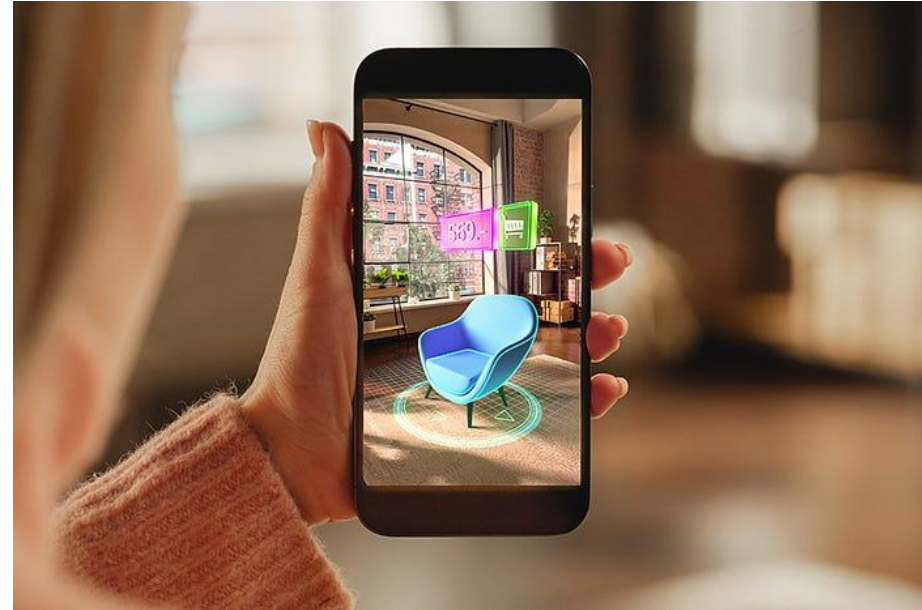


ENHANCING YOUR FLUTTER APPLICATION WITH AR/VR ELEMENTS

AUGMENTED REALITY (AR)

- Technology that combines the real world with virtual elements (computer-generated virtual objects)
- The perception of the real world is enhanced and consolidated by the addition of virtual elements



AUGMENTED REALITY(AR) VS VIRTUAL REALITY(VR)

- **AR:** Users view virtual objects superimposed on them while still seeing the physical world.
- **VR:** It prevents seeing the real world and puts the user in a completely virtual world.



AR VS VR DEVICES

- AR Devices
 - smartphones, tablets or special AR glasses.
- VR Devices
 - special VR glasses or headset, computer or game console.



<https://medium.com/@kadriyemacit/develop-ar-apps-with-flutter-44ce1f0b9b85>

AUGMENTED REALITY

USAGE AREAS

education, entertainment, business, health, tourism,
industrial design, ...



EXAMPLE: POLAROIDLAB - BRING YOUR POLAROIDS TO LIFE



https://www.youtube.com/watch?v=dm5xn_fidrk

AR LIBRARIES

- Android: ARCore
 - Developed by Google
 - Available on Android, iOS devices (augmenting ARKit), Web, Adobe Aero & Unity
 - Android Emulator
 - **API Level 27+** (Oreo 8.1)
 - Android devices
 - AR Optional: **API Level 19+** (KitKat 4.4.4)
 - AR Required: **API Level 24+** (Nougat 7.0)
 -
- iOS: ARKit
 - Developed by Apple
 - Latest version is **6**
 - Available on iOS **11.0+**, iPadOS **11.0+**, Mac Catalyst **14.0+**, visionOS **1.0+**

AR FLUTTER PLUGINS (1/3)

- Android: arcore flutter plugin
 - Not officially maintained by Google (developed by Gian Marco di Francesco)
 - Latest version is **0.1.0**, prerelease is **0.2.0-alpha**
 - Last updated on **Feb 7, 2023**
- iOs: arkit plugin
 - Not officially maintained by Apple (developed by Oleksandr Leuschenko)
 - Latest version is **1.2.1**
 - Last updated on **Sep 7, 2025**

AR FLUTTER PLUGINS (3/3)

- ar_flutter_plugin engine
 - Developed by David Heredia, fork of the original *ar_flutter_plugin*
 - Latest version is **1.0.1**
 - Last updated on **July 4, 2024**
 - Not actively maintained, like most *ar_flutter_plugin* forks
 - It will be used in this laboratory exercise

PREREQUISITES: 1. ENABLE AR IN YOUR ANDROID APP

- Official Instructions

- a. Add the following snippet into **android/app/src/main/AndroidManifest.xml**

```
<manifest ...>
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera.ar" />
    <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

- b. Find your Java version by typing **flutter doctor -v**. Then consult the Gradle Compatibility Matrix to identify the corresponding gradle version (3rd column). Finally set the *distributionUrl* parameter of **android/gradle-wrapper.properties** accordingly

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.5-all.zip
```

PREREQUISITES: 2. ENABLE AR IN YOUR ANDROID APP (CONT...)

