỗi

Trả lời các câu hỏi sau

Câu hỏi 1

Tên của tệp layout cho main activity là gì?

- MainActivity.java
- AndroidManifest.xml
- **Activity_main.xml**

- Build.gradle

Câu hỏi 2

Tên của resource string xác định tên ứng dụng là gì?

- app_name
- xmlns:app
- android:name
- **applicationId**

Câu hỏi 3

Công cụ nào để tạo trình giả lập mới?

- Android Device Monitor
- **AVD Manager**
- SDK Manager
- Theme Editor

Câu hỏi 4

Giả sử ứng dụng của bạn có chứa câu lệnh log sau đây:

Log.i ("MainActivity", "MainActivity layout is complete");

Bạn sẽ thấy dòng thông báo “MainActivity layout is complete” trong ngăn Logcat nếu menu Log được đặt ở mức nào trong các lựa chọn sau? (Gợi ý: nhiều câu trả lời đúng)

- **Verbose**
- Debug
- **Info**
- Warn
- Error
- Assert

Gửi ứng dụng của bạn để chấm điểm

Đảm bảo kiểm tra ứng dụng có những điều sau:

- Hoạt động hiển thị “Hello World” trên màn hình
- Các câu lệnh Log trong onCreate() trong hoạt động chính

- Mức Log trong ngăn Logcat hiển thị các câu lệnh gỡ lỗi hoặc ghi lỗi

1.2) Giao diện người dùng tương tác đầu tiên

Giới thiệu

Giao diện người dùng (UI) xuất hiện trên màn hình của thiết bị Android bao gồm một hệ thống phân cấp các đối tượng được gọi là view – mỗi phần tử trên màn hình là một view. Lớp view biểu thị khối xây dựng cơ bản cho tất cả các thành phần UI, và là lớp cơ sở cho các lớp cung cấp các thành phần UI tương tác như nút, hộp kiểm và trường nhập văn bản. Các lớp con View thường được sử dụng để trình bày trong nhiều bài học:

- TextView : Hiển thị văn bản.
- EditText : cho phép người dùng nhập và chỉnh sửa văn bản.
- Button và các thành phần có thể nhấp khác (như RadioButton, Checkbox và Spinner) : để cung cấp hành vi tương tác.
- ScrollView và RecyclerView : để hiển thị các mục có thể cuộn được.
- ImageView : để hiển thị hình ảnh.
- ConstraintLayout và LinearLayout : để chứa các thành phần View khác và định vị của chúng.

Mã Java hiển thị và điều khiển giao diện người dùng(UI) được chứa trong một lớp kế thừa Activity. Một Activity thường được liên kết với một bố cục của các View(Thành phần giao diện) được định nghĩa dưới dạng tệp XML(Ngôn ngữ đánh dấu mở rộng). Tệp XML này thường được đặt tên theo Activity của nó và định nghĩa bố cục của các thành phần View trên màn hình.

Ví dụ, mã MainActivity trong ứng dụng Hello World hiển thị một bố cục được xác định trong tệp layout activity_main.xml, bao gồm một TextView với văn bản “Hello World”.

Trong các ứng dụng phức tạp hơn, một Activity có thể thực hiện các hành động để phản hồi với các lần chạm của người dùng, vẽ nội dung đồ họa, hoặc yêu cầu dữ liệu từ cơ sở dữ liệu hoặc internet. Bạn sẽ tìm hiểu thêm về lớp Activity trong một bài học khác.

Trong bài thực hành này, bạn sẽ học cách tạo ứng dụng tương tác đầu tiên của mình - một ứng dụng cho phép người dùng tương tác. Bạn sẽ tạo một ứng dụng sử dụng mẫu Empty Activity. Bạn cũng sẽ học cách sử dụng trình

chỉnh sửa bố cục để thiết kế một bố cục, và cách chỉnh sửa bố cục trong XML. Bạn cần phát triển những kỹ năng này để có thể hoàn thành các bài thực hành khác trong khóa học này.

Những gì Bạn nên biết:

Bạn nên có quen thuộc với:

- Cách cài đặt và mở Android Studio.
- Cách tạo ứng dụng HelloWorld.
- Cách chạy ứng dụng HelloWorld.

Những gì Bạn sẽ học:

- Cách tạo ứng dụng với hành vi tương tác.
- Cách sử dụng trình chỉnh sửa bố cục để thiết kế một bố cục
- Cách chỉnh sửa bố cục trong XML.

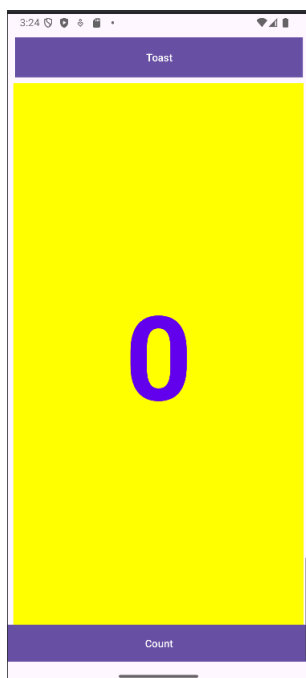
Những gì Bạn sẽ làm:

- Tạo một ứng dụng và thêm hai thành phần Button và một TextView và bố cục.
- Thao tác với từng thành phần trong ConstraintLayout để ràng buộc chúng với các lề và các thành phần khác.
- Thay đổi các thuộc tính thành phần UI.
- Chỉnh sửa bố cục của ứng dụng trong XML.
- Trích xuất các chuỗi được mã hóa cứng thành tài nguyên chuỗi.
- Triển khai các phương thức xử lý sự kiện nhấp chuột để hiển thị thông điệp trên màn hình khi người dùng chạm vào từng Button.

Tổng quan về ứng dụng

Ứng dụng HelloToast bao gồm hai thành phần Button và TextView. Khi người dùng nhấn vào Button đầu tiên, ứng dụng sẽ hiển thị một thông điệp ngắn(một Toast) trên màn hình. Việc nhấn vào button thứ hai sẽ tăng một bộ đếm “nhấp chuột” được hiển thị trong TextView, bắt đầu từ số 0.

