

TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN



GIÁO TRÌNH

THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG

Hà Nội, 2.2025

MỤC LỤC

| | | |
|-----------|---|----|
| CHƯƠNG 1. | LÀM QUEN | 4 |
| Bài 1) | Tạo ứng dụng đầu tiên | 4 |
| 1.1) | Android Studio và Hello World | 4 |
| a. | Cài đặt Android Studio | 5 |
| b. | Tạo ứng dụng Hello World | 6 |
| c. | Sử dụng thiết bị vật lý | 16 |
| d. | Thay đổi cấu hình Gradle của ứng dụng | 17 |
| e. | Thêm câu lệnh nhật ký vào ứng dụng của bạn | 18 |
| 1.2) | Giao diện người dùng tương tác đầu tiên | 20 |
| 1.3) | Trình chỉnh sửa bố cục | 27 |
| 1.4) | Văn bản và các chế độ cuộn | 32 |
| 1.5) | Tài nguyên có sẵn | 32 |
| Bài 2) | Activities | 32 |
| 2.1) | Activity và Intent | 32 |
| 2.2) | Vòng đời của Activity và trạng thái | 32 |
| 2.3) | Intent ngầm định | 32 |
| Bài 3) | Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ | 32 |
| 3.1) | Trình gỡ lỗi | 32 |
| 3.2) | Kiểm thử đơn vị | 32 |
| 3.3) | Thư viện hỗ trợ | 32 |
| CHƯƠNG 2. | TRẢI NGHIỆM NGƯỜI DÙNG | 33 |
| Bài 1) | Tương tác người dùng | 33 |
| 1.1) | Hình ảnh có thể chọn | 33 |
| 1.2) | Các điều khiển nhập liệu | 33 |
| 1.3) | Menu và bộ chọn | 33 |
| 1.4) | Điều hướng người dùng | 33 |

| | | |
|-----------|--|----|
| 1.5) | RecyclerView..... | 33 |
| Bài 2) | Trải nghiệm người dùng thú vị | 33 |
| 2.1) | Hình vẽ, định kiểu và chủ đề | 33 |
| 2.2) | Thẻ và màu sắc | 33 |
| 2.3) | Bố cục thích ứng..... | 33 |
| Bài 3) | Kiểm thử giao diện người dùng | 33 |
| 3.1) | Espresso cho việc kiểm tra UI..... | 33 |
| CHƯƠNG 3. | LÀM VIỆC TRONG NỀN..... | 33 |
| Bài 1) | Các tác vụ nền | 33 |
| 1.1) | AsyncTask..... | 33 |
| 1.2) | AsyncTask và AsyncTaskLoader | 33 |
| 1.3) | Broadcast receivers | 33 |
| Bài 2) | Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền | 33 |
| 2.1) | Thông báo..... | 33 |
| 2.2) | Trình quản lý cảnh báo | 33 |
| 2.3) | JobScheduler | 33 |
| CHƯƠNG 4. | LƯU DỮ LIỆU NGƯỜI DÙNG | 34 |
| Bài 1) | Tùy chọn và cài đặt | 34 |
| 1.1) | Shared preferences | 34 |
| 1.2) | Cài đặt ứng dụng | 34 |
| Bài 2) | Lưu trữ dữ liệu với Room | 34 |
| 2.1) | Room, LiveData và ViewModel | 34 |
| 2.2) | Room, LiveData và ViewModel | 34 |

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

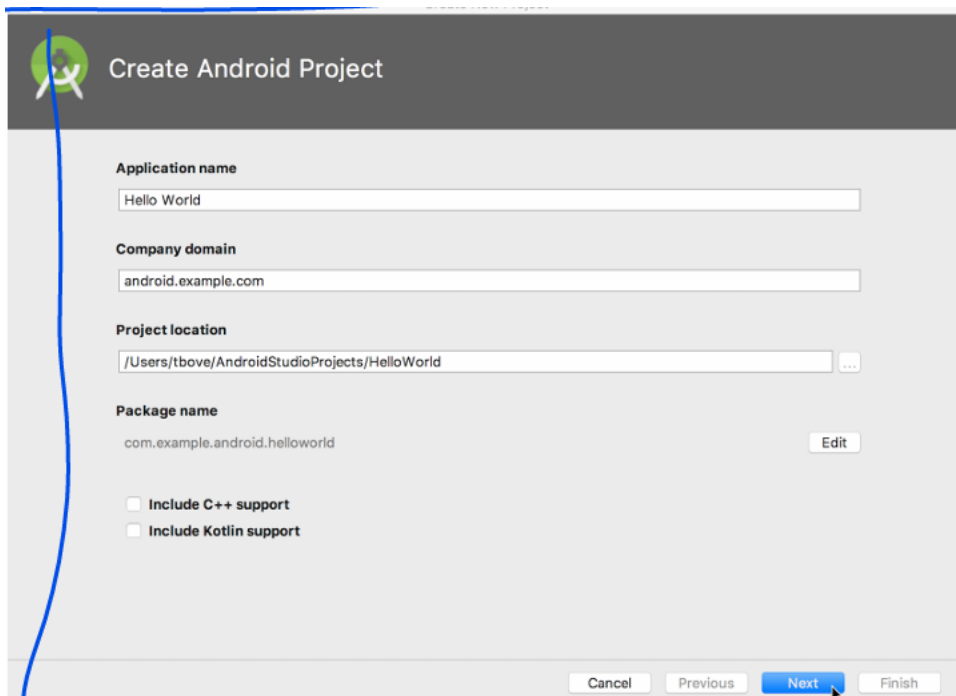
Giới thiệu

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 ứng dụng

Sau khi cài đặt thành công Android Studio, bạn sẽ tạo một 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 ảo hoặc vật lý Android.

a. Cài đặt Android Studio

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

Android Studio có sẵn cho máy tính chạy Windows hoặc Linux và máy Mac chạy macOS. OpenJDK (Java Development Kit) mới nhất được đi kèm với Android Studio.

Để thiết lập và chạy Android Studio, trước tiên hãy kiểm tra các [yêu cầu hệ thống](#) để đảm bảo hệ thống của bạn đáp ứng các yêu cầu đó. Việc cài đặt tương tự cho tất cả các nền tảng. Bất kỳ sự khác biệt nào đều được lưu ý bên dưới.

1. Truy cập trang web của [nhà phát triển Android](#) và làm theo hướng dẫn để tải về 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 được chọn để cài đặt.

3. Sau khi cài đặt xong, Trình hướng dẫn Cài đặt sẽ tải xuống và cài đặt một số thành phần bổ sung bao gồm Android SDK. Hãy kiên nhẫn, điều này có thể mất một thời gian tùy thuộc vào tốc độ Internet của bạn, và một số bước có thể dường như lặp lại.

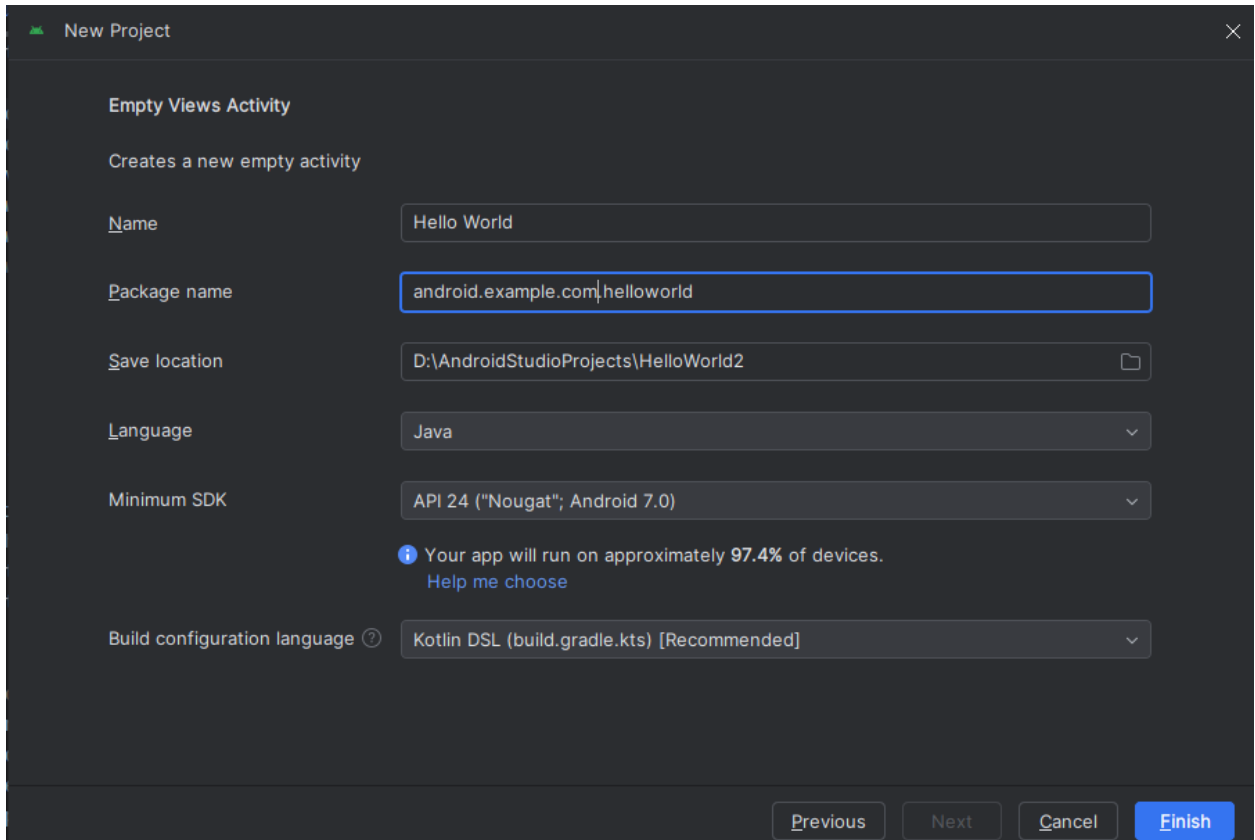
4. Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động, và bạn đã sẵn sàng để tạo dự án đầu tiên của mình.

