アンドロイドアプリの更新方法

著者：関 勝寿　公開日：2024 年 7 月 22 日　ソース：https://sekika.github.io/2024/07/22/Android-update/

Flutter でナンプレアプリを作って公開している。モバイルアプリは、定期的に新しい開発環境でアプリを作成し直す必要がある。iOS 版では Xcode と Flutter のバージョンアップをすれば良いが、Android 版では様々なバージョンアップが必要なため、この記事でまとめる。

1.Android Studio のアップデート

Android Studio を開き、メニューの Android Studio から Check for updates とする。Android Studio そのもののアップデートをした場合には、さらにもう一度 Check for updates として、コンポーネントをアップデートする。Check for updates としたときに "You already have the latest version of Android Studio and plugins installed." と出ればアップデート完了となる。

2.SDK と NDK のアップデート

Android API Levels を参考に、SDK の最新バージョンを使う。Android Studio の Tools メニューから SDK Manager を起動し、Languages & Farameworks > Android SDK を選ぶ。そして、

- SDK Platforms のタブから、最新版の Android SDK Platform package をチェックする。
- SDK Tools のタブから、Show package details でパッケージを表示して、最新版の Android SDK Build-Tools と NDK (Side by side) をチェックする。NDK のバージョン番号 (27.0.01207793 のような番号) は「build.gradle の設定」の項で使う。
- OK ボタンを押すことで、最新版の SDK と NDK がインストールされる。

3.Java のアップデート

私の場合は Mac で Homebrew を使っている。openjdk を最新バージョンにする。

java -version でバージョンを確認する。最新バージョンになっていない場合、Homebrew openjdk によると、macOS >= 10.15 では以下のコマンドを実行する必要がある。

```
sudo ln -sfn $(brew --prefix)/opt/openjdk/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk.jdk
```

4.build.gradle の設定

SDK と NDK と Java のバージョンが定まったら、android/app/build.gradle を設定する。以下は、このファイルの中で関係のあるところのみを示す。

```
def localProperties = new Properties()
def localPropertiesFile = rootProject.file('local.properties')
if (localPropertiesFile.exists()) {
  localPropertiesFile.withReader('UTF-8') { reader ->
    localProperties.load(reader)
  }
}

def                flutterVersionCode                =
localProperties.getProperty('flutter.versionCode')
def                flutterVersionName                =
localProperties.getProperty('flutter.versionName')
def ndkVersionName = localProperties.getProperty('ndk.version')

def javaVersion = localProperties.getProperty('javaVersion')

android {
  namespace "io.github.sekika.sudoku"
  compileSdk flutter.targetSdkVersion
  ndkVersion ndkVersionName

  compileOptions {
    sourceCompatibility JavaVersion.toVersion(javaVersion)
    targetCompatibility JavaVersion.toVersion(javaVersion)
  }

  kotlinOptions {
    jvmTarget = javaVersion.toString()
  }

  java {
    toolchain {
      languageVersion.set(JavaLanguageVersion.of(javaVersion))
    }
  }

  defaultConfig {
    applicationId "io.github.sekika.sudoku"
    minSdkVersion flutter.minSdkVersion
    targetSdkVersion flutter.targetSdkVersion
    versionCode flutterVersionCode.toInteger()
    versionName flutterVersionName
  }
}
```

これは、必要な設定を android/local.properties から読み込んでいる。local.properties には、たとえば次のように記述する。

```
sdk.dir=/Users/seki/Library/Android/sdk
flutter.sdk=/opt/homebrew/Caskroom/flutter/3.22.2/flutter
flutter.buildMode=release
flutter.versionName=1.0.5
flutter.versionCode=14
flutter.minSdkVersion=24
flutter.targetSdkVersion=37
ndk.version=30.0.15729638
javaVersion=17
```

- flutter.versionCode の行までは Flutter が自動的に設定する（ただしアプリのバージョンは pubspec.yaml を更新する）。
- flutter.minSdkVersion は対応する最小 SDK バージョンで、flutter.targetSdkVersion は最新の SDK バージョンである（compile と target で別の値にすることもできるが、ここでは同じ値を使うようにしている）。
- flutter.ndkVersion は NDK のバージョンである。flutter.targetSdkVersion と ndk.Version は「SDK と NDK のアップデート」でインストールしたバージョンである。Mac の場合は、ls ~/Library/Android/sdk/ndk/ とするとインストールされている NDK のバージョン一覧が出るので、それをコピーするのが確実である。
- javaVersion はアプリの形式としての Java のバージョンであり、古い Android スマートフォンで実行されることや Android へ変換するツール（D8/R8）が対応しているかを考えると、必ずしも高ければ良いというものでもない。コンパイラの Java のバージョンは javaVersion と同じかより高くなければならない。

5.Gradle のアップデート

- Gradle のバージョンについては、Gradle のリリースと Java との互換性を確認して、適切なバージョンを選ぶ。

・Gradle が入っていない場合はインストールする。Homebrew の場合は brew install gradle
・Gradle wrapper が入っていない場合（android/gradlew がない場合）には、android ディレクトリで gradle wrapper とすることで入れる。
・android/gradle/wrapper/gradle-wrapper.properties に、以下のように記述する。ここで 9.5.0 は Gradle のバージョンである。

```
distributionUrl=https\://services.gradle.org/distributions/gradle-9.5.0-all.zip
```

6. プラグインのアップデート

android/settings.gradle には、以下のようなプラグインの設定がある。

```
plugins {
  id "dev.flutter.flutter-plugin-loader" version "1.0.0"
  id "com.android.application" version "9.3.1" apply false
}
```

ここで、"com.android.application" は AGP (Android Gradle Plugin) である。
・AGP のリリース / リリースノート
The 'org.jetbrains.kotlin.android' plugin is no longer required for Kotlin support since AGP 9.0.

7.Flutter のアップデート

・flutter upgrade で　Flutter をアップデートする。
・flutter pub upgrade で Flutter のパッケージをアップデートする。

8. パッケージの作成

android ディレクトリで ./gradlew clean build としてアプリをビルドする。ここで、ビルド時に NewerVersionAvailable というエラーが出ることがある。たとえば、/Users/seki/.pub-cache/hosted/pub.dev/url_launcher_android-6.3.5/android/build.gradle:71: Error: A newer version of org.mockito:mockito-core than 5.1.1 is available: 5.12.0 [NewerVersionAvailable] というようなエラーである。この場合には、エラーで表示されたファイルを直接編集する。たとえば、この場合には /Users/seki/.pub-cache/hosted/pub.dev/url_launcher_android-6.3.5/android/build.gradle というファイルの 71 行目を

```
  testImplementation 'org.mockito:mockito-core:5.12.0'
```

と書き換えることで、エラーを回避することができる。
flutter build appbundle --release でリリースパッケージ build/app/outputs/bundle/release/app-release.aab を作成する。