

Nhiệm vụ 1: Cài đặt Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp (IDE) hoàn chỉnh, bao gồm trình soạn thảo mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó bao gồm các công cụ để phát triển, gỡ lỗi, kiểm tra và hiệu suất, giúp phát triển ứng dụng nhanh hơn và dễ dàng hơn. Bạn có thể sử dụng rất nhiều trình giả lập được cấu hình sẵn hoặc thử nghiệm ứng dụng của mình trên thiết bị di động của riêng bạn, xây dựng ứng dụng sản xuất và xuất bản chúng trên Cửa hàng Google Play.

Nhiệm vụ 2: Tạo ứng dụng hello word

2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu chưa mở

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

3. Trong cửa sổ Create Android Project , Nhập Hello World cho tên ứng dụng

4. Xác minh rằng 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 thành thư mục bạn muốn.

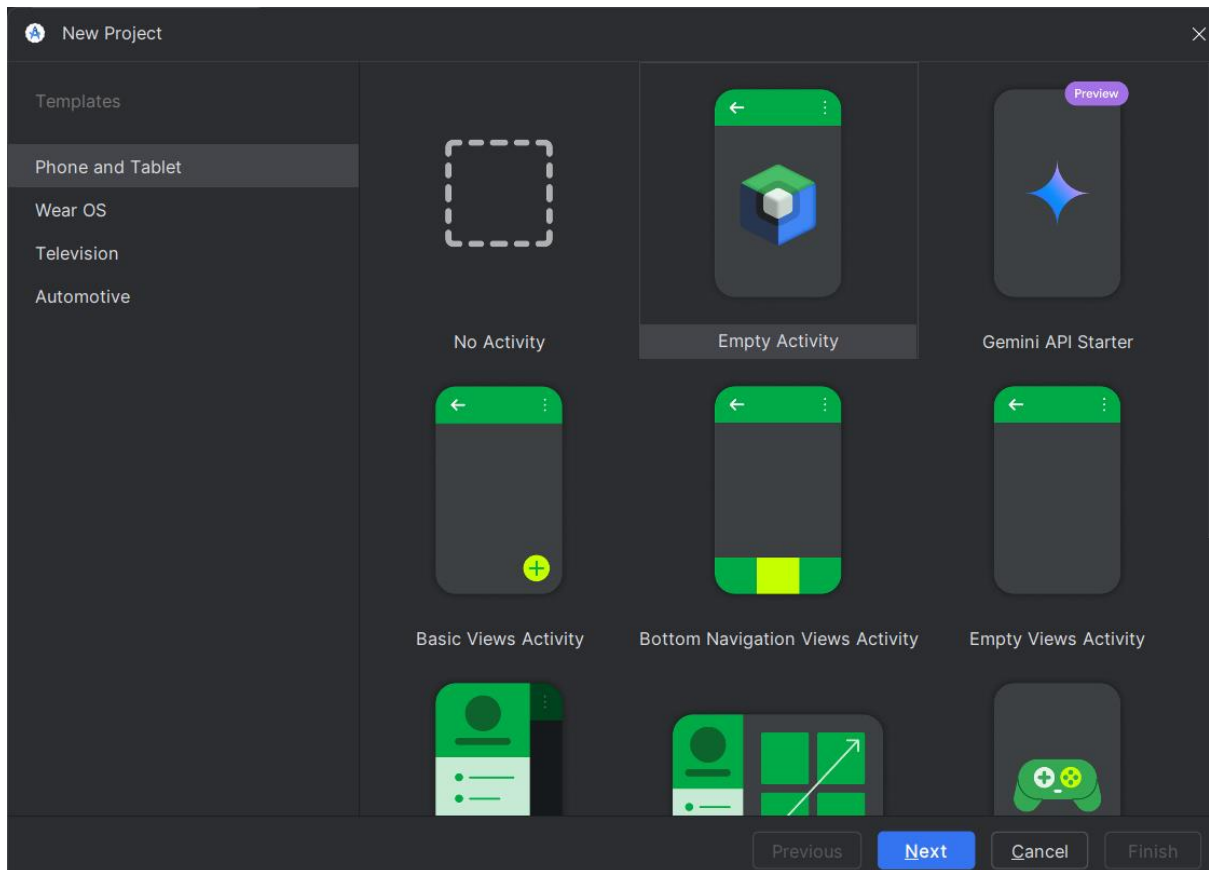
5. Chấp nhận mặc định là android.example.com cho Tên miền công ty hoặc tạo một tên miền công ty duy nhất. Nếu bạn không có kế hoạch phát hành ứng dụng, bạn có thể chấp nhận mặc định. Lưu ý rằng việc thay đổi tên gói ứng dụng sau này sẽ tốn thêm công sức.