b. Tạo ứng dụng Hello World

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

i. Tạo dự án ứng dụng

1. Mở Android Studio nếu nó chưa được mở.
2. Trong cửa sổ **New Project**, chọn **Phone and Tablet**, trong Phone and Tablet chọn **Empty Views Activity** và nhấn **Next**.
3. Tại mục Name, nhập Hello World cho Application name.



4. Xác minh rằng **Save location** của Dự án là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án khác của Android Studio, hoặc thay đổi nó thành thư mục ưa thích của bạn.
5. Chấp nhận mặc định **android.example.com** cho **Company Domain** , hoặc tạo một tên miền công ty duy nhất.

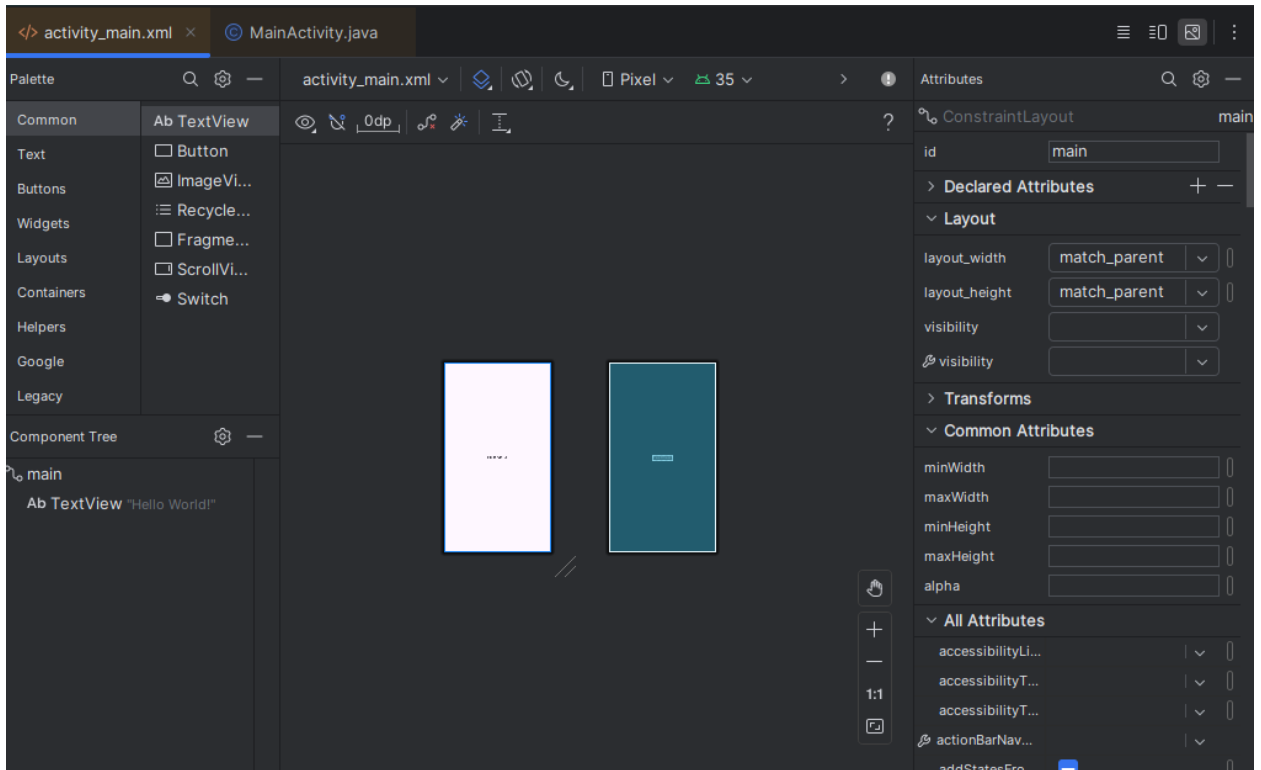
Nếu bạn không có kế hoạch công bố ứng dụng của mình, bạn có thể chấp nhận mặc định. Hãy nhớ rằng việc thay đổi tên gói của ứng dụng sau này là công việc phụ.

6. Chọn ngôn ngữ **Java** và nhấn **Finish**.

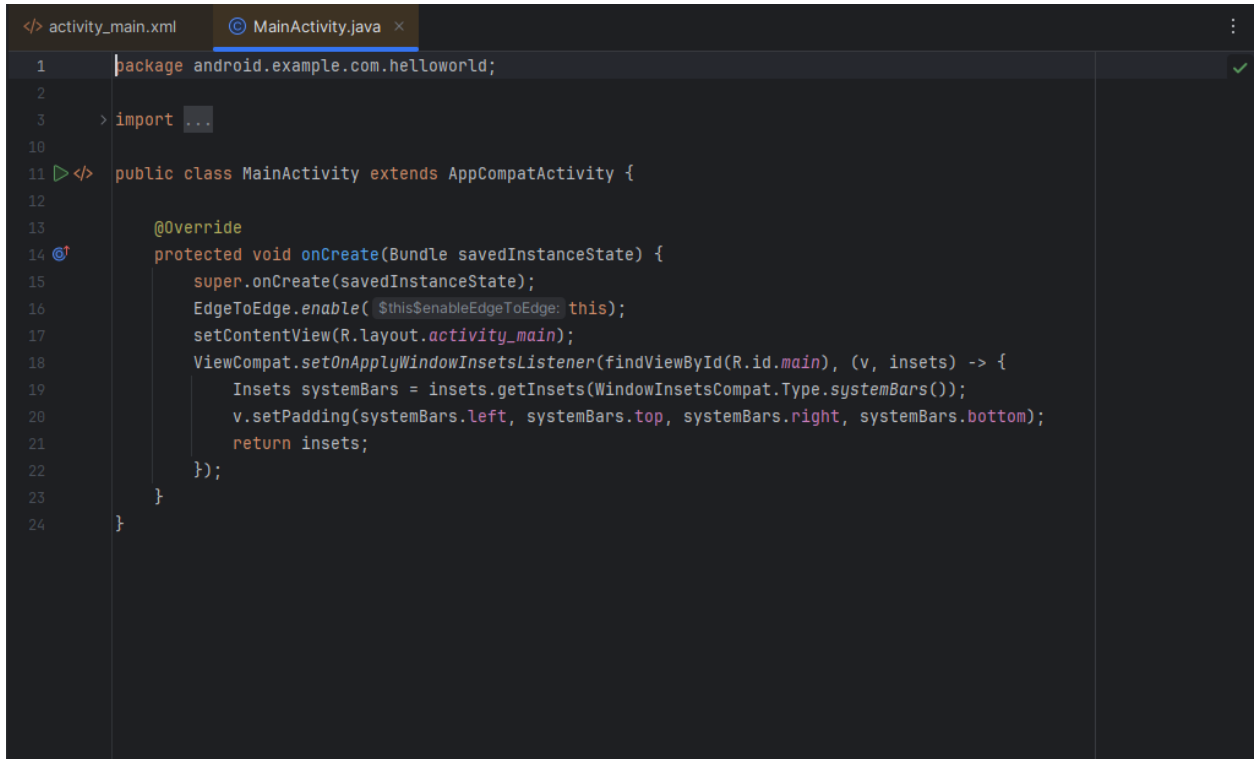
Android Studio tạo ra một thư mục cho các dự án của bạn và xây dựng dự án với Gradle (điều này có thể mất một chút thời gian).

