



"Excited to announce the new Android XR platform and its first device, Project Moonbeam! It's an incredible feeling to finally share the project I've been dedicated to for years. Stay tuned, as there's so much more exciting news (glasses and more) to come!"

— Kihwan Kim, EVP Samsung XR

"I feel that Augmented reality is perhaps the ultimate computer."

— Satya Nadella

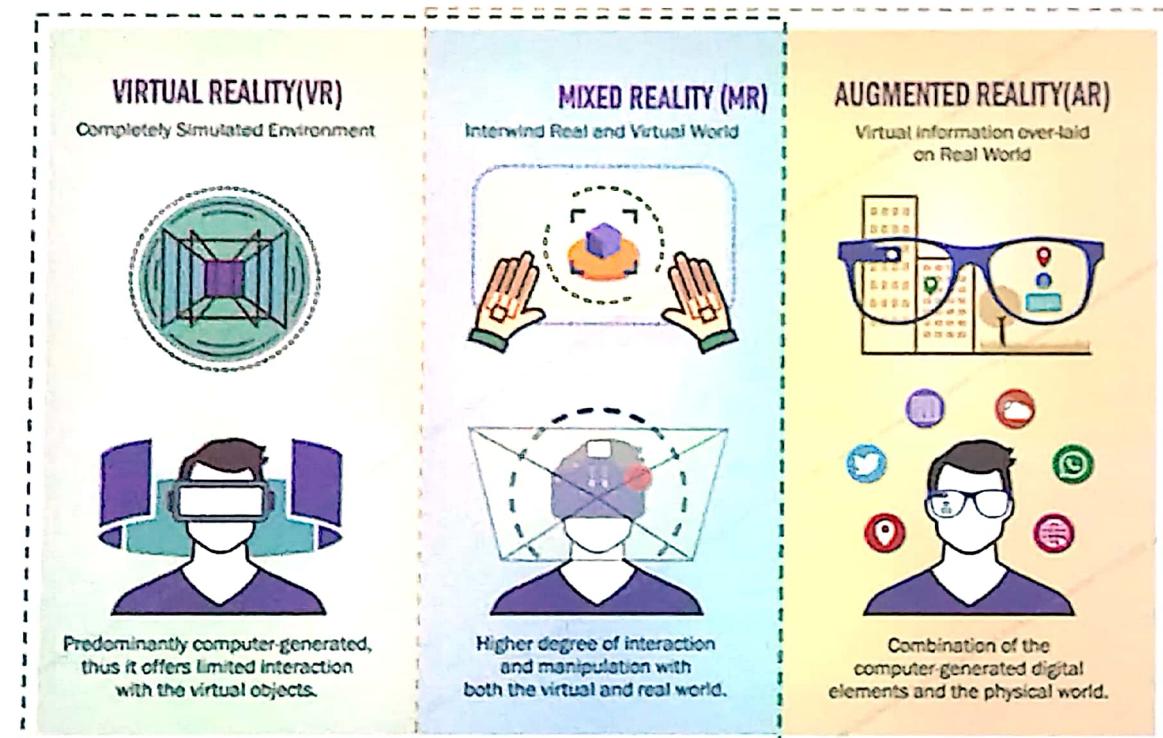
"The smartphone is for everyone. I think AR is that big, it's huge. I get excited because of the things that could be done that could improve a lot of lives."

— Tim Cook

"There will come a time when smartphones are less frequently used than they are now, potentially being replaced by smart glasses" (2030?)