6. Bỏ chọn các tùy chọn I nclude C++ support và I nclude Kotlin support, rồi nhấp vào Tiếp theo.

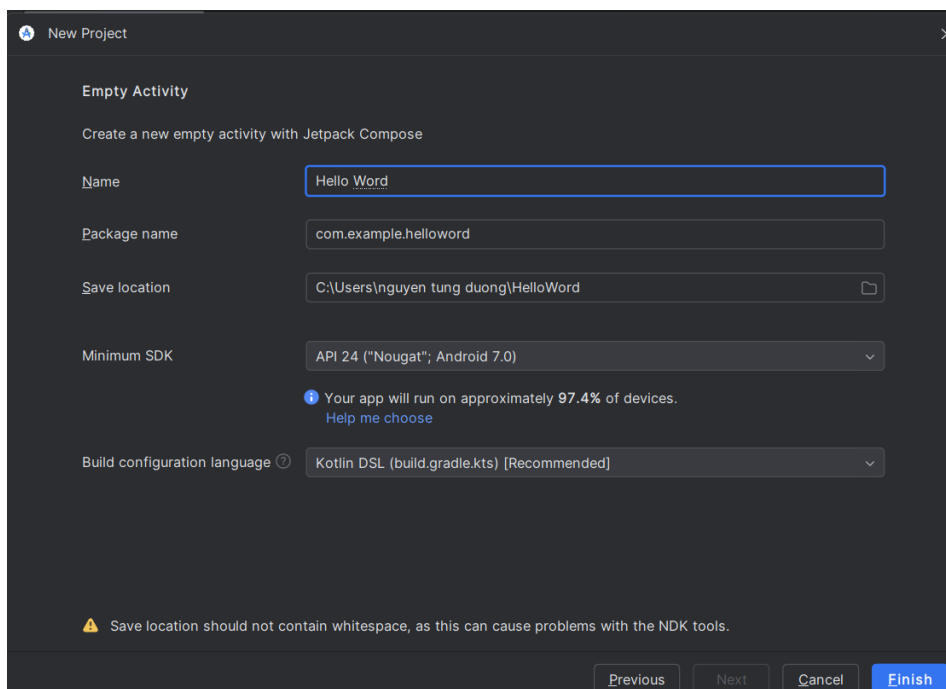
7. Trên màn hình Target Android Devices, Điện thoại và Máy tính bảng phải được chọn. Đảm bảo rằng API 15: Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu; nếu không, hãy sử dụng menu bật lên để đặt.

8. Bỏ chọn mục I nclude Instant App support và tất cả các tùy chọn khác. Sau đó, nhấp vào Next. 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 Add an Activity xuất hiện. Một Activity là một việc duy nhất, tập trung mà người dùng có thể thực hiện. Đây là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một Activity thường có một bố cục liên kết với nó để xác định cách các thành phần UI xuất hiện trên màn hình. Android Studio cung cấp các mẫu Activity để giúp bạn bắt đầu. Đối với dự án Hello World, hãy chọn Empty Activity như được hiển thị bên dưới 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 Tạo tệp bố cục được chọn. Tên bố cục theo mặc định là `activity_main`. Bạn có thể thay đổi tùy chọn này nếu muốn, nhưng bài học này sử dụng `ctivity_main`.