Trình thiết kế Android Studio xuất hiện. Thực hiện theo các bước sau:

1. Nhấn the **activity_main.xml** tab để xem layout editor.
2. Nhấp vào tab Layout Editor Design, nếu chưa được chọn, để hiển thị biểu diễn đồ họa của bố cục như hình bên dưới.



3. Nhấp vào tab **MainActivity.java** để xem trình chỉnh sửa code như hiển thị bên dưới.



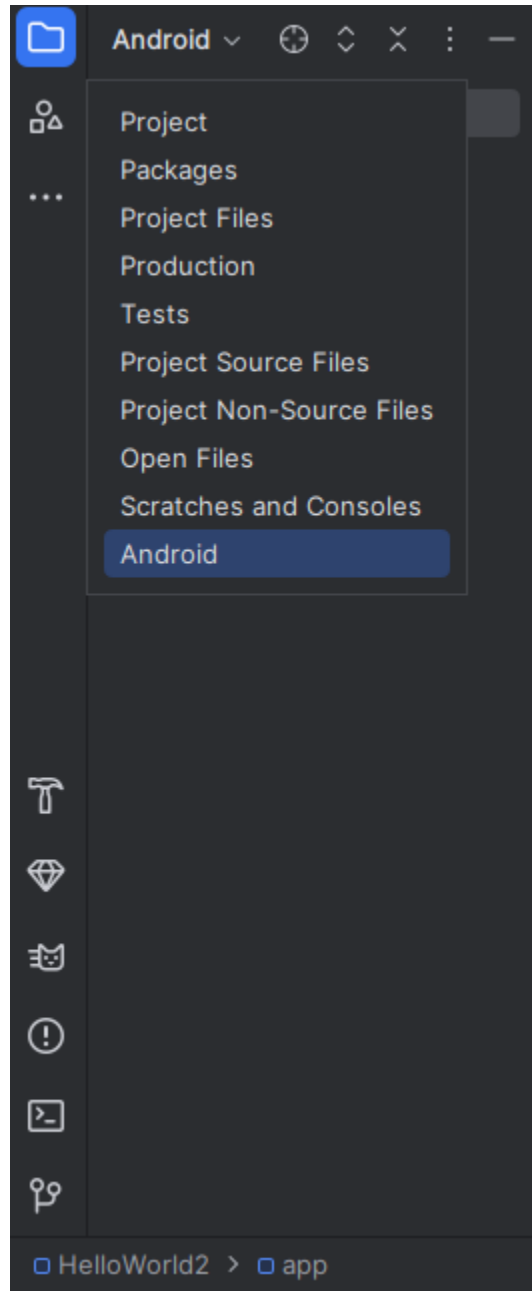
The screenshot shows the MainActivity.java file in an Android Studio editor. The code is as follows:

```
1 package android.example.com.helloworld;
2
3 > import ...
10
11 <> public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         EdgeToEdge.enable(this);
17         setContentView(R.layout.activity_main);
18         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
19             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
20             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
21             return insets;
22         });
23     }
24 }
```

ii. Khám phá dự án > Android pane

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

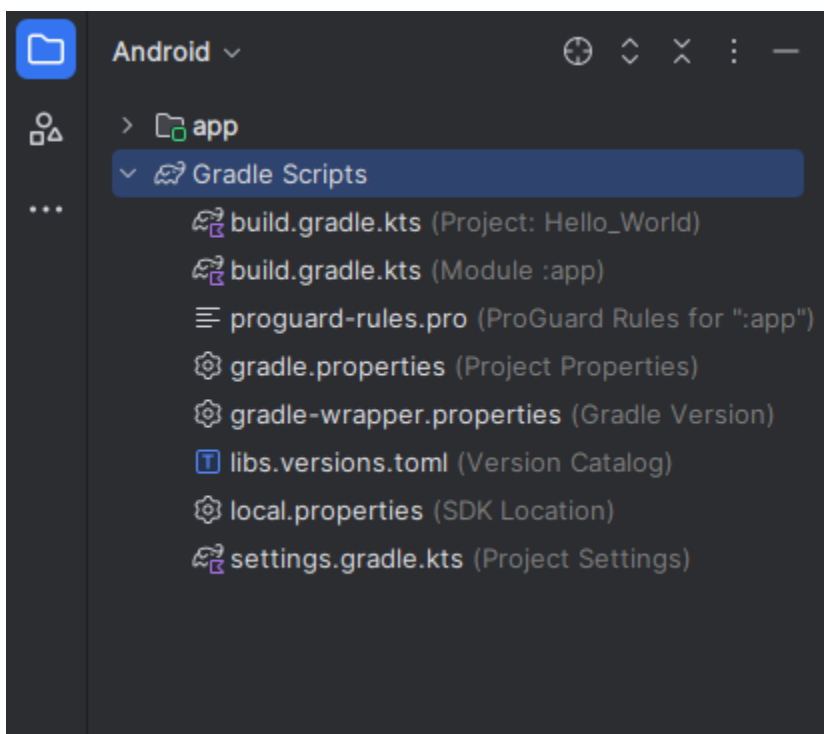
1. Nếu chưa chọn, hãy nhấp vào tab Project trong cột tab dọc ở bên trái cửa sổ Android Studio. Ngăn Dự án xuất hiện.
2. Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, hãy chọn Android từ menu bật lên ở đầu ngăn Dự án, như được hiển thị bên dưới.



iii. Khám phá thư mục Gradle Scripts

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 bên dưới.



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

1. Nếu thư mục Tập lệnh Gradle không được mở rộng, hãy nhấp vào hình dấu mũ để mở rộng thư mục đó. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng.
2. Tìm tệp **settings.gradle**(Project Settings).

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 nó vẫn hữu ích để 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 Google Maven) là vị trí kho lưu trữ:

```
repositories {  
    google {  
        content {  
            includeGroupByRegex( groupRegex: "com\\.android.*")  
            includeGroupByRegex( groupRegex: "com\\.google.*")  
            includeGroupByRegex( groupRegex: "androidx.*")  
        }  
    }  
    mavenCentral()  
    gradlePluginPortal()  
}
```

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 build.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 phần phụ thuộc. 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 thực hiện `fileTree(dir: 'libs', include: ['*.jar'])` thêm phần phụ thuộc của tất cả các tệp ".jar" bên trong thư mục libs. Sau đây là tệp build.gradle(Module:app) cho ứng dụng HelloWorld:

```

plugins {
    alias(libs.plugins.android.application)
}

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

    > defaultConfig {...}

    > buildTypes {...}
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

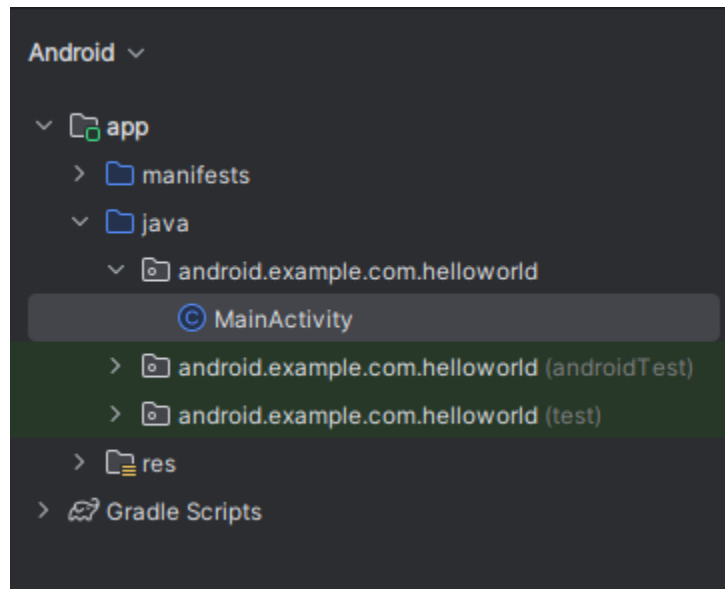
dependencies {
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
     androidTestImplementation(libs.espresso.core)
}

```

iv. Khám phá ứng dụng và thư mục res

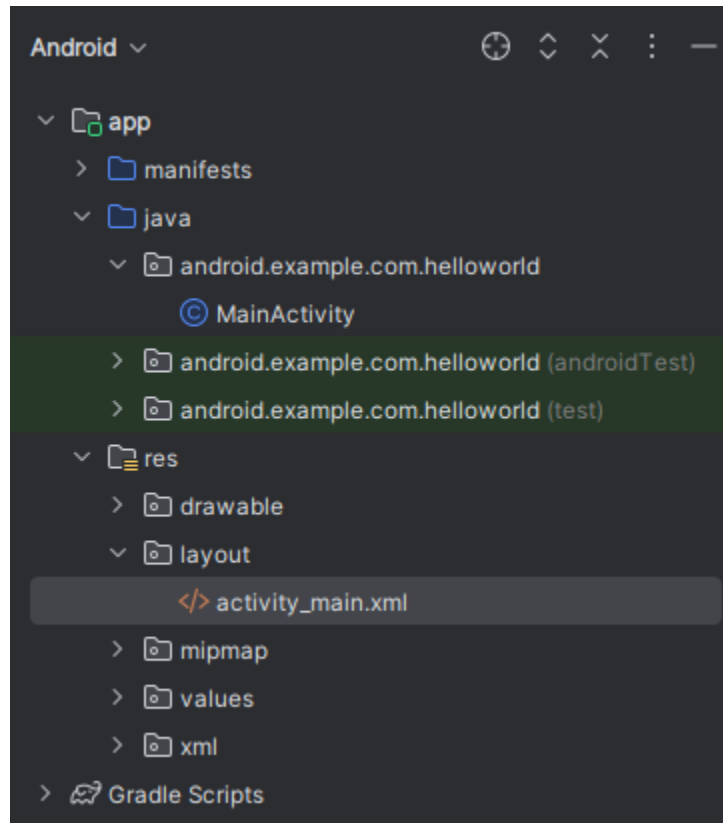
Tất cả mã và tài nguyên cho ứng dụng nằm trong thư mục ứng dụng và res.