— Mark Zuckerberg

XR | Evolution

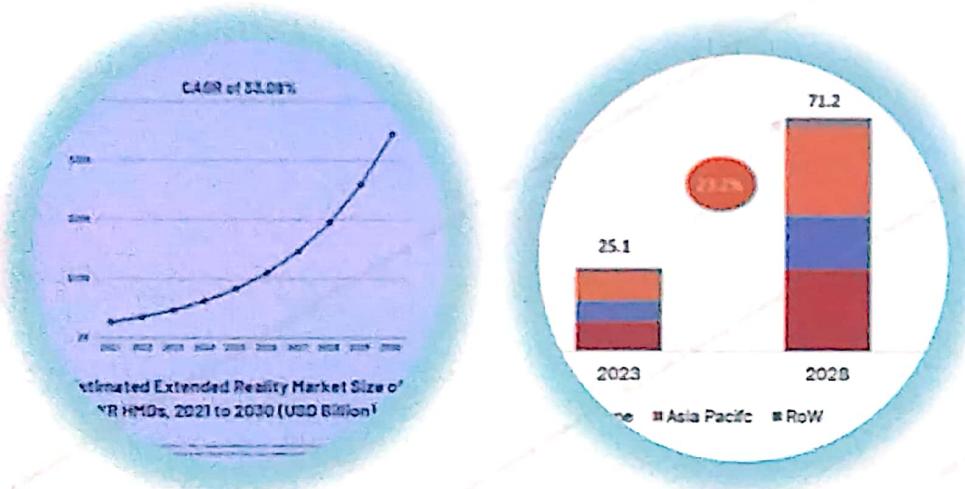


Extended Reality (XR)				
	Reality	Augmented Reality (AR)	Mixed Reality (MR)	Virtual Reality (VR)
Display	Naked eye/optical glasses	Translucent display	Translucent display	Occlusion display
Display example				
Example				

Source: https://www.researchgate.net/figure/Schematic-illustration-of-virtual-reality-augmented-reality-and-mixed-reality_fig1_349459034

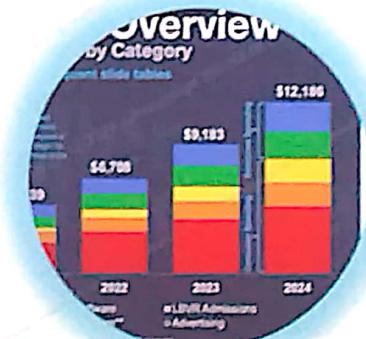
XR | Market Overview

XR Market growing at a consensus
~30-35% CAGR



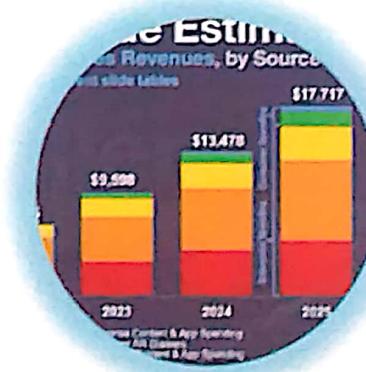
*US currently the largest market;
APAC expected to show highest growth*

2 Markets to remain – AR & VR; B2C
expected to be majority



VR

- *B2C driven*
- *B2B2C (LBVR) – significant portion*



AR

- *Currently B2B driven*
- *Transition to B2C over Time*

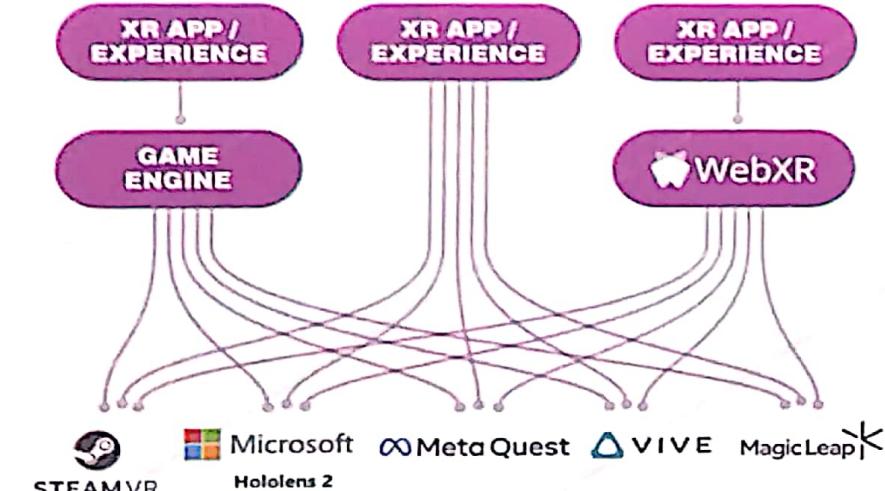
XR | OpenXR

OpenXR is a royalty-free, open standard offering a unified API for developing AR/VR applications across diverse devices.