Dưới đây là mô tả về giao diện của ứng dụng hoàn chỉnh:

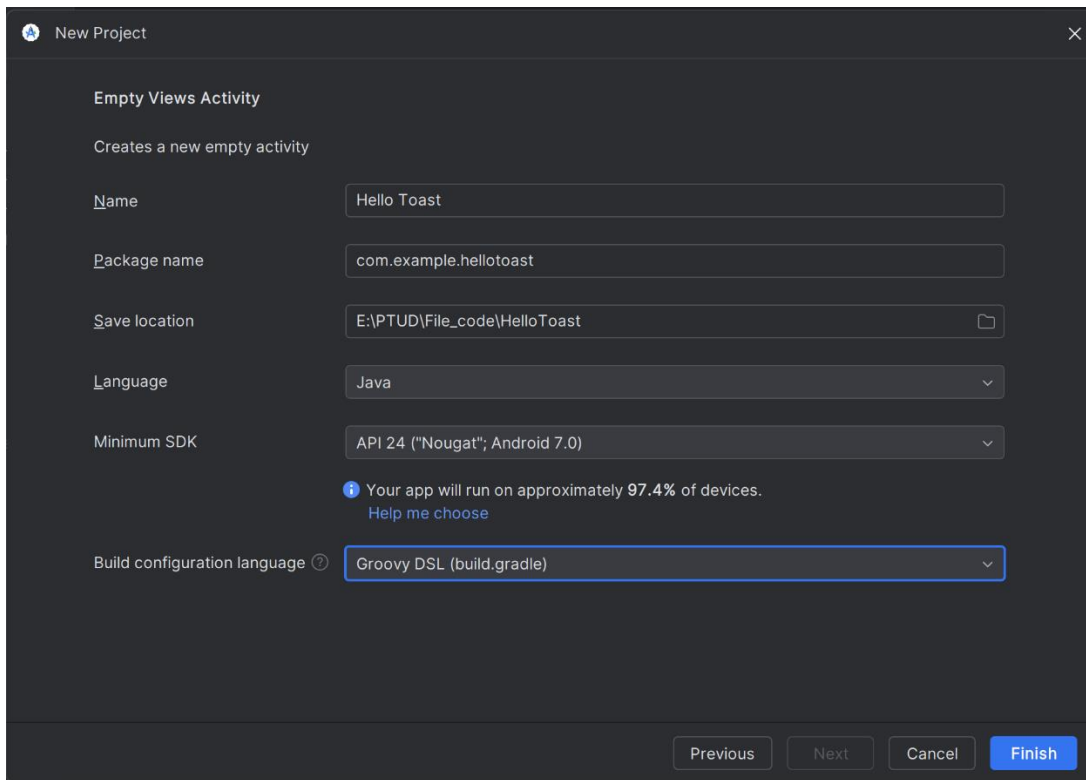


Nhiệm vụ 1: Tạo và khám phá một dự án mới


Trong bài thực hành này, bạn sẽ thiết kế và triển khai một dự án cho ứng dụng HelloToast. Một liên kết đến mã giải pháp được cung cấp ở cuối.

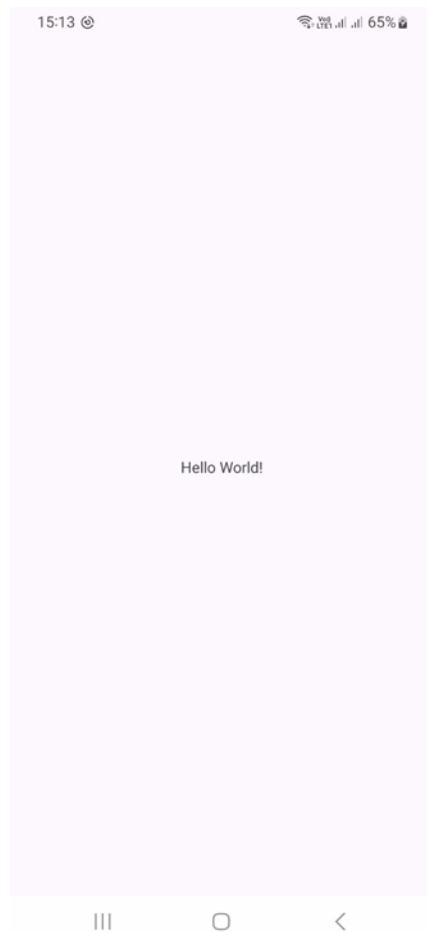
1.1 Tạo dự án Android Studio

Khởi động Android Studio và tạo một dự án mới với các tham số sau:



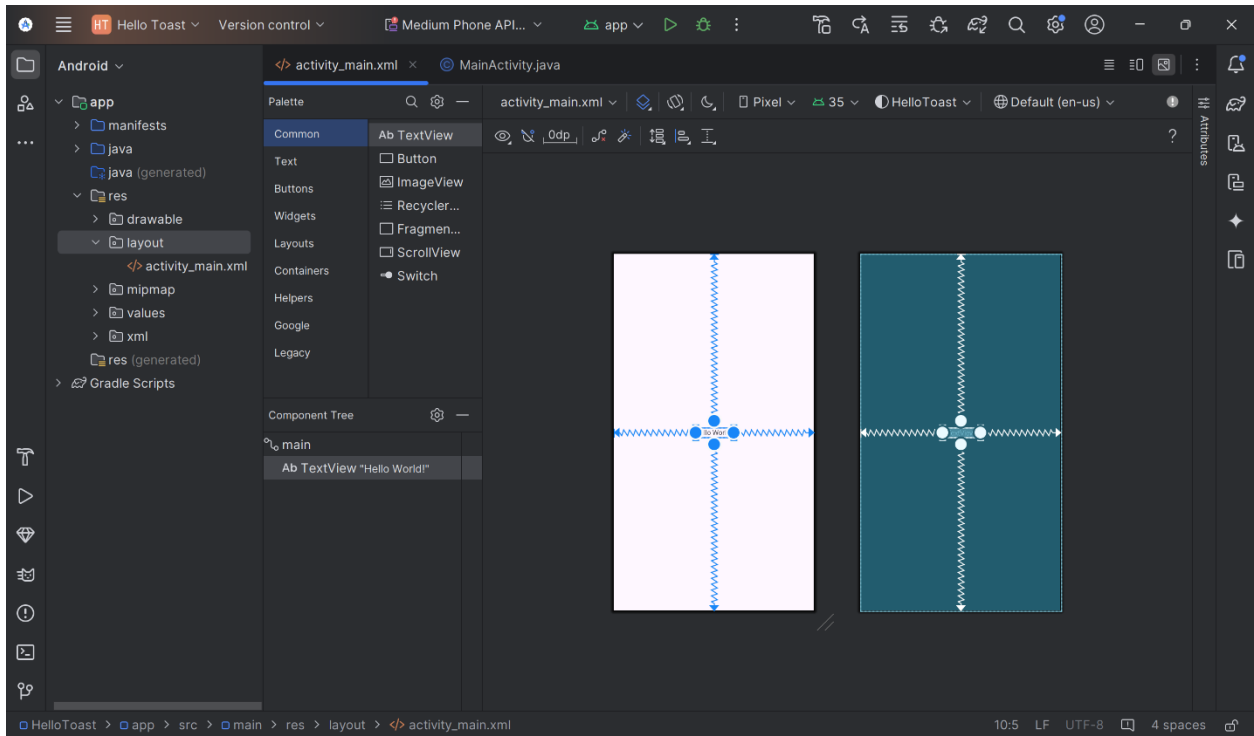
15.