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 sẽ mở tệp đó 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 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 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 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**, đồng thời nhấp đúp vào tệp **activity_main.xml** để mở nó trong trình chỉnh sửa bố cục



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

v. Khám phá 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. 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 khóa học khác, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để biết thông tin giới thiệu, hãy xem **App Manifest Overview**.

c. 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 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 [Using Hardware Devices](#). 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 [OEM USB Drivers](#).

i. 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 nhà phát triển** trên thiết bị của bạn.

Trên Android 4.2 trở lên, màn hình **Tùy chọn nhà phát triển** 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, mở **Cài đặt**, tìm kiếm **Giới thiệu về điện thoại**, nhấp vào Giới thiệu về điện thoại và nhấn vào **Số bản dựng** (building number) bảy lần.
2. Quay lại màn hình trước đó (Cài đặt / Hệ thống). Tùy chọn nhà phát triển xuất hiện trong danh sách. Nhấn vào Tùy chọn nhà phát triển.
3. Chọn USB Debugging.

ii. Chạy ứng dụng trên thiết bị

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

1. Connect your device to your development machine with a USB cable.
2. Bấm vào nút Run trên thanh công cụ. Cửa sổ Select Deployment Target sẽ 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 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:

1. Rút phích cắm và cắm lại thiết bị của bạn.
2. 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:

1. Rút phích cắm của thiết bị
2. Trên thiết bị, mở Tùy chọn nhà phát triển trong ứng dụng Cài đặt.
3. Nhấn vào Thu hồi ủy quyền gỡ lỗi USB
4. Kết nối lại thiết bị với máy tính của bạn.
5. Khi được nhắc, hãy cấp ủy quyền.

Bạn có thể cần cài đặt trình điều khiển USB thích hợp cho thiết bị của mình. Xem **Using Hardware Devices documentation**.

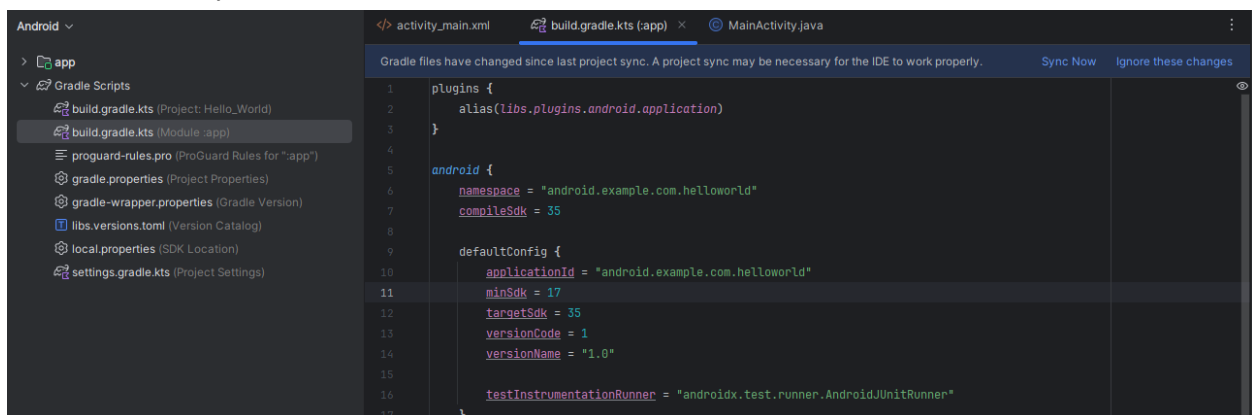
d. Thay đổi cấu hình Gradle của ứng dụng

Trong tác vụ này, bạn sẽ thay đổi điều gì đó về cấu hình ứng dụng trong tệp **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.

i. Thay đổi phiên bản SDK tối thiểu cho ứng dụng

Các bước:

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 xuất hiện trong trình soạn thảo mã.
2. Trong khối `defaultConfig`, hãy thay đổi giá trị của `minSdkVersion` thành 17 như hình dưới đây (ban đầu nó được đặt thành 15).



Trình chỉnh sửa mã hiển thị thanh thông báo ở trên cùng với liên kết **Sync now**

ii. Đồng bộ hóa cấu hình Gradle mới

Khi bạn thực hiện các thay đổi đối với các 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 về 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 hóa các tệp dự án, hãy bấm vào **Sync now** trong thanh thông báo xuất hiện khi thực hiện thay đổi

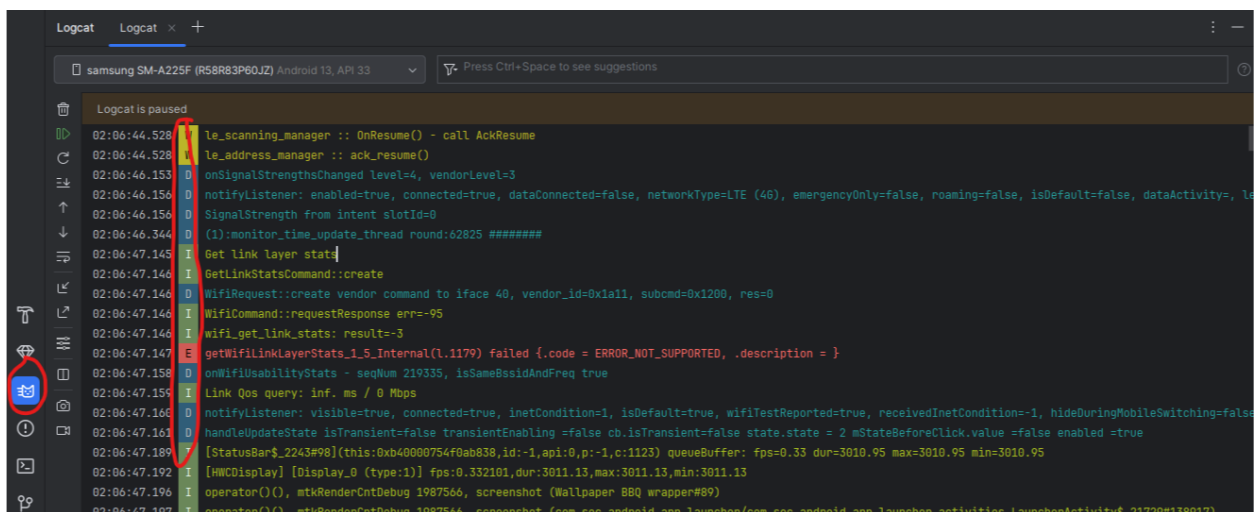
Khi quá trình đồng bộ hóa Gradle kết thúc, thông báo Bản dựng Gradle đã hoàn tất sẽ xuất hiện ở góc dưới cùng bên trái của cửa sổ Android Studio.

e. Thêm câu lệnh nhật ký vào ứng dụng của bạn

Trong nhiệm vụ này, bạn sẽ thêm câu lệnh Nhật ký vào ứng dụng của mình, cụm từ này hiển thị thông báo trong ngăn Logcat. Thông báo nhật ký 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 ngoại lệ.

i. Xem ngăn Logcat

Để xem ngăn Logcat, hãy nhấp vào thẻ Logcat ở cuối cửa sổ Android Studio như trong hình bên dưới.



Trong hình trên:

1. Tab Logcat để mở và đóng ngăn Logcat, hiển thị thông tin về ứng dụng của bạn khi ứng dụng đang chạy. Nếu bạn thêm câu lệnh Nhật ký vào ứng dụng của mình, thông báo Nhật ký sẽ xuất hiện ở đây.
2. Menu Mức nhật ký được đặt thành Verbose (mặc định), hiển thị tất cả các thông báo Nhật ký. Các cài đặt khác bao gồm **Debug** , **Error** , **Info** , and **Warn** .

ii. Thêm câu lệnh nhật ký vào ứng dụng của bạn

Câu lệnh nhật ký trong mã ứng dụng của bạn sẽ hiển thị thông báo trong ngăn Logcat.

