



Android

What we know so fAR **XR**



Dec 12, 2024 - First Signs

<https://blog.google/products/android/android-xr/>



Android XR





22 Jan, 2025 - Samsung Galaxy Unpacked

https://youtu.be/HinL5jCy_ol?t=1603



Android XR



26:49 / 1:09:54 • Integrated AI Platform >





28 Jan, 2025 - MKBHD

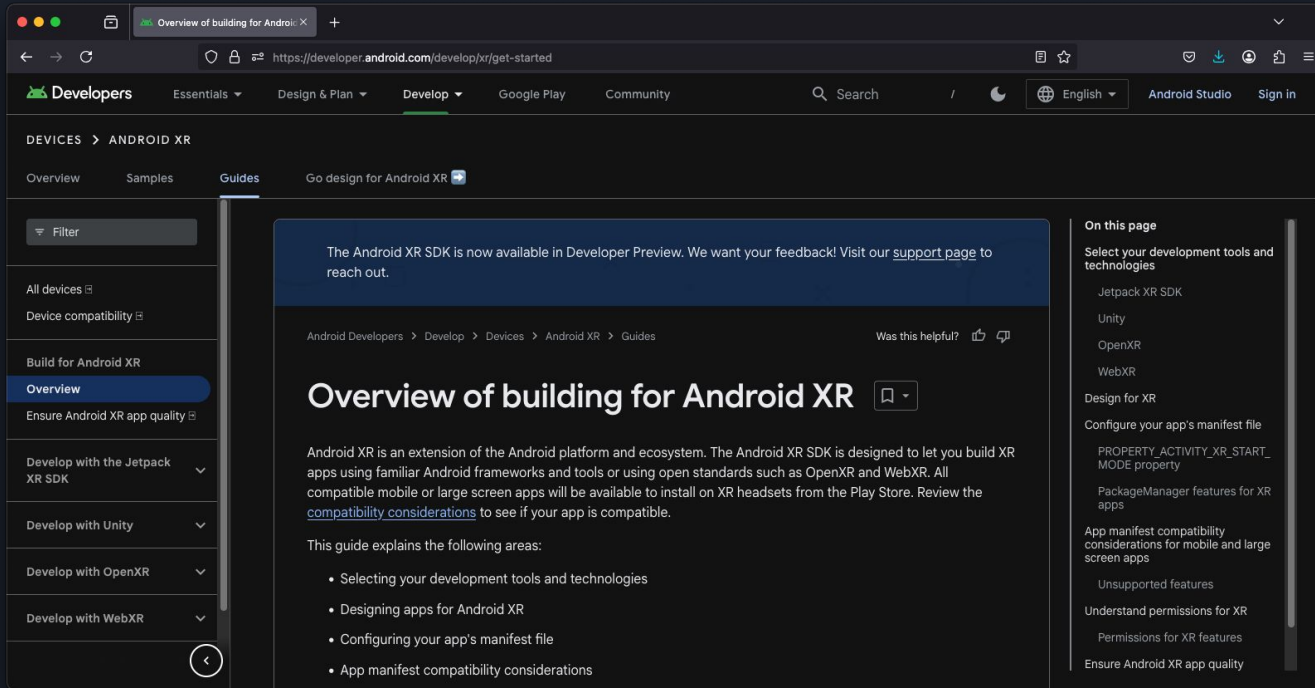
https://www.youtube.com/watch?v=az5QL_NLBvg





now: Developer Preview

<https://developer.android.com/develop/xr/get-started>



The screenshot shows a web browser displaying the Android Developer website. The page title is "Overview of building for Android XR". The navigation bar includes "Developers", "Essentials", "Design & Plan", "Develop", "Google Play", and "Community". The main content area features a blue header with the text: "The Android XR SDK is now available in Developer Preview. We want your feedback! Visit our [support page](#) to reach out." Below this is a breadcrumb trail: "Android Developers > Develop > Devices > Android XR > Guides". The main heading is "Overview of building for Android XR". The text states: "Android XR is an extension of the Android platform and ecosystem. The Android XR SDK is designed to let you build XR apps using familiar Android frameworks and tools or using open standards such as OpenXR and WebXR. All compatible mobile or large screen apps will be available to install on XR headsets from the Play Store. Review the [compatibility considerations](#) to see if your app is compatible." Below this, it says "This guide explains the following areas:" followed by a list of topics: "Selecting your development tools and technologies", "Designing apps for Android XR", "Configuring your app's manifest file", and "App manifest compatibility considerations". On the right side, there is a sidebar titled "On this page" with a list of links: "Select your development tools and technologies", "Jetpack XR SDK", "Unity", "OpenXR", "WebXR", "Design for XR", "Configure your app's manifest file", "PROPERTY_ACTIVITY_XR_START_MODE property", "PackageManager features for XR apps", "App manifest compatibility considerations for mobile and large screen apps", "Unsupported features", "Understand permissions for XR", "Permissions for XR features", and "Ensure Android XR app quality".

Android XR developer bootcamp

<https://forms.gle/Xh7MGBbFa7K5eWBr6>

Get hands-on help at an Android XR developer bootcamp

In early 2025, we are hosting **Android XR developer bootcamps** around the world. At these two-day events, you can:

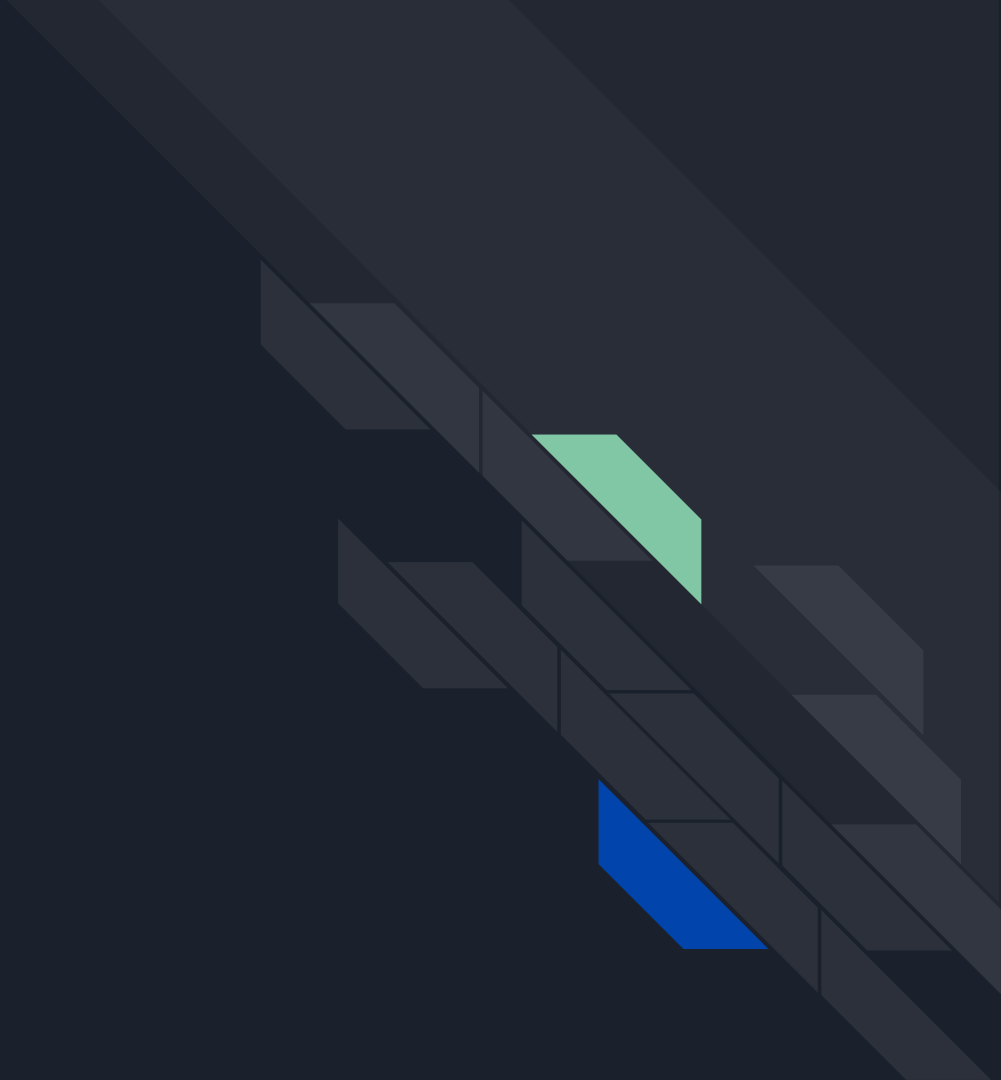
- Participate in guided technical sessions.
- Work with the Android XR team as you build a new app or adapt an existing one.
- Get access to prerelease headset hardware.

Space is limited. Register today to be considered for participation.

[Apply to the developer bootcamp](#)




or:
Emulation



Canary needed

About Android Studio



Android Studio Meerkat Feature Drop | 2024.3.2 Canary 6

Build #AI-243.24978.46.2432.13093109, built on February 20, 2025



Runtime version: 21.0.5+-13047016-b750.29 aarch64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

Kotlin plugin: K2 mode

Powered by [open-source software](#)
Copyright © 2000–2025 Google

Add Device

Search for a device by name

Name	Play	API	Width	Height	Density
 XR Device		34	2560	2558	320 dpi

XR Device

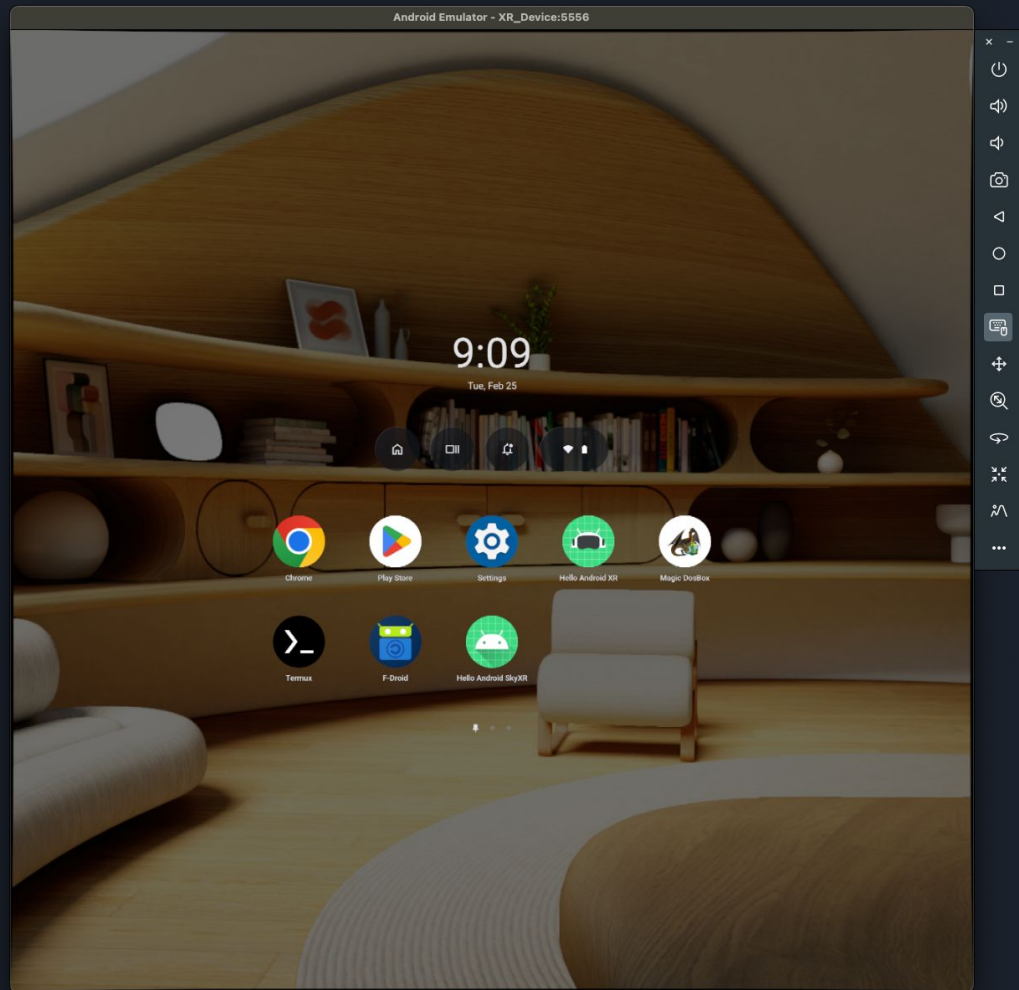
2560 px

2558 px

6.0"

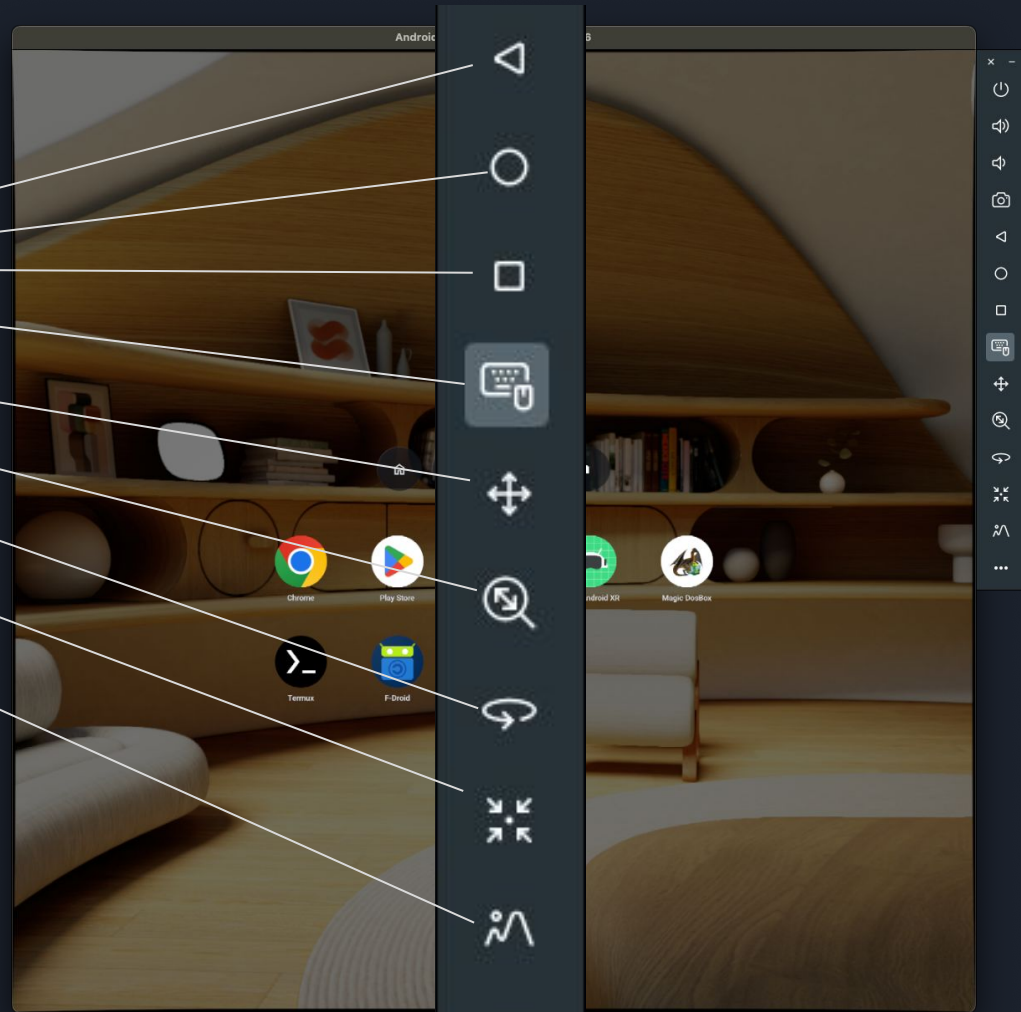
Device
OEM Google

Emulator



Emulator

- Android nav
- Tap / Drag
- Move in space
- Move forwards
- Rotate camera
- Recenter
- See through virtualenv



Emulator - Emulating Games



Storage
-1% used - 64.69 GB free

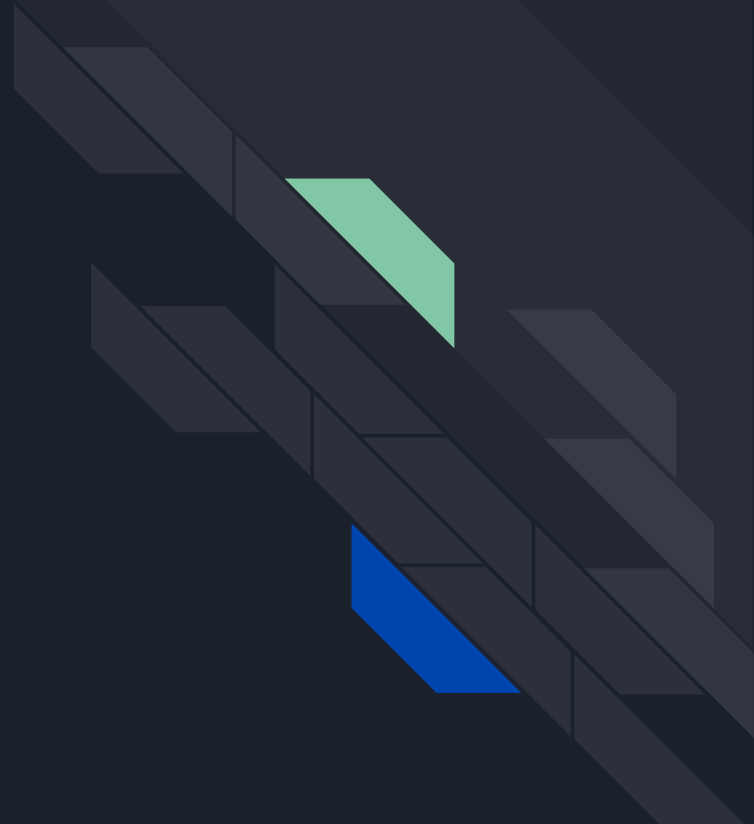




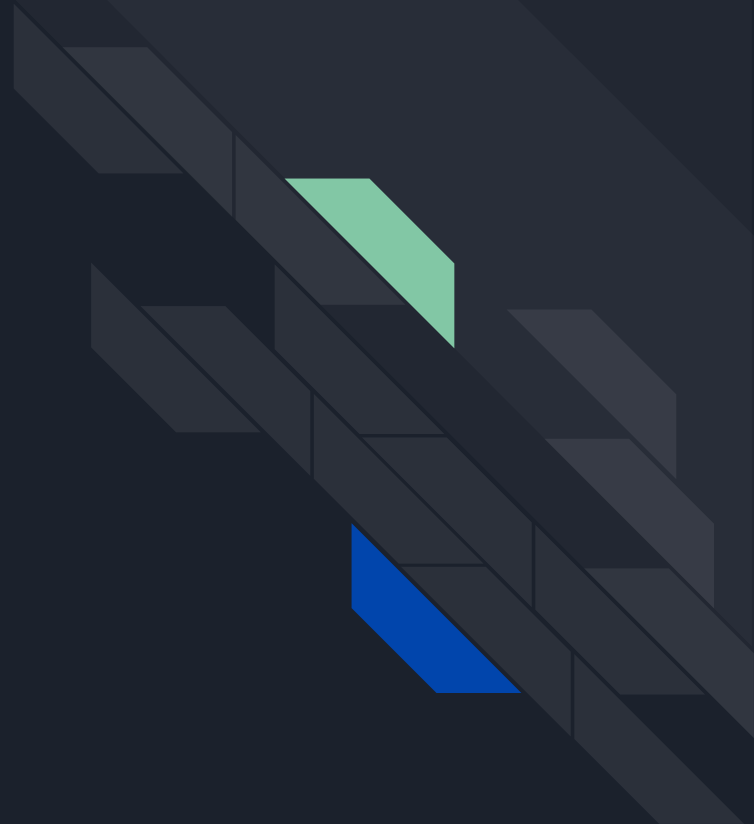


Android Apps

- F-droid
- Termux
- Play Store
- GApps
- Youtube
- No Netflix
- None in full spatial mode

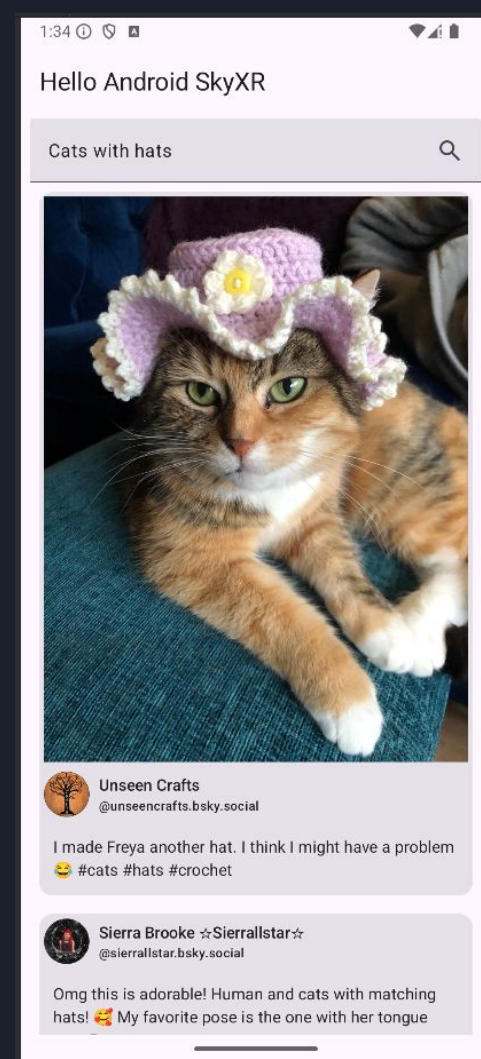


Code



SkyXR - A normal app.

- Search BlueSky for a term
- List result
 - on a card
 - with big picture
 - author
 - text
- VM
- Flows
- Android





Serialization

```
@Serializable
```

```
data class SearchResponse(  
    val posts: List<Post>,  
    val cursor: String? = null,  
)
```

```
@Serializable
```

```
data class Reference(  
    @SerializedName("\$link")  
    val link: String  
)
```

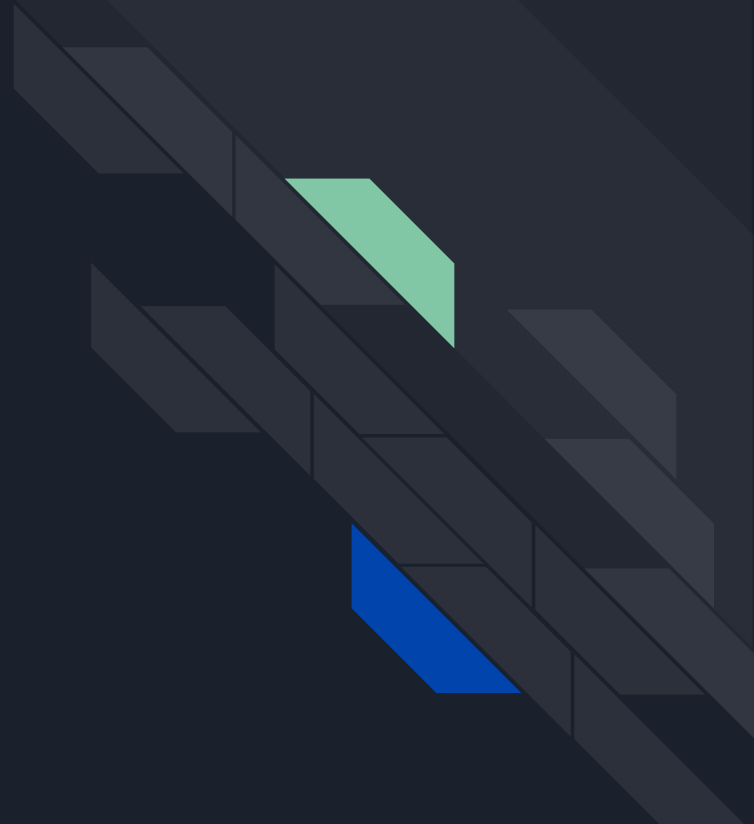


Retrofit

```
interface BskyService {
    @GET("xrpc/app.bsky.feed.searchPosts")
    suspend fun searchPosts(
        @Query("q") query: String,
        @Query("lang") language: String = "en",
        @Query("limit") limit: Int = 3,
    ): SearchResponse
}