Chọn Run > Run app hoặc nhấp chuột vào biểu tượng Run  trên thanh công cụ để xây dựng và thực thi ứng dụng trên trình giả lập hoặc thiết bị của bạn.



1.2 Khám phá trình chỉnh sửa bố cục

Android Studio cung cấp trình chỉnh sửa bố cục để nhanh chóng xây dựng bố cục của các thành phần giao diện người dùng (UI) trong ứng dụng. Nó cho phép bạn kéo các thành phần vào chế độ thiết kế trực quan và chế độ bản vẽ, định vị chúng trong bố cục, thêm các ràng buộc và thiết lập thuộc tính. Các ràng buộc xác định vị trí của một thành phần UI trong bố cục. Một ràng buộc đại diện cho một kết nối hoặc sự căn chỉnh với một view khác, bố cục cha hoặc một hướng dẫn vô hình.



1. Trong thư mục app > res > layout trong bảng Project > Android, nhấp đúp vào tệp activity_main.xml để mở nó nếu nó chưa được mở.
2. Nhấp vào tab Design nếu nó chưa được chọn. Bạn sử dụng tab Design để thao tác với các thành phần và bố cục, và tab Text để chỉnh sửa mã XML cho bố cục.
3. Bảng Palettes hiển thị các thành phần UI mà bạn có thể sử dụng trong bố cục của ứng dụng.
4. Bảng Component hiển thị cấu trúc phân cấp các thành phần UI. Các thành phần View được tổ chức thành một cấu trúc cây với các thành phần cha và con, trong đó một phần tử con được kế thừa các thuộc tính với thành phần cha. Trong hình trên TextView là một phần tử con của ConstraintLayout. Bạn sẽ tìm hiểu các thành phần này sau trong bài học này.
5. Các bảng thiết kế và bản vẽ của trình chỉnh sửa bố cục hiển thị các thành phần UI trong bố cục. Trong hình trên, bố cục hiển thị một thành phần: TextView hiển thị "Hello World".
6. Tab Attributes hiển thị bảng Attributes để thiết lập các thuộc tính cho một thành phần UI.

Mẹo: Xem [Building a UI with Layout Editor](#) để biết chi tiết về việc sử dụng trình chỉnh sửa bố cục và [Meet Android Studio](#) để có tài liệu đầy đủ về Android Studio.

Nhiệm vụ 2: Thêm các thành phần View trong trình chỉnh sửa bố cục

Trong nhiệm vụ này, bạn sẽ tạo bố cục giao diện người dùng(UI) cho ứng dụng HelloToast trong trình chỉnh sửa bố cục bằng cách sử dụng các tính năng của ConstraintLayout. Bạn có thể tạo các ràng buộc một cách thủ công, như sẽ được trình bày sau đây, hoặc tự động bằng cách sử dụng công cụ Autoconnect.

2.1 Kiểm tra ràng buộc của các thành phần

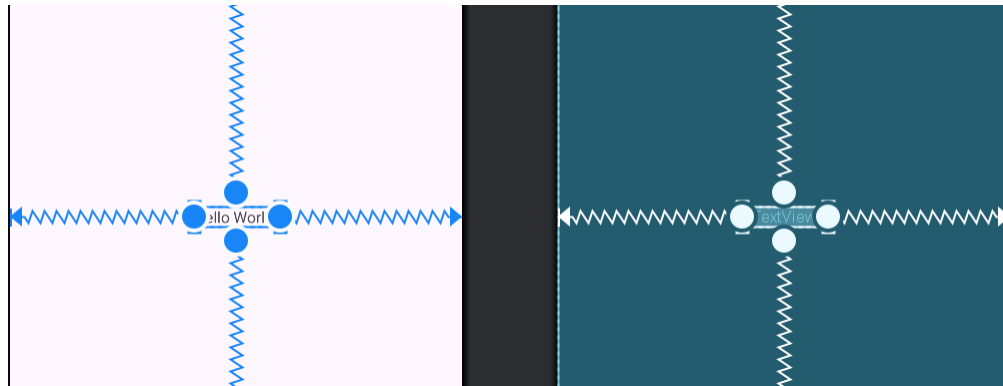
1. Mở tệp activity_main.xml:

- Trong bảng Project > Android, mở tệp activity_main.xml nếu nó chưa được mở. Nếu tab Design chưa được chọn, hãy nhấp vào nó.
- Nếu không có bản vẽ, hãy nhấp vào nút Select Design Surface



trong thanh công cụ và chọn Design and Blueprint.

2. Công cụ Autoconnect cũng nằm trong thanh công cụ. Nó được bật theo mặc định. Để thực hiện bước này, hãy đảm bảo rằng công cụ này không bị vô hiệu hóa.
3. Nhấn vào nút Zoom in để phóng to khu vực thiết kế và bản vẽ nhằm có cái nhìn rõ hơn về các thành phần giao diện người dùng trong bố cục.
4. Chọn TextView trong bảng Component Tree. TextView có văn bản “Hello World” sẽ được làm nổi bật trong các khu vực thiết kế và bản vẽ và các ràng buộc cho thành phần này sẽ hiển thị rõ ràng.
5. Tham khảo hình ảnh động bên dưới cho các bước trên. Nhấp vào nút hình tròn ở phía bên phải của TextView để xóa ràng buộc ngang liên kết thành phần với bên phải bố cục. TextView sẽ nhảy sang bên trái vì nó không còn bị ràng buộc với bên phải. Để thêm lại ràng buộc ngang, nhấp vào cùng một nút và kéo một đường đến bên phải bố cục.



Trong các khung bản đồ hoặc thiết kế, các nút sau đây xuất hiện trên thành phần TextView:

- Xử lý ràng buộc: Để tạo một ràng buộc như trong hình trên, hãy nhấp vào một ràng buộc, được biểu thị bằng một hình tròn ở bên cạnh thành phần. Sau đó, kéo ràng buộc đó đến một ràng buộc khác, hoặc đến một ranh giới bố cục cha. Một đường zigzag sẽ đại diện cho ràng buộc.



- Thay đổi kích thước nút: Để thay đổi kích thước thành phần, hãy kéo các nút thay đổi kích thước hình vuông. Nút sẽ chuyển thành góc cạnh trong khi bạn đang kéo nó.