- Official Instructions

- a. Set the version of plugin id `“com.android.application”` to `“8.3.2”` in **app/settings.gradle**
- b. Modify **android/app/build.gradle** to
 - tell the gradle build tool that our application should be compiled with SDK version 36
 - support the latest LTS version of Android Native Developer Kit,
 - to run on SDKs from version 28
 - add dependency on the ARCore library.

PREREQUISITES: 2. ENABLE AR IN YOUR ANDROID APP (CONT...)

- **android/app/build.gradle**

```
android {  
    compileSdk = 36  
    ndkVersion = "27.3.13750724"  
    defaultConfig {  
        ...  
        minSdkVersion 28  
    }  
    buildTypes {  
        release {  
            ...  
            shrinkResources false  
            minifyEnabled false  
        }  
    }  
}  
  
dependencies {  
    ...  
    implementation 'com.google.ar:core:1.33.0'  
}
```

PREREQUISITES: 3. ENABLE AR IN YOUR ANDROID EMULATOR

- Official Instructions

- a. Software requirements: *Android Studio 3.1+*, *Android Emulator 27.2.9+*
- b. Install a *Google APIs Intel x86 Atom System Image API Level 27+*
 - Preferably use API level 33 (Android Tiramisu)
- c. Install Google Play Services for AR from [GitHub](#)

PREREQUISITES: 4. ENABLE THE PLUGIN

- Add the following snippet under dependencies in **pubspec.yaml**

```
ar_flutter_plugin_engine: ^1.0.0
```

```
vector_math: ^2.1.4
```

RUNNING AN EXAMPLE

- The plugin code examples will likely not work. We are going to adapt the code example of [this article](#)
 - Issue **flutter build apk** and copy & install *build/app/outputs/flutter-apk/app-release.apk* to your device
- You may download more objects from this [Khronos Group](#) repository (now archived)
- You may download even more object from [SketchFab](#)
- You may create your own objects using [Unity](#) of [Vuforia](#)!

AR LOCATION VIEW: A MORE ADVANCED EXAMPLE

- Augmented reality for geolocation
- Designed to be used in areas with large concentration of static POIs
- Could be used in the context of a tourism or recommendation-related app
- Plugin repository on pub.dev
- Working example directory on [github](https://github.com)



AR LOCATION VIEW: MINOR MODIFICATIONS

- You have to make the same modifications to your project files as in the previous example, plus the following

- a. Add dependencies on **pubspec.yaml**

```
ar_location_view: ^2.0.16
geolocator: ^13.0.1
uuid: ^4.4.2
```

- b. Add the following snippet into
android/app/src/main/AndroidManifest.xml

```
<manifest ...>
  <uses-permission android:name="android.permission.CAMERA"/>
  <uses-permission android:name="android.permission.RECORD_AUDIO"/>
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
</manifest>
```

VIRTUAL REALITY

—

USAGE AREAS

healthcare, retail, tourism, real estate, architecture,
interior design, ...



EXAMPLE: BMW EMPLOYEE TRAINING ON SAFETY



<https://www.youtube.com/watch?v=5EAa3H8LP5>

VR LIBRARY

- vr_player
 - Lets you play 360° and VR videos smoothly, delivering an immersive viewing experience with touch and device motion controls.
 - Developed by flutter.wtf
 - Available on Android and iOs devices
 - Latest version is **0.3.0**
 - Last updated on **Mar 10, 2025**
 - We will go through the plugin's example code

VR_PLAYER CODE EXAMPLE

- You have to make the same modifications to your project files as in the previous example, plus the following
 - Add `vr_player: ^0.3.0` in **pubspec.yaml**
 - Add the following snippet in **android/app/src/main/AndroidManifest.xml**

```
<manifest ...>
    <uses-permission android:name="android.permission.INTERNET"/>
    ...
</manifest>
```

- Issue **flutter release apk** and copy & install `build/app/outputs/flutter-apk/app-release.apk` to your device
- Preview the video on your device as-is or use VR gear!