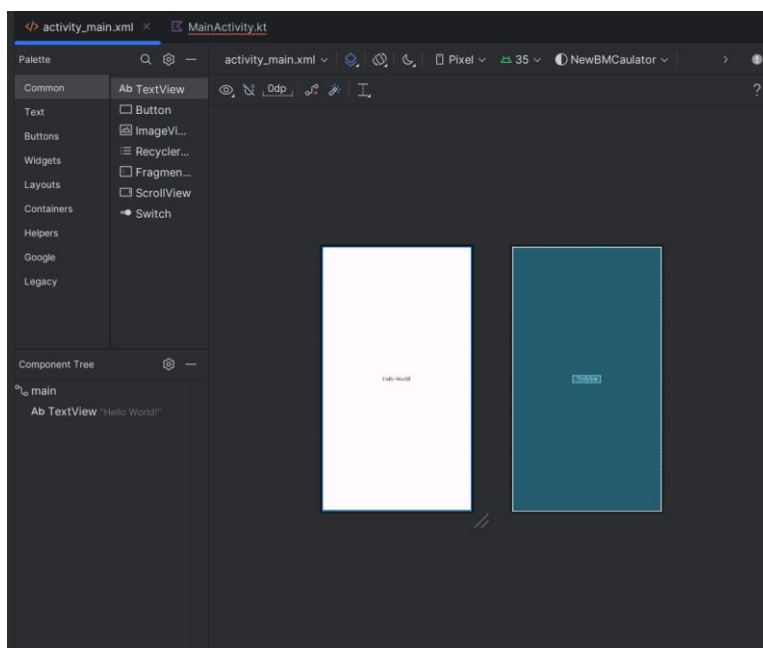
12. Đảm bảo rằng tùy chọn Tương thích ngược (Tương thích ứng dụng) được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước đó.

13. Nhấp vào Kết thúc.

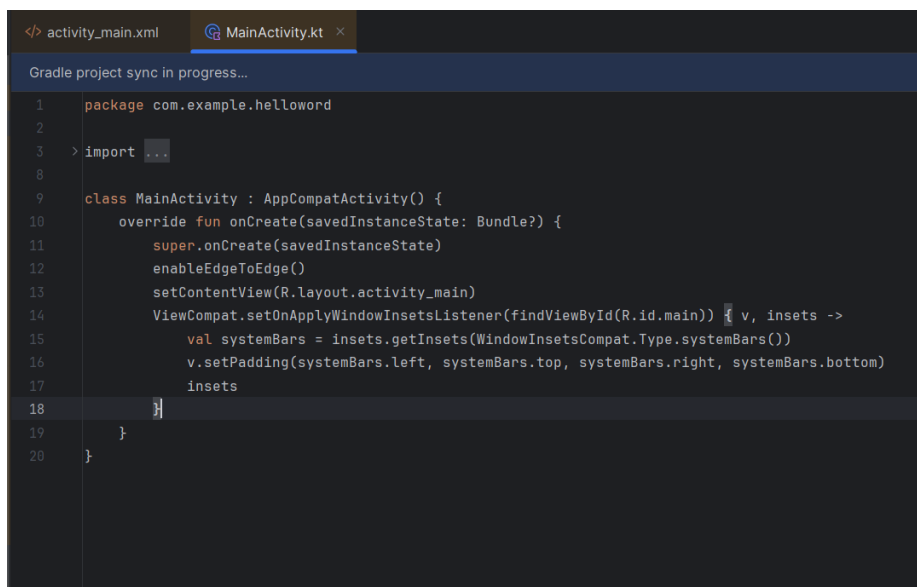
Trình chỉnh sửa Android Studio xuất hiện. Thực hiện theo các bước sau:

1. Nhấp vào tab `ctivity_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 được hiển thị bên dưới.



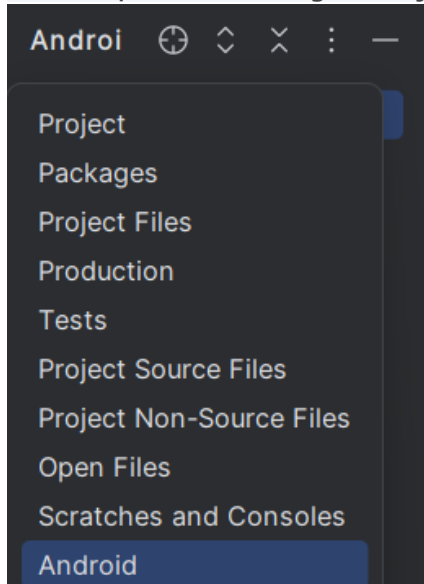
3. Nhấp vào tab `MainActivity.java` để xem trình soạn thảo mã như hiển thị bên dưới.