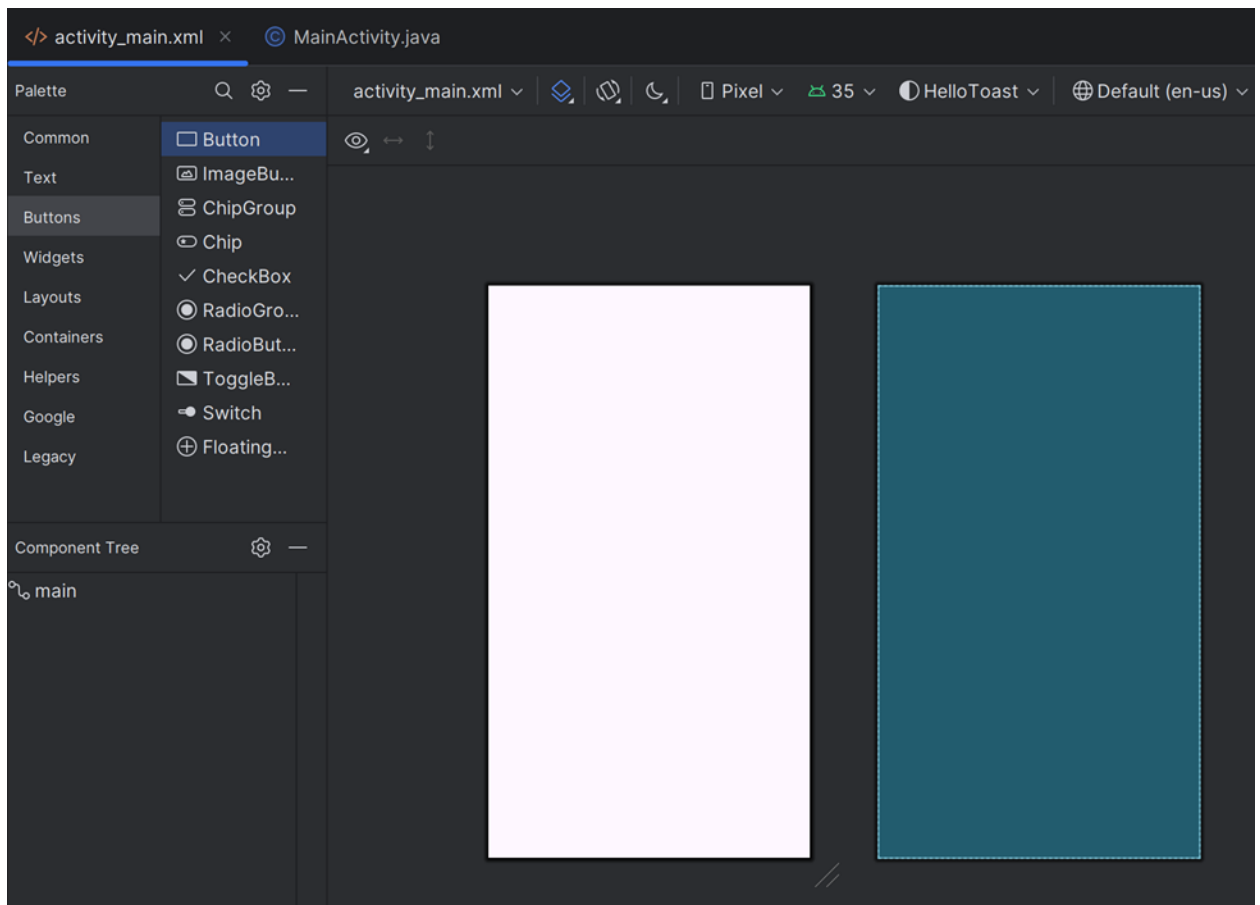


2.2 Thêm Button vào bố cục

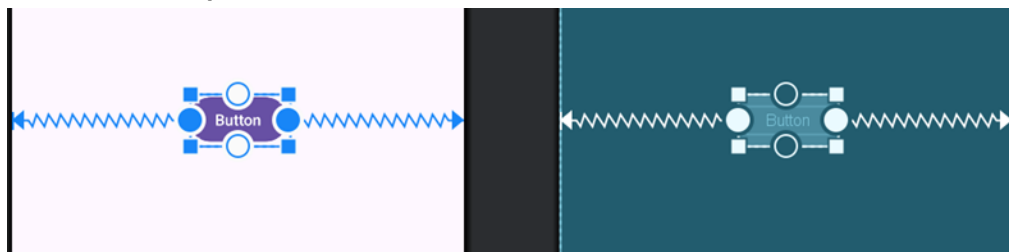
Khi được kích hoạt, công cụ Autoconnect sẽ tự động tạo hai hoặc nhiều ràng buộc cho một thành phần UI đến bố cục cha. Sau khi bạn kéo thành phần vào bố cục, nó sẽ tạo các ràng buộc dựa trên vị trí của thành phần.

Làm theo các bước sau để thêm Button:

1. Bắt đầu bằng một trang trắng. Thành phần TextView là không cần thiết, vì vậy trong khi nó vẫn được chọn, hãy nhấn phím Delete hoặc chọn Edit > Delete. Bây giờ bạn đã có một bố cục hoàn toàn trống.