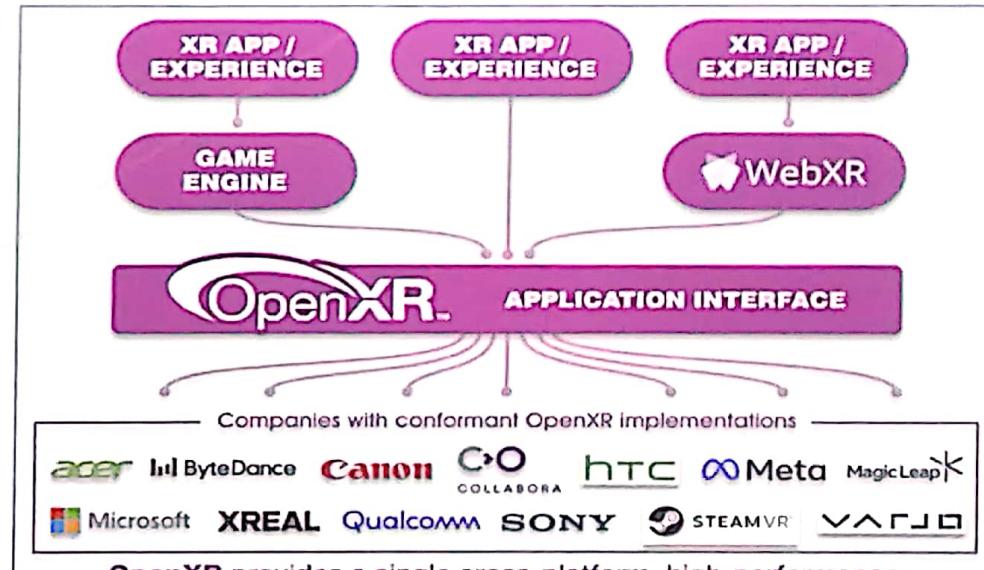


Maintained by Khronos Group
<https://www.khronos.org/OpenXR/>

- Cross-Platform Compatibility
- Interoperability
- Extensibility
- Performance Optimization
- Community-Driven Development



Before OpenXR: Applications and engines needed separate proprietary code for each device on the market.



OpenXR provides a single cross-platform, high-performance API between applications and all conformant devices.

Source: <https://www.khronos.org/OpenXR/>

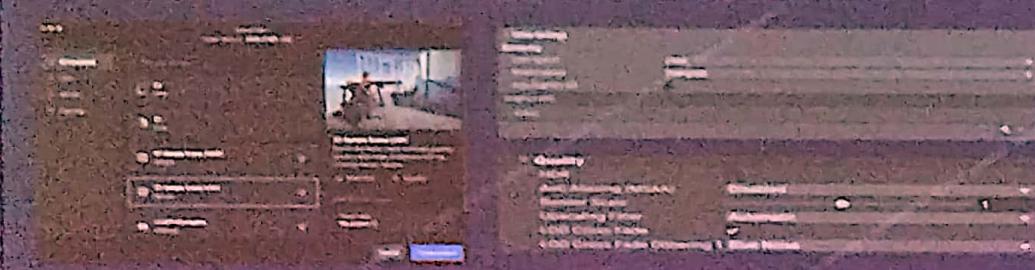


Scanned with OKEN Scanner

Android XR | Unity – Quick start

1. Set up Unity & new project

- Download and Install Unity hub: <https://unity.com/download>
- Install Unity Engine using Unity Hub with Android development tools
- Create new URP project (Unity Hub -> New Project -> URP Template -> Create)
- Project Setup - <https://developer.android.com/develop/xr/unity/setup>



Source: <https://unity.com/download>

Source: <https://developer.android.com/develop/xr/unity/setup>

2. Develop for Android XR

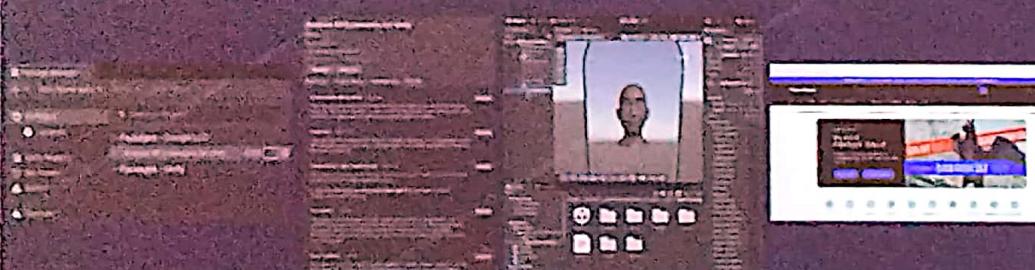
- Import Android XR unity package (Window -> Package Manager -> Git URL)
 - URL - <https://github.com/android/android-xr-unity-package.git>
- Enable OpenXR & Android XR feature group (Edit -> Project Settings -> XR Plug-in Management)
- Enable feature (Edit -> Project Settings -> Xr Plug-in Management -> OpenXR)