Ví dụ:

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

Các phần của thông điệp là:

- **Log** : Lớp Log để gửi thông báo nhật ký đến ngăn Logcat.
- **d** : Cài đặt mức nhật ký Debug để lọc thông báo nhật ký hiển thị trong ngăn Logcat. Các cấp độ nhật ký khác là **E** cho Error , **W** cho Warn và **I** cho 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 Hoạt động 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 nhật ký được định nghĩa là hằng số cho Hoạt động:

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

- **"Hello world"** : Đối số thứ hai là thông điệp thực tế.

Làm theo 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 **android.util.Log** cần thiết để sử dụng Log), hãy chọn **File > Settings** trong Windows hoặc Tùy chọn **Android Studio > Preferences** trong macOS.
3. Chọn **Editor > General > Auto Import**. Chọn tất cả các hộp kiểm và **Insert imports on paste to All**.
4. Nhấp vào **Apply** và sau đó nhấp vào **OK**
5. Trong phương thức onCreate() của MainActivity , thêm câu lệnh sau

```
Log.v("main", "Hello World");
```

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

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    Log.v(tag: "main", msg: "Hello World");
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
}
```

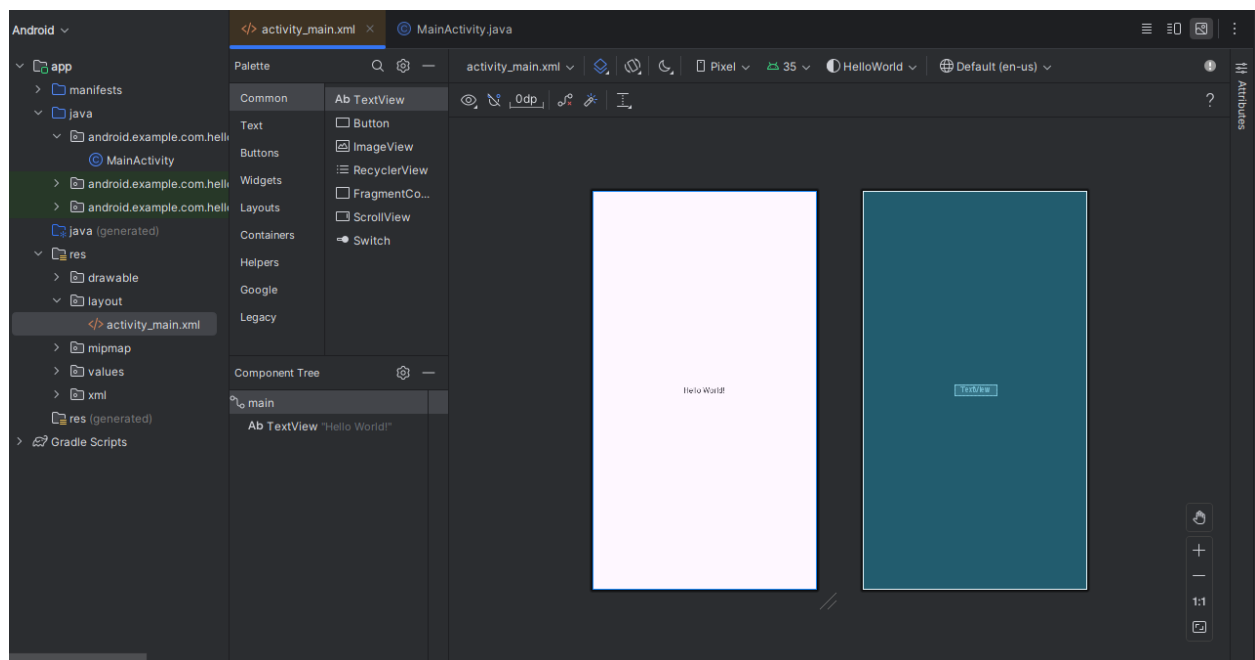
6. Nếu ngăn Logcat chưa mở, hãy nhấp vào tab Logcat ở cuối Android Studio để mở.
7. Kiểm tra xem tên của mục tiêu và tên gói của ứng dụng có chính xác không.
8. Thay đổi mức nhật ký trong ngăn Logcat thành Debug (hoặc để nguyên ở chế độ Verbose vì có rất ít thông báo nhật ký).
9. Chạy ứng dụng của bạn.

2025-02-28 07:53:27.297 26270-26270 main android.example.com.helloworld V Hello World

1.2) Giao diện người dùng tương tác đầu tiên

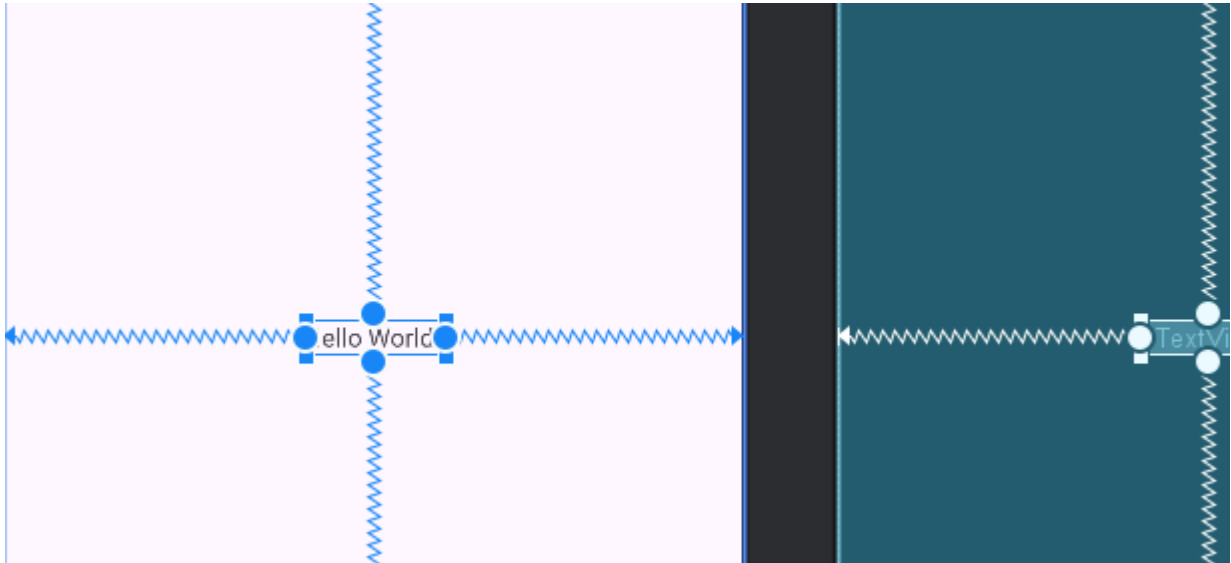
a. Tạo và khám phá một dự án mới