2.2 Khám phá ngăn Project > Android

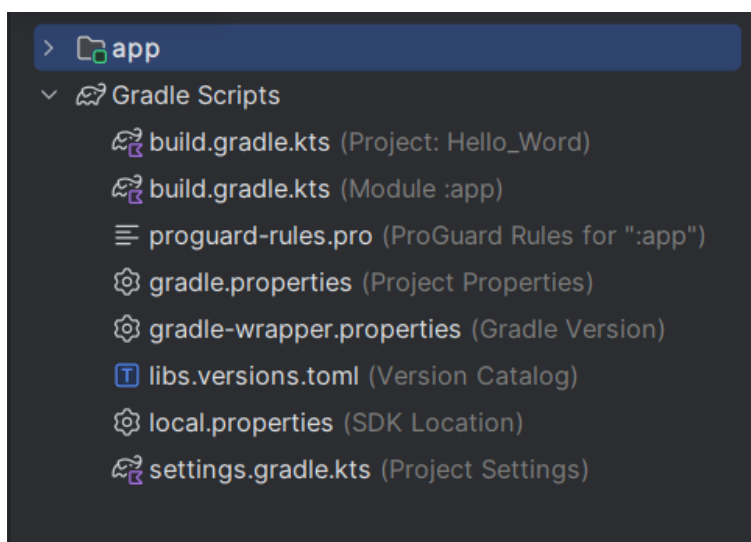
Trong bài thực hành này, bạn sẽ khám phá cách tổ chức dự án trong Android Studio.

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 Project, như hiển thị bên dưới.



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 của mình dưới dạng phụ thuộc.

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

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 xây 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 xây 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 dependency của bạn là thứ gì đó khác ngoài 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:

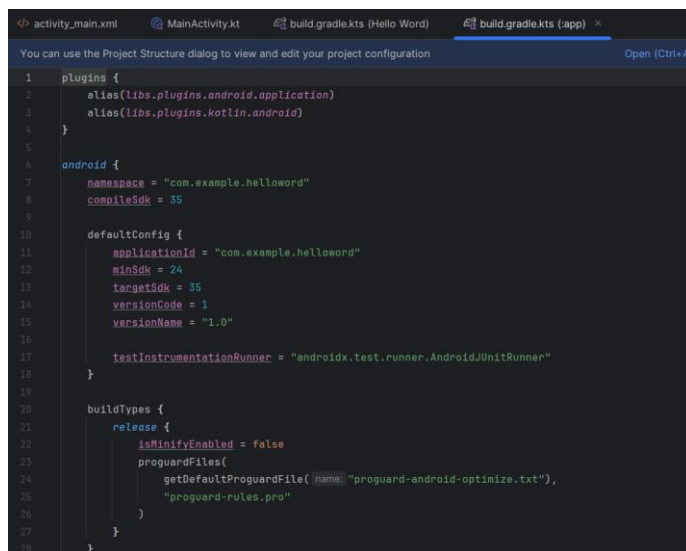
```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2 plugins {
3     alias(libs.plugins.android.application) apply false
4     alias(libs.plugins.kotlin.android) apply false
5 }
```

3. Tìm tệp build.gradle (Module: ứng dụng).

Ngoài tệp build.gradle cấp dự án, mỗi mô-đun đều có tệp build.gradle riêng, cho phép bạn định cấu hình cài đặt dựng cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Định cấu hình các cài đặt dựng này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như các loại dựng bổ sung và hương vị sản phẩm. Bạn cũng có thể ghi đè 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 phụ thuộc trong phần phụ thuộc. 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 implementation fileTree(dir: 'libs', include: ['*.jar']) thêm một 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:



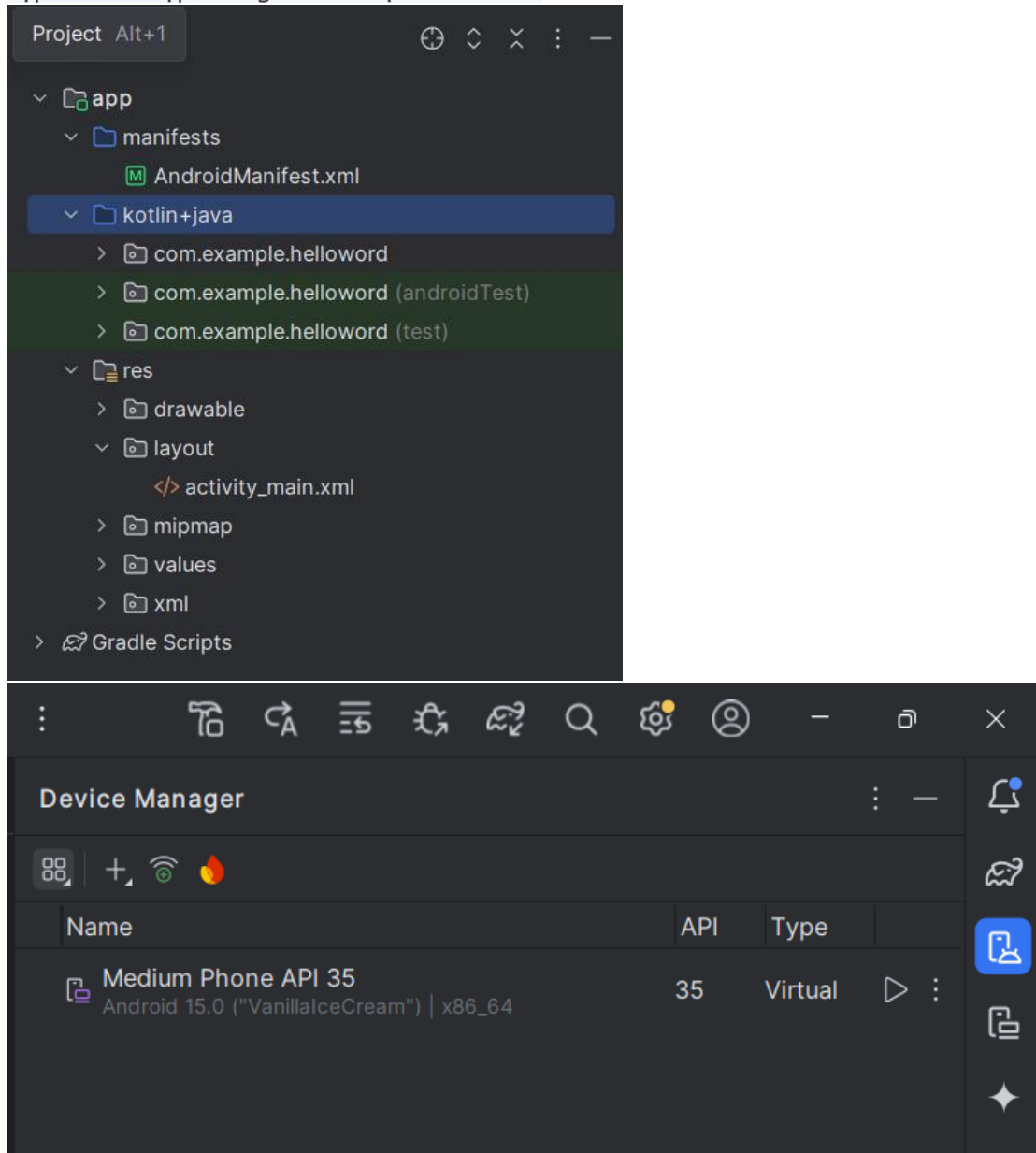
```
1 plugins {
2     alias(libs.plugins.android.application)
3     alias(libs.plugins.kotlin.android)
4 }
5
6 android {
7     namespace = "com.example.helloworld"
8     compileSdk = 35
9
10    defaultConfig {
11        applicationId = "com.example.helloworld"
12        minSdk = 24
13        targetSdk = 35
14        versionCode = 1
15        versionName = "1.0"
16
17        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
18    }
19
20    buildTypes {
21        release {
22            isMinifyEnabled = false
23            proguardFiles(
24                getDefaultProguardFile("proguard-android-optimize.txt"),
25                "proguard-rules.pro"
26            )
27        }
28    }
29 }
```