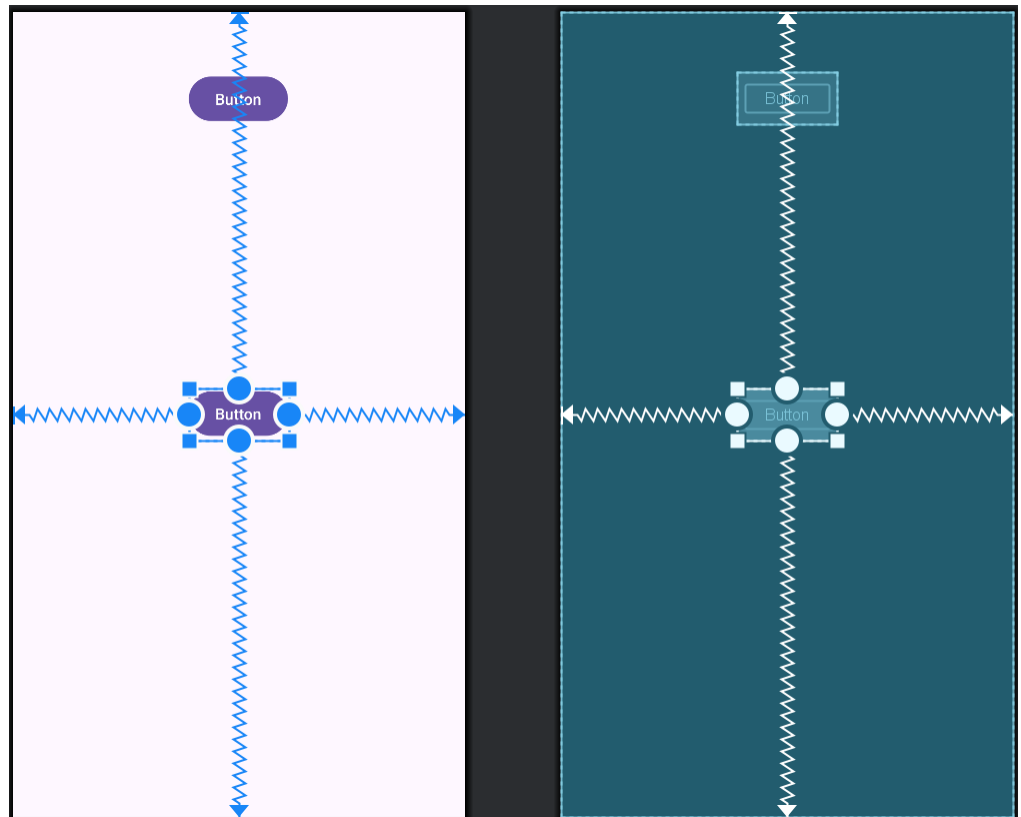


2. Kéo một Button từ bảng Palette vào bất kỳ vị trí nào trong bố cục. Nếu bạn thả Button ở khu vực giữa trên cùng của bố cục, các ràng buộc có thể tự động xuất hiện. Nếu không, bạn có thể kéo thả các ràng buộc đến phía trên, bên trái và bên phải của bố cục như được hiển thị bên dưới.



2.3 Thêm Button thứ hai vào bố cục

1. Kéo một Button khác từ bảng Palette vào giữa bố cục như được hiển thị trong hình bên dưới. Autoconnect có thể cung cấp ràng buộc ngang cho bạn (nếu không, bạn có thể tự kéo chúng).



2. Kéo một ràng buộc dọc đến đáy của bố cục (tham khảo hình bên dưới)



Bạn có thể xóa các ràng buộc khỏi một thành phần bằng cách chọn thành phần đó và di chuột qua nó để hiển thị nút Clear Constraints



Nhấp vào nút này để xóa tất cả các ràng buộc trên thành phần đã chọn. Để xóa một ràng buộc cụ thể, hãy nhấp vào tay cầm cụ thể thiết lập ràng buộc đó.

Để xóa tất cả các ràng buộc trong toàn bộ bố cục, hãy nhấp vào công cụ Clear All Constraints trên thanh công cụ. Công cụ này rất hữu ích nếu bạn muốn làm lại tất cả các ràng buộc trong bố cục của mình.

Nhiệm vụ 3: Thay đổi các thành phần giao diện người dùng(UI)

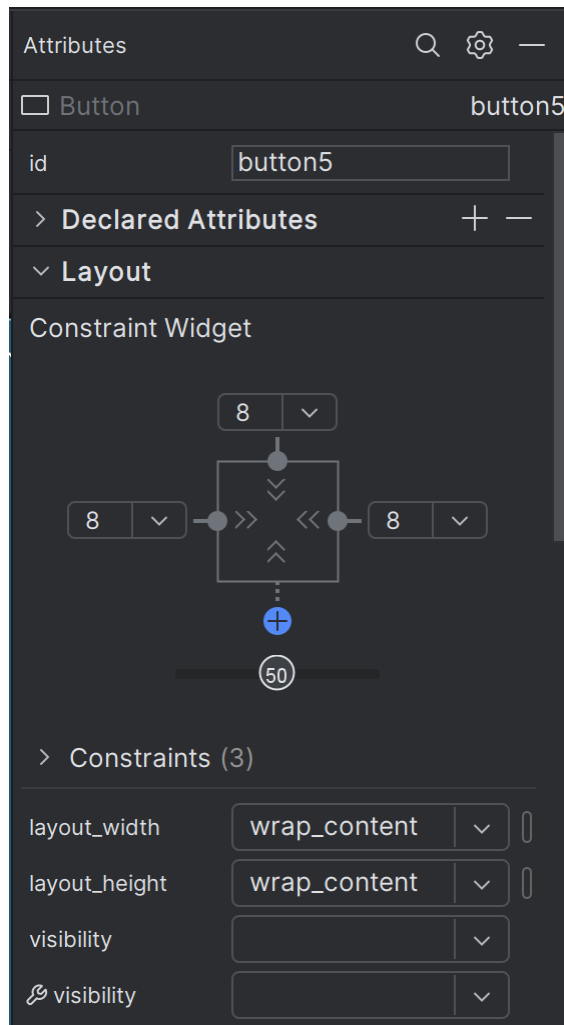
Bảng thuộc tính cung cấp quyền truy cập vào tất cả các thuộc tính XML mà bạn có thể gán cho một thành phần UI. Bạn có thể tìm thấy các thuộc tính(được gọi

là thuộc tính) ching cho tất cả các view trong tài liệu lớp view. Trong nhiệm vụ lần này, bạn sẽ nhập các giá trị mới và thay đổi các giá trị cho các thuộc tính quan trọng của Button, áp dụng cho hầu hết các loại view.

3.1 Thay đổi kích thước Button

Trình chỉnh sửa bố cục cung cấp các nút điều chỉnh kích thước ở cả bốn góc của View để bạn có thể thay đổi kích thước View một cách nhanh chóng. Bạn có thể kéo các nút điều chỉnh ở mỗi góc của view để thay đổi kích thước, nhưng làm như vậy sẽ cố định các kích thước chiều rộng và chiều cao. Tránh việc cố định kích thước cho hầu hết các thành phần của View, vì các kích thước cố định không thể thích ứng với nội dung và kích thước màn hình khác nhau.

Thay vào đó, hãy sử dụng bảng thuộc tính ở bên phải của trình chỉnh sửa bố cục để chọn chế độ kích thước không sử dụng kích thước cố định. Bảng thuộc tính bao gồm một bảng kích thước hình vuông gọi là trình kiểm tra chế độ xem ở phía trên. Các ký hiệu bên trong hình vuông đại diện cho các cài đặt chiều cao và chiều rộng như sau:



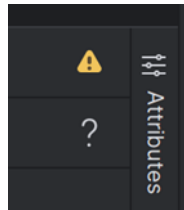
Trong hình trên có:

1. Điều chỉnh chiều cao: Điều chỉnh này được xác định bởi thuộc tính "layout_height" và xuất hiện ở hai phần trên và dưới của hình vuông. Các góc cho thấy điều chỉnh này được thiết lập để "wrap_content", có nghĩa là "View" sẽ mở rộng theo chiều dọc khi cần thiết để phù hợp với nội dung của nó. Số "8" chỉ ra rằng có một khoảng cách tiêu chuẩn được thiết lập là 8dp.
2. Điều chỉnh chiều rộng: Điều chỉnh này được xác định bởi thuộc tính "layout_width" và xuất hiện ở hai phần trái và phải của hình vuông. Các góc cho thấy điều chỉnh này được thiết lập để "wrap_content", có nghĩa là "View" sẽ mở rộng theo chiều ngang khi cần thiết để phù hợp với nội dung của nó, tối đa đến một khoảng cách là 8dp.