Trình chỉnh sửa bố cục



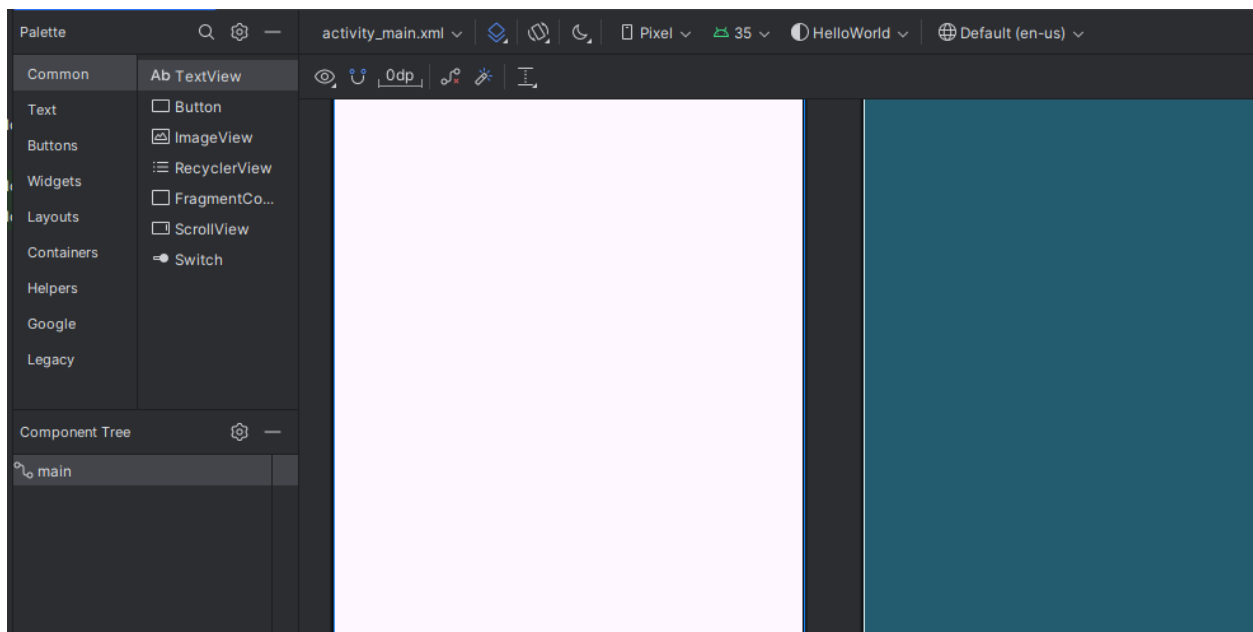
b. Thêm các phần tử View vào trình chỉnh sửa bố cục

i. Kiểm tra ràng buộc phần tử

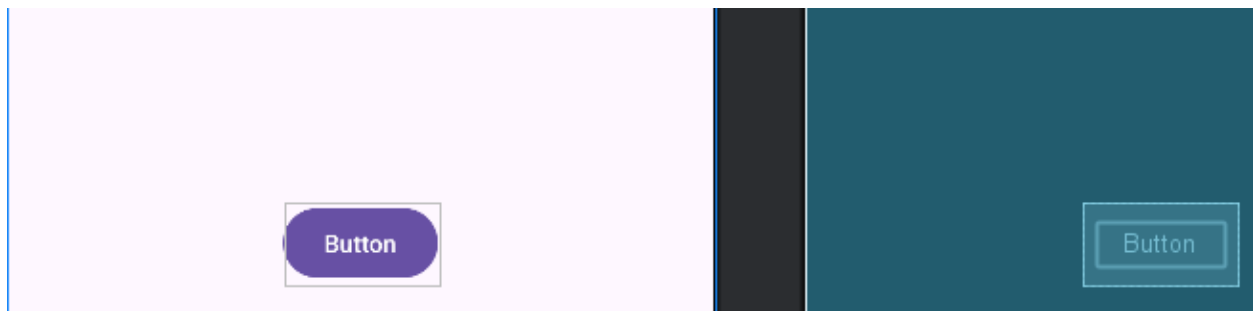


ii. Thêm một nút vào bố cục

Bố cục trống

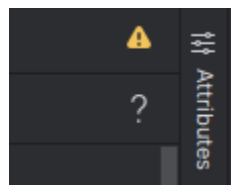
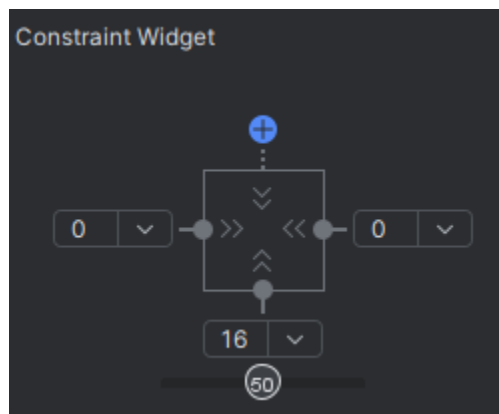


Thêm nút bấm

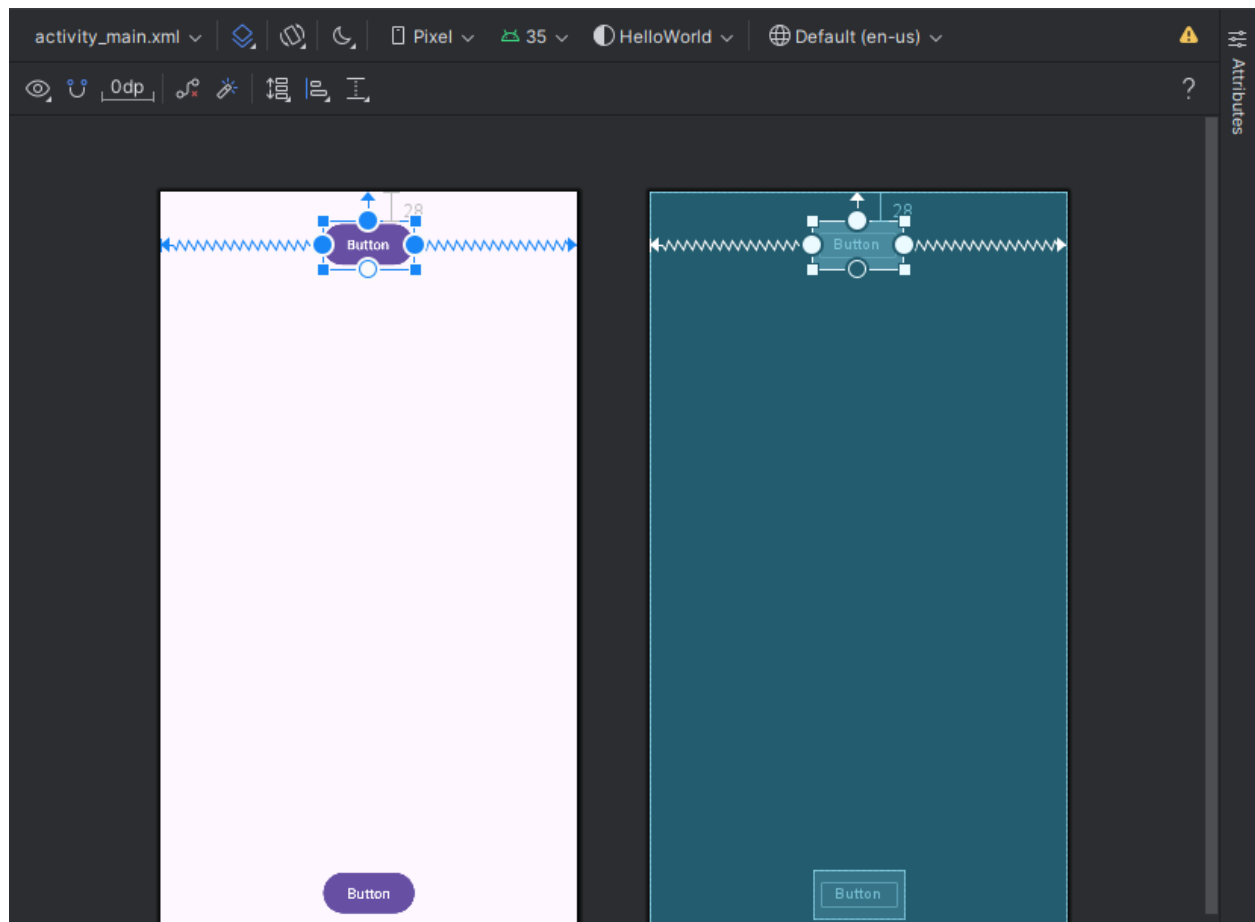


c. Thay đổi thuộc tính của phần tử giao diện người dung

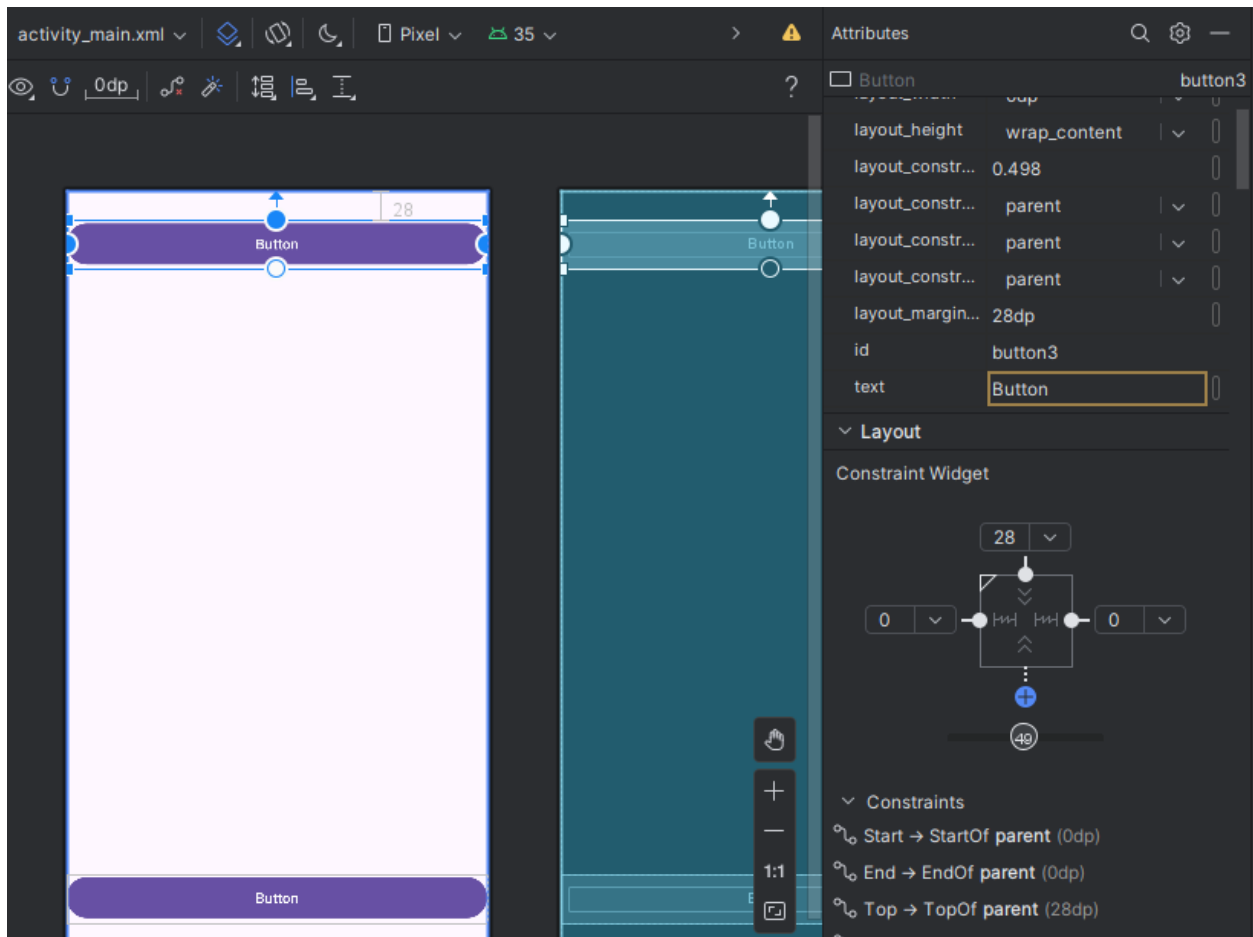
i. Thay đổi kích cỡ nút



Thêm 2 nút bấm

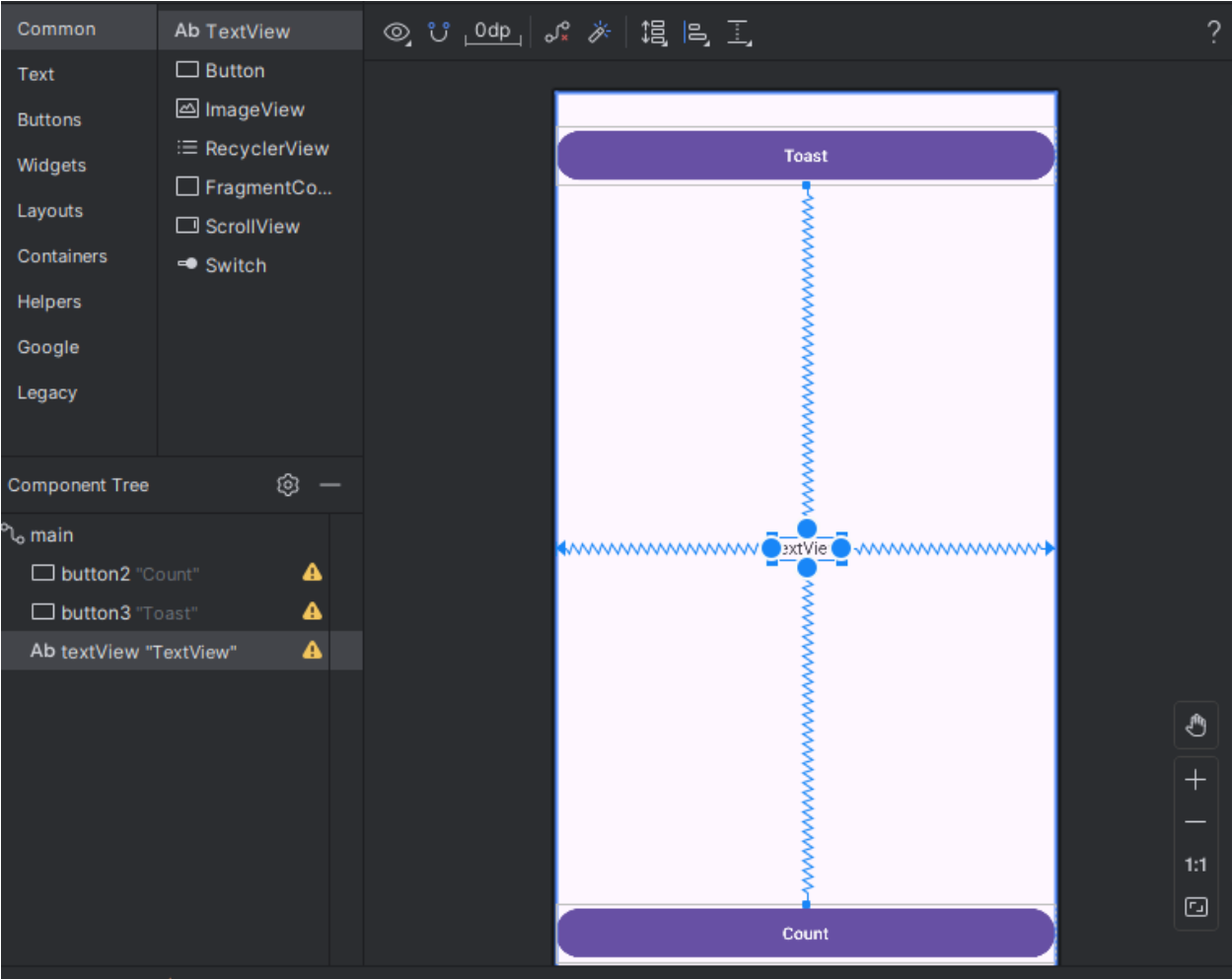


Thay đổi kích cỡ nút

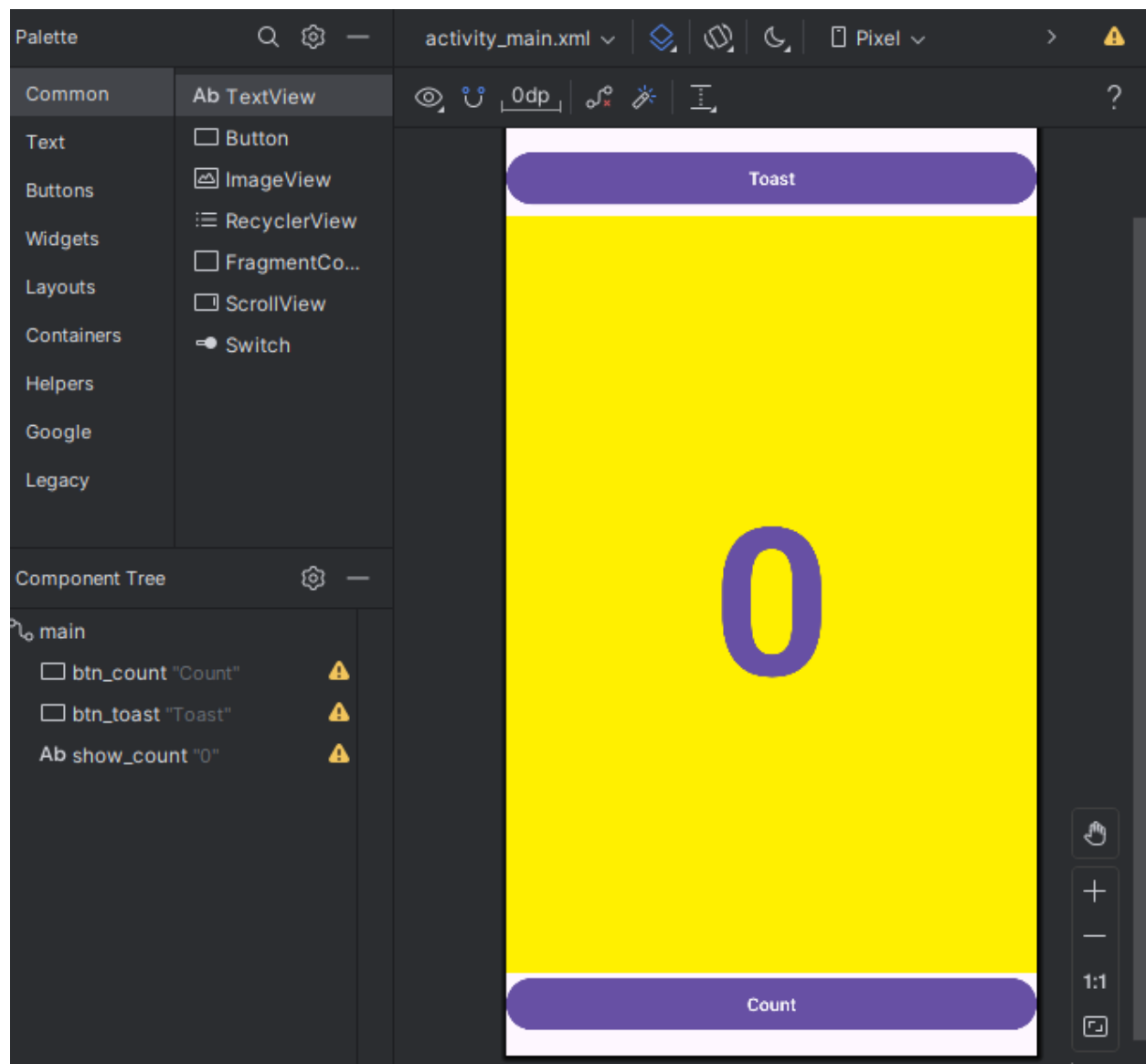


d. Thêm một **TextEdit** và đặt thuộc tính cho nó

Thêm **TextView** vào màn hình và kết nối bên trên của nó với bên dưới của nút Toast, bên dưới của nó với bên trên của nút Count



ii. Thay đổi thuộc tính TextView



e. Thiết kế bố cục trong XML

i. Mở mã XML cho bố cục