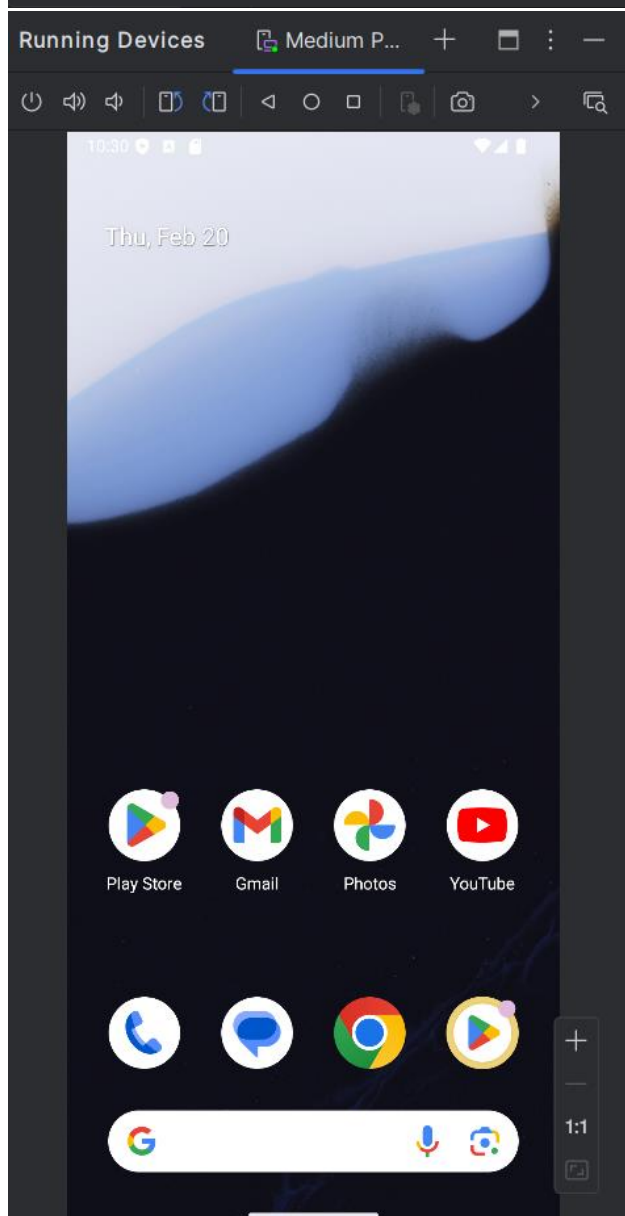
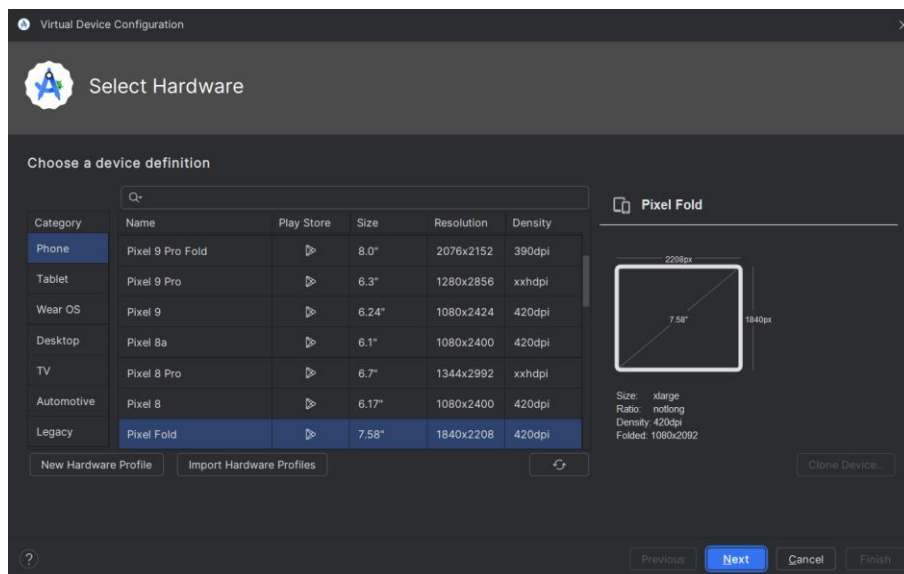
4. Nhấp vào hình tam giác để đóng Gradle Scripts.