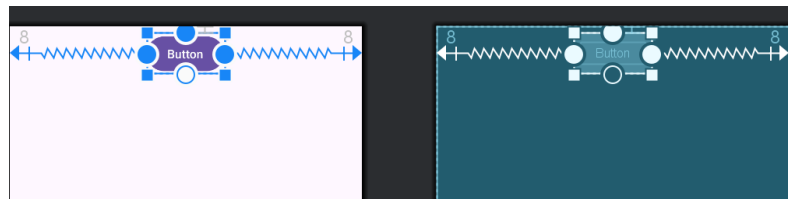
3. Nút đóng bảng thuộc tính. Nhấn để đóng bảng.

Thực hiện các bước sau:

1. Chọn nút ở trên cùng trong bảng Component Tree.
2. Nhấp vào tab Attributes ở bên phải của cửa sổ trình chỉnh sửa bố cục.

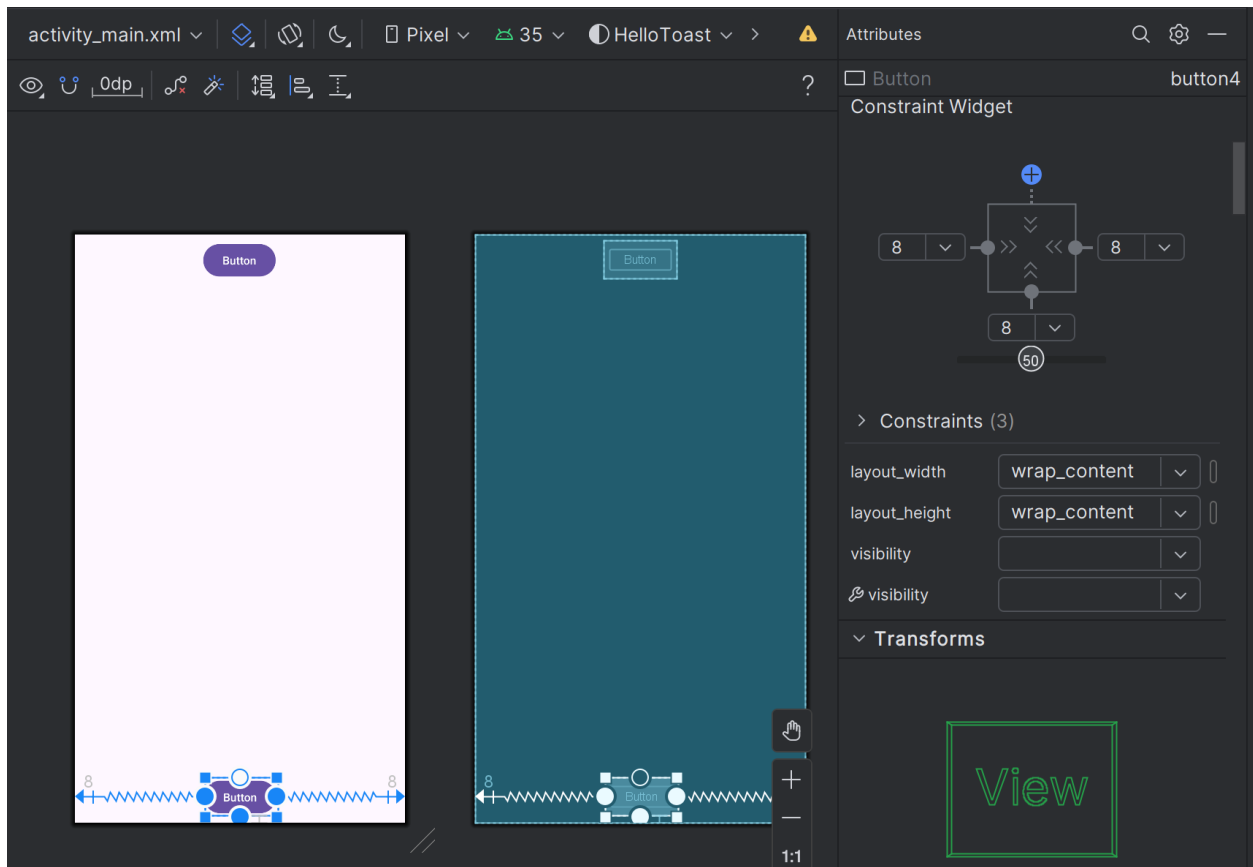


3. Nhấp vào điều chỉnh chiều rộng hai lần – lần nhấp đầu tiên sẽ thay đổi nó thành cố định với các đường thẳng, và nhấp lần thứ hai sẽ thay đổi nó thành khớp ràng buộc với các cuộn lò xo như hình bên dưới:



Do thay đổi điều khiển chiều rộng, `layout_width` thuộc tính trong ngăn Thuộc tính sẽ hiển thị giá trị `match_constraint` và Button phần tử sẽ kéo dài theo chiều ngang để lấp đầy khoảng trống giữa bên trái và bên phải của bố cục.

4. Chọn nút thứ hai và thực hiện các thay đổi tương tự `layout_width` như ở bước trước, thể hiện trong hình bên dưới:



Như đã được trình bày ở các bước trước, các thuộc tính `layout_width` và `layout_height` trong ngăn Attributes thay đổi khi bạn thay đổi các điều khiển hiệu cao và chiều rộng trong trình kiểm tra. Các thuộc tính này có thể lấy một trong ba giá trị cho bố cục, đó là `ConstraintLayout`:

- Cài đặt `Match_constraint` mở rộng phần tử chế độ xem để lấp đầy phần tử cha theo chiều rộng hoặc chiều cao – lên đến một lề, nếu có. Phần tử cha trong trường hợp này là `ConstraintLayout`. Bạn tìm hiểu thêm về `ConstraintLayout` trong nhiệm vụ tiếp theo.
- Cài đặt `wrap_content` thu nhỏ kích thước của phần tử View để nó vừa đủ lớn để bao bọc nội dung của nó. Nếu không có nội dung, phần tử View sẽ trở nên vô hình.
- Để chỉ định kích thước cố định điều chỉnh cho kích thước màn hình của thiết bị, hãy sử dụng một số pixel cố định không phụ thuộc vào mật độ (đơn vị dp). Ví dụ: 16dp có nghĩa là 16 pixel không phụ thuộc vào mật độ.