```
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
20 <Button
22     android:layout_width="0dp"
23     android:layout_height="wrap_content"
24     android:layout_marginTop="28dp"
25     android:text="Toast"
26     app:layout_constraintEnd_toEndOf="parent"
27     app:layout_constraintHorizontal_bias="0.498"
28     app:layout_constraintStart_toStartOf="parent"
29     app:layout_constraintTop_toTopOf="parent" />
30
31 <TextView
32     android:id="@+id/show_count"
33     android:layout_width="0dp"
34     android:layout_height="0dp"
35     android:layout_marginTop="5dp"
36     android:background="#FFF000"
37     android:gravity="center_horizontal"
38     android:text=""
39     android:textAlignment="center"
40     android:textColor="?attr/colorPrimary"
41     android:textSize="160sp"
42     android:textStyle="bold"
43     app:layout_constraintBottom_toTopOf="@id/btn_count"
44     app:layout_constraintEnd_toEndOf="parent"
45     app:layout_constraintStart_toStartOf="parent"
46     app:layout_constraintTop_toBottomOf="@id/btn_toast" />
47 </androidx.constraintlayout.widget.ConstraintLayout>
```

ii. Trích xuất tài nguyên chuỗi

```
<resources>
    <string name="app_name">Hello World</string>
    <string name="button_label_toast">Toast</string>
    <string name="button_label_count">Count</string>
    <string name="count_initial_value">0</string>
</resources>
```

```
<resources>
    <string name="app_name">Hello World</string>
    <string name="button_label_toast">Toast</string>
    <string name="button_label_count">Count</string>
    <string name="count_initial_value">0</string>
    <string name="toast_message">Hello Toast!</string>
</resources>
```

f. Thêm bộ điều khiển onClick cho nút

i. Thêm thuộc tính onClick và bộ điều khiển cho mỗi nút

```
android:onClick="showToast" />
```

```
android:onClick="countUp" />
```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity" >

    <Button
        android:id="@+id/btn_count"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/button_label_count"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:onClick="countUp" />

    <Button
        android:id="@+id/btn_toast"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="28dp"
        android:text="@string/button_label_toast"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:onClick="showToast" />

    <TextView
        android:id="@+id/show_count"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="5dp"
        android:background="#FFF000"
        android:gravity="center_horizontal"
        android:text="@string/count_initial_value"
        android:textAlignment="center"
        android:textColor="?attr/colorPrimary"
        android:textSize="160sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@id/btn_count"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btn_toast" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Mở tệp MainActivity.java

```

package android.example.com.helloworld;

import android.os.Bundle;
import android.util.Log;
import android.view.View;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;

```

```
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        Log.v("main", "Hello World");
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets)
-> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

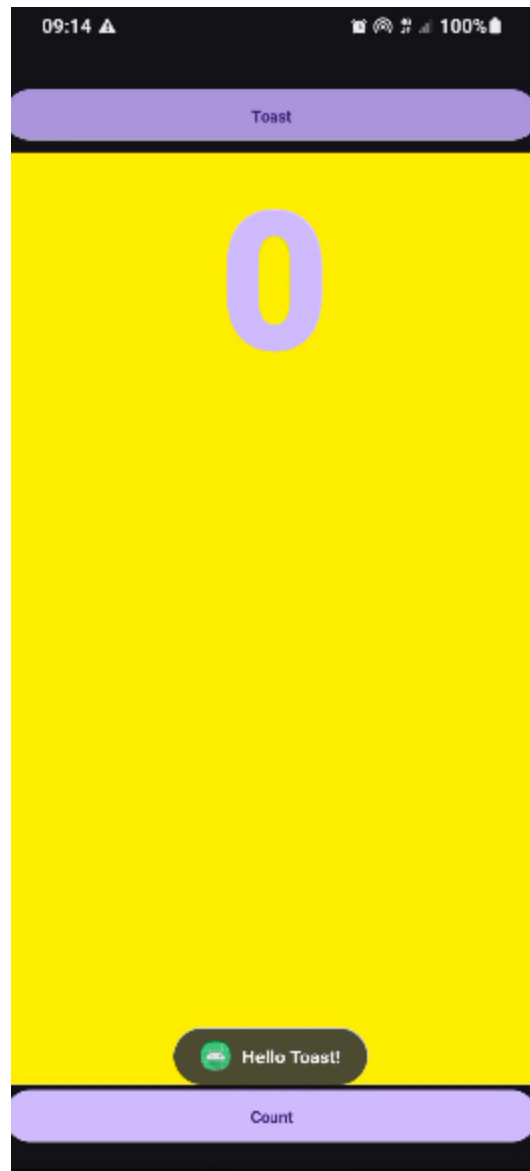
        public void showToast(View view) {
        }

        public void countUp(View view) {
        }
    }
}
```

ii. Thiết kế bộ điều khiển cho Toast Button

```
public void showToast(View view) {
    Toast toast = Toast.makeText(this, R.string.toast_message, Toast.LENGTH_SHORT);
    toast.show();
}
```

Chạy chương trình



iii. Thiết kế bộ điều khiển cho Count Button

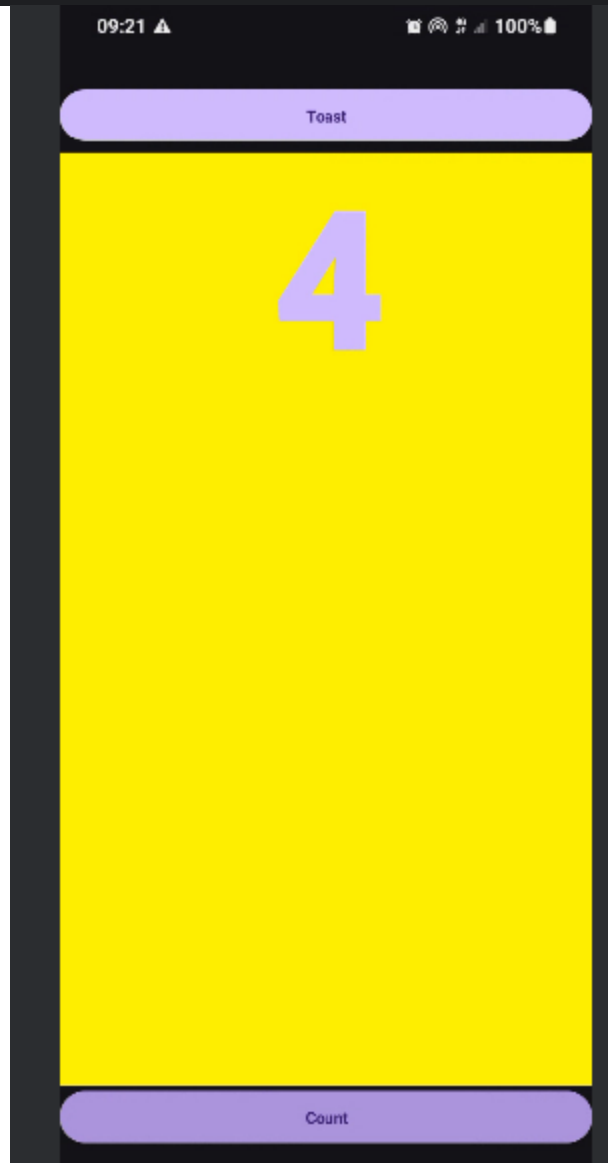
```
public void countUp(View view) {  
    mCount++;  
}
```

```
public class MainActivity extends AppCompatActivity {  
  
    private int mCount = 0;  
    private TextView mShowCount;
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable(this);
```

```
setContentView(R.layout.activity_main);  
mShowCount = (TextView) findViewById(R.id.show_count);
```

```
public void countUp(View view) {  
    mCount++;  
    if (mShowCount != null)  
        mShowCount.setText(Integer.toString(mCount));  
}
```



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