Source: <https://developer.android.com/develop/xr/unity/xr-extensions-quickstart>

3. Asset/Custom & Sample scenes handling

- Import Sample Scene (e.g. Face Tracking) from android XR package
 - Package Manager -> In project -> Android XR Extensions for Unity
 - Samples -> Face Tracking -> Import
- Customize sample scene as per requirement or create new scene
- Asset market place: Unity Asset Store (<https://assetstore.unity.com/>)

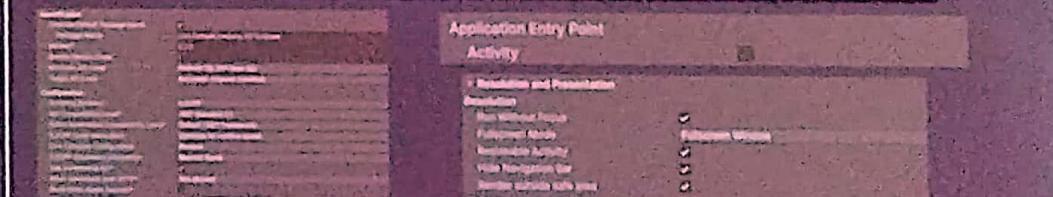


Source: <https://developer.android.com/develop/xr/unity/xr-extensions-quickstart>

Source: <https://assetstore.unity.com/>

4. Build & Deploy

- Build settings (Edit -> Project Settings -> Player)
(<https://docs.unity3d.com/2022.3/Documentation/Manual/android-BuildProcess.html>)
 - Minimum API level: 24
 - Application entry point: Game Activity
 - Pop-up windows (Android -> Resolution & Presentation -> Resolution -> Resizable Activity)
- Deploy: <https://developer.android.com/develop/xr/package-and-distribute>



Android XR | Jet Pack XR SDK - Overview

- Jetpack XR SDK lets you build immersive XR experience using Kotlin, Compose, Java & android views.
- Spatialize your UI, Load & render 3D model, semantically understand the world.

Github Sample Project: <https://github.com/android/xr-samples>

Documentation: <https://developer.android.com/develop/xr/jetpack-xr-sdk>

Develop & Debug

Android Studio



- Android XR Emulator
- XR Support in Layout Inspector
- XR Project template



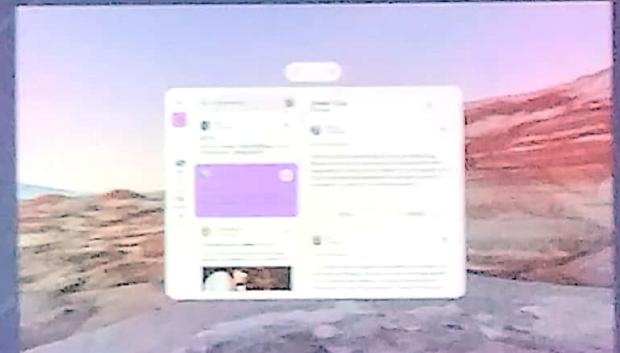
Jetpack Compose



Create Spatial Layouts in XR

Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/compose>

Material design for XR



Provides spatial-ready design components

Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/material-design>

Jetpack SceneCore



Place and arrange 3D content, defined by entities

Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/entity-based-3d-content>

AR Core for Jetpack



Blend digital world to real world

Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/ar-core>

Developer Preview 3: <https://android-developers.googleblog.com/2025/12/build-for-ai-glasses-with-android-xr.html>



Scanned with OKEN Scanner

Android XR | Jet Pack – Quick start

1. Set up Android Studio & Jetpack

- Download & Install Android Studio(Canary Build)
<https://developer.android.com/studio/preview>
- SDK Tools: Build-tools, emulator, Platform-Tools, Layout Inspector for API 31-36
- SDK Platforms: Google Play XR ARM system image (macOS) Google Play XR Intel x86_64 (Windows)



Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/setup>

2. Develop for Android XR

- Create new project with XR template. (XR -> Basic Headset Activity)
- Create Emulator for XR project.
 - New Android Virtual Device (Tools -> AVD Manager)
 - Add device (XR -> XR Headset)
 - Configure Virtual Device -> Finish



Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/setup>

3. Asset/Custom & Sample scenes handling

- Supports GLTF model loading with Jetpack SDK. (GLTF, GLB)
- Add XR Features to existing App (e.g. dialog to SpatialDialog)
- Add spatial capability to your app: Add Environment, 3D Model, spatial video & audio; transition from Home space (2D panel) to Full space (Immersive view)
- Develop GUI with Jet Pack Compose



Source: <https://developer.android.com/develop/xr/jetpack-xr-sdk/>

4. Build & Deploy

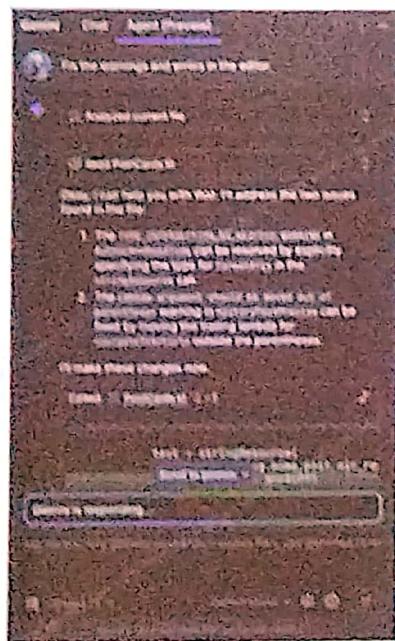
- Min SDK: 24
- Must compile to SDK 34 or higher
- Release on play store - <https://developer.android.com/develop/xr/package-and-distribute>
- "Made for XR" label with headset icon on play store.



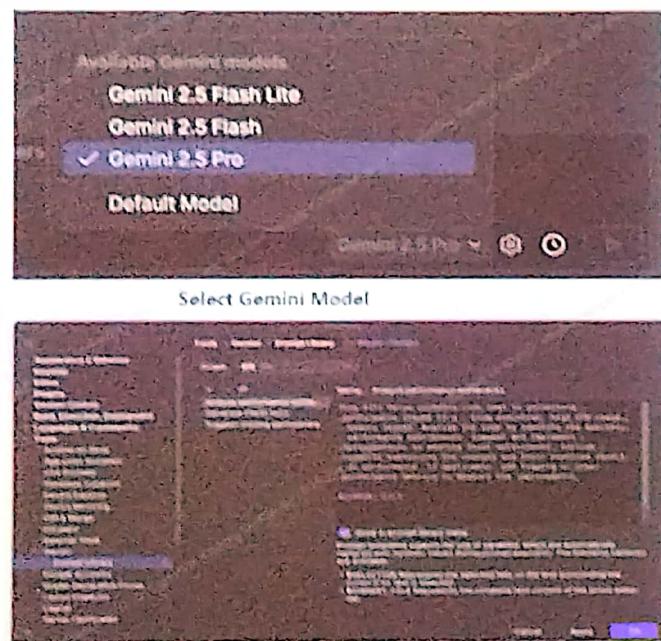
Scanned with OKEN Scanner

Gemini | Agentic AI

- Gemini Agent mode is supported in Android Studio
- Goal Understanding: Interprets user intent and breaks it into actionable items
- Autonomous task execution: Carries out task without constant user prompting
- Tools & App Integration: Uses Gmail, Drive, Calendar, Docs & APIs
- Iterative Reasoning: Monitor progress, self-corrects & improve results
- Multi-Modal Intelligence: Works with text, images, audio, video & code
- Provide a high-level goal, and the agent creates and executes a plan, invoking the necessary tools, making changes across multiple files, and iteratively fixing bugs.



Gemini in Android Studio: Agent mode



Rules in Prompt Library

A screenshot of the Gemini interface in Android Studio showing the 'Get Started – Android Studio' tab. It provides step-by-step instructions: 'Click Gemini in the tool window bar. Sign in and onboard if you need to.', 'Select the Agent tab.', and 'Describe the task you want the agent to perform.' Below these instructions are three cards: 'Describe task for Agent', 'Review & approve change', and 'Auto approve changes (Optional)'. Further down, there are sections for 'Use cases', 'Fix build errors', 'Update UI Elements', and 'Generate Mock Data', each with corresponding icons and descriptions.

Source: <https://developer.android.com/studio/gemini/agent-mode>

Source: <https://android-developers.googleblog.com/2025/07/android-studio-narwhal-feature-drop-stable-agent-mode.html>