2.4 Khám phá các thư mục app và res

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

1. Mở rộng thư mục app, thư mục java và thư mục `com.example.android.helloworld` để xem tệp java MainActivity. Nhấp đúp vào tệp để mở tệp trong trình soạn thảo mã.





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.
- Một 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 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 phù hợp cho thiết bị của mình. Đối với trình điều khiển USB dựa trên Windows, hãy xem OEM USB Drivers.

4.1 Bật gỡ lỗi USB


Để cho phép 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 **Developer options** của thiết bị của bạn.

Trên Android 4.2 trở lên, màn hình **Developer options** bị ẩn theo mặc định. Để hiển thị **developer options** và bật USB Debugging:

1. Trên thiết bị của bạn, hãy mở **Settings**, tìm kiếm **About phone**, nhấp vào **About phone** và chạm vào **Build number** bảy lần.
2. Quay lại màn hình trước đó (**Settings / System**). **Developer options** xuất hiện trong danh sách. Nhấn vào **Developer options**.
3. Chọn **USB Debugging**.

4.2 Chạy ứng dụng của bạn trên thiết bị

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. Nhấp vào nút Run  trên thanh công cụ. Cửa sổ **Select Deployment Target** sẽ mở ra với danh sách các trình giả lập khả dụng và các thiết bị được kết nối.
3. Chọn thiết bị của bạn, và nhấp **OK**.

Android Studio sẽ 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, có thể thử cách sau:

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

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

1. Rút phích cắm thiết bị
2. Trên thiết bị, mở **Developer Options in Settings app**.
3. Nhấn vào thu hồi quyền **USB Debugging**.
4. Kết nối lại thiết bị với máy tính của bạn.
5. Khi được nhắc, hãy cấp quyền.

Bạn có thể cần cài đặt trình điều khiển USB phù hợp cho thiết bị của mình. Tham khảo Using Hardware Devices documentation.

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

Làm theo các bước sau:

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

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

2. Trong khối defaultConfig, hãy thay đổi giá trị của minSdk thành 26 như hiển thị bên dưới (ban đầu giá trị này được đặt thành 24).

```
android {  
    namespace = "com.example.helloworld"  
    compileSdk = 35  
  
    defaultConfig {  
        applicationId = "com.example.helloworld"  
        minSdk = 26  
        targetSdk = 35  
        versionCode = 1  
        versionName = "1.0"  
    }  
}
```

5.2 Đồng bộ cấu hình Gradle mới

Khi bạn thực hiện thay đổi đối với các tệp cấu hình xây 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 **Sync Now** trên thanh thông báo xuất hiện khi thực hiện thay đổi (như thể hiện trong hình trước) hoặc nhấn vào biểu tượng **Sync Project with**

Gradle Files  trong 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.