Mẹo: Nếu bạn thay đổi thuộc tính `layout_width` bằng `menu` bật lên của nó, thuộc tính `layout_width` được đặt thành không vì không có thứ nguyên được đặt. Cài đặt này giống như `match_constraint` - chế độ xem có thể mở rộng càng nhiều càng tốt để đáp ứng các ràng buộc và cài đặt lề.

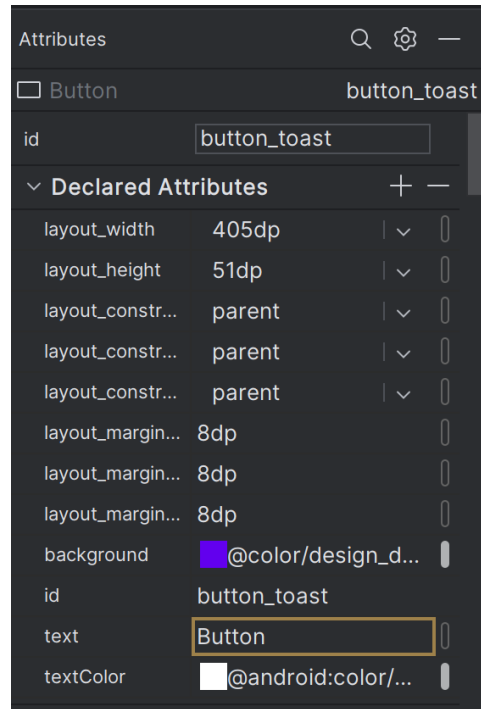
3.2 Thay đổi thuộc tính của Button

Để xác định từng phần tử View một cách duy nhất trong một bố cục Activity, mỗi phần tử View hoặc View lớp con (như Button) cần một ID duy nhất. Và để có thể sử dụng, các Button cần có văn bản. Các phần tử View cũng có thể có nền có thể là màu sắc hoặc hình ảnh .

Ngăn Attributes cung cấp quyền truy cập vào tất cả các thuộc tính bạn có thể gán cho một phần tử View. Bạn có thể nhập giá trị cho từng thuộc tính, chẳng hạn như các thuộc tính `android:id`, `background`, `textColor`, và `text attributes`.

Hình động sau đây minh họa cách thực hiện các bước này:

1. Sau khi chọn Button đầu tiên, hãy chỉnh sửa trường ID ở đầu ngăn Attributes thành `button_toast` cho thuộc tính `android:id`, được sử dụng để xác định phần tử trong bố cục.
2. Đặt thuộc tính `background` thành `@color/colorPrimary` . (Khi bạn nhập `@c` , các lựa chọn sẽ xuất hiện để bạn dễ dàng lựa chọn.)
3. Đặt thuộc tính `textColor` thành `@android:color/white` .
4. Chỉnh sửa thuộc tính `text` thành `Toast`



- Thực hiện các thay đổi thuộc tính tương tự cho Button thứ hai, sử dụng button_count làm ID, Count cho thuộc tính text và cùng màu cho nền và văn bản như các bước trước.

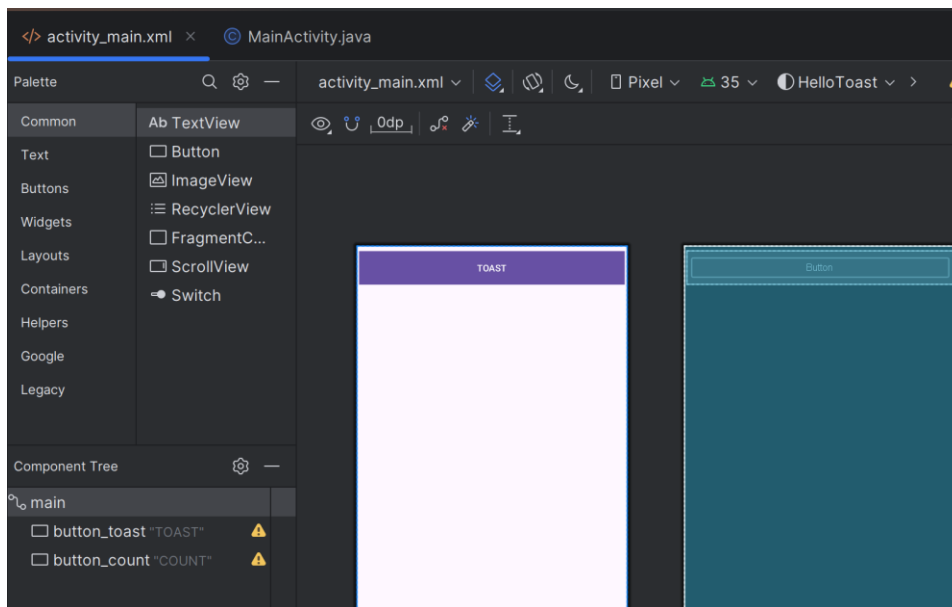
ColorPrimary là màu chính của chủ đề, một trong những màu cơ sở chủ đề được xác định trước trong tệp tài nguyên colors.xml. Nó được sử dụng cho thanh ứng dụng. Sử dụng màu cơ sở cho các thành phần UI khác sẽ tạo ra một UI thống nhất. Bạn sẽ tìm hiểu thêm về chủ đề ứng dụng và Material Design trong bài học khác.

Nhiệm vụ 4: Thêm Textedit và đặt các thuộc tính của nó

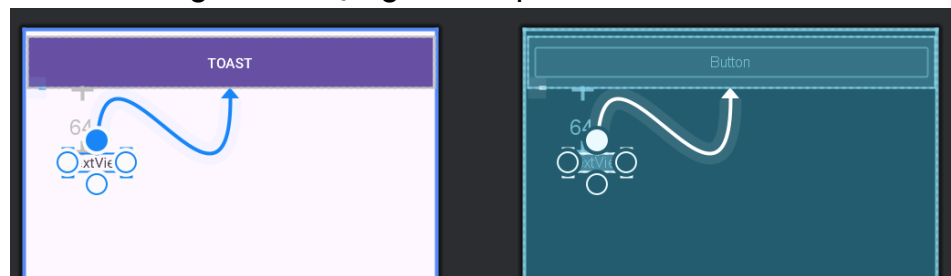
Một trong những lợi ích của ConstraintLayout là khả năng căn chỉnh hoặc hạn chế các phần tử liên quan đến các phần tử khác. Trong nhiệm vụ này, bạn sẽ thêm một TextView ở giữa bố cục và hạn chế nó theo chiều ngang vào lề và theo chiều dọc vào hai phần tử Button. Sau đó, bạn sẽ thay đổi các thuộc tính cho TextView trong ngăn Attributes (Thuộc tính).

4.1 Thêm TextView và các ràng buộc

- Như hình minh họa bên dưới, hãy kéo TextView từ ngăn Palette đến phần trên của bố cục và kéo một ràng buộc từ đầu TextView đến tay cầm ở cuối Nút Toast. Điều này hạn chế TextView nằm bên dưới Button .



- Như trong hình bên dưới, hãy kéo một ràng buộc từ cuối TextView đến tay cầm ở trên cùng của Nút count và từ hai bên của TextView đến hai bên của bố cục. Điều này hạn chế TextView ở giữa bố cục giữa hai phần tử Button.

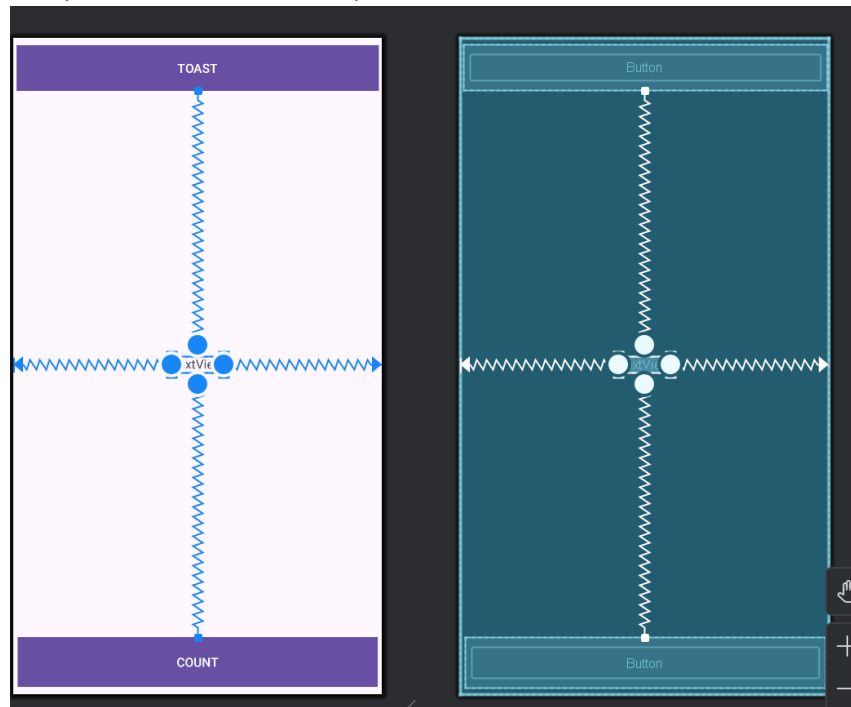


4.2 Đặt các thuộc tính TextView

Với TextView được chọn, hãy mở ngăn Attributes, nếu nó chưa mở. Đặt thuộc tính cho TextView như trong hình động bên dưới. Các thuộc tính bạn chưa gặp được giải thích sau hình:

- Đặt ID thành `show_count` .
- Đặt text thành `0` .
- Đặt `textSize` thành `160sp` .
- Đặt `textStyle` thành `B` (in đậm) và `textAlignment` thành `ALIGNCENTER` (căn giữa đoạn văn).
- Thay đổi các điều khiển kích thước chế độ xem theo chiều ngang và chiều dọc (`layout_width` và `layout_height`) thành `match_constraint` .
- Đặt `textColor` thành `@color/colorPrimary` .

7. Cuộn xuống ngăn và nhấp để mở rộng xem tất cả thuộc tính , cuộn xuống trang thuộc tính thứ hai đến background, sau đó nhập #FFFF00 để có màu vàng.
8. Cuộn xuống đến gravity, mở rộng gravity và chọn center_vertical (cho center-vertical).



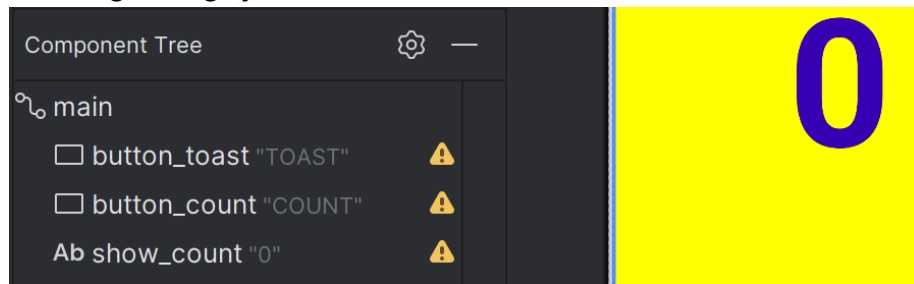
- **textSize:** Kích thước văn bản của TextView. Đối với bài học này, kích thước được đặt thành 160sp.Sp - chữ viết tắt của pixel không phụ thuộc tỷ lệ , và tương tự dp, là một đơn vị chia tỷ lệ theo mật độ màn hình và tùy chọn kích thước phông chữ của người dùng. Sử dụng đơn vị dp khi bạn chỉ định kích thước phông chữ để kích thước được điều chỉnh cho cả mật độ màn hình và sở thích của người dùng.
- **textStyle** và **textAlignment:** Kiểu văn bản, được đặt thành B (in đậm) trong bài học này và căn chỉnh văn bản, được đặt thành ALIGNCENTER(căn giữa đoạn văn).
- **gravity:** Thuộc tính gravity này chỉ định cách View được căn chỉnh trong View hoặc ViewGroup. Trong bước này, bạn căn giữa phần tử TextView theo chiều dọc trong ConstraintLayout.

Bạn có thể nhận thấy rằng thuộc tính background nằm trên trang đầu tiên của ngăn Attributes cho Button, nhưng trên trang thứ hai của ngăn Attributes cho TextView. Ngăn Attributes thay đổi cho từng loại View. Các thuộc tính phổ biến nhất cho View loại này xuất hiện trên trang đầu tiên và các thuộc tính còn lại

được liệt kê trên trang thứ hai. Để quay lại trang đầu tiên của ngăn Attributes , hãy bấm vào biểu tượng trên thanh công cụ ở đầu ngăn.

Nhiệm vụ 5: Chỉnh sửa bố cục trong XML

Bố cục ứng dụng Hello Toast gần như đã hoàn thành! Tuy nhiên, một dấu chấm than xuất hiện bên cạnh mỗi phần tử giao diện người dùng trong Cây thành phần. Di con trỏ của bạn qua các dấu chấm than này để xem thông báo cảnh báo, như được hiển thị bên dưới. Cảnh báo tương tự xuất hiện cho cả ba phần tử: chuỗi được mã hóa cứng nên sử dụng tài nguyên .



Cách dễ nhất để khắc phục sự cố bố cục là chỉnh sửa bố cục trong XML. Mặc dù trình soạn thảo bố cục là một công cụ mạnh mẽ, nhưng một số thay đổi dễ dàng thực hiện trực tiếp trong mã nguồn XML.

- 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