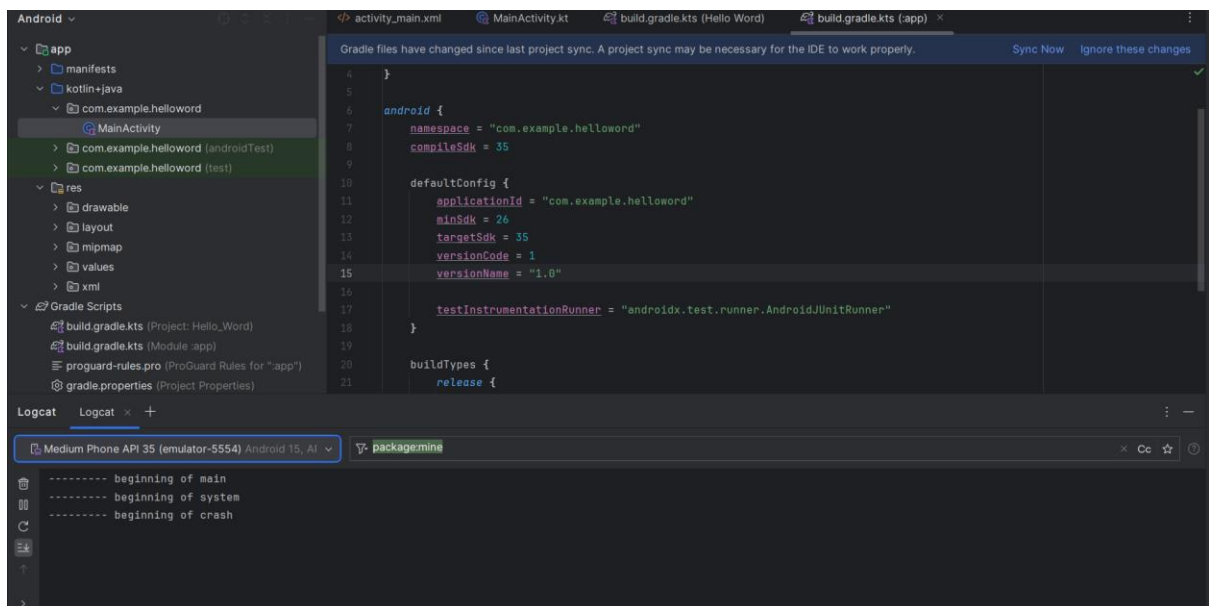
Để hiểu sâu hơn về Gradle, hãy tham khảo tài liệu Build System Overview và Configuring Gradle Builds.

Nhiệm vụ 6: Thêm log statement 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 ngoại lệ.

6.1 Xem ngăn Logcat

Để xem ngăn **Logcat**, hãy nhấp vào biểu tượng **Logcat**  ở thanh công cụ bên trái màn hình Android Studio như minh họa 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 Log vào ứng dụng, thông báo Log sẽ xuất hiện ở đây.
2. Menu cấp độ của Log được để mặc định, hiển thị tất cả các thông báo Log. Các thiết lập 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 log đến ngăn Logcat.
- d: Cài đặt mức **Debug** Log để lọc hiển thị thông báo log trong ngăn Logcat. Các mức log 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 Activity mà tin nhắn bắt nguồn. Tuy nhiên, bạn có thể biến nó thành bất kỳ 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.

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 lệnh 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 **Android Studio > Preferences** trong macOS.
3. Chọn **Editor > General > Auto Import**. Chọn tất cả các hộp kiểm và thiết lập **Insert imports on paste to All**.
4. Chọn **Apply** và sau đó nhấn **OK**.
5. Trong phương thức onCreate() của MainActivity, thêm câu lệnh sau:

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

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

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

6. Nếu ngăn Logcat chưa mở, hãy nhấp vào biểu tượng Logcat ở thanh công cụ bên trái Android Studio để mở.
7. Kiểm tra xem tên mục tiêu và tên gói của ứng dụng có đúng không.
8. Thay đổi mức Log trong ngăn **Logcat** thành **Debug** (hoặc giữ nguyên vì có rất ít thông báo log).
9. Chạy ứng dụng của bạn.

Bài 1.2 Phần A: Giao diện người dùng tương tác đầu tiên của bạn

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 — mọi thành phần của màn hình là một view. Lớp view biểu diễn 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 được mô tả trong một số bài học bao gồm:

- TextView để hiển thị văn bản.
- EditText để 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.
- ImageView để hiển thị hình ảnh.
- ConstraintLayout và LinearLayout để chứa các thành phần View khác và định vị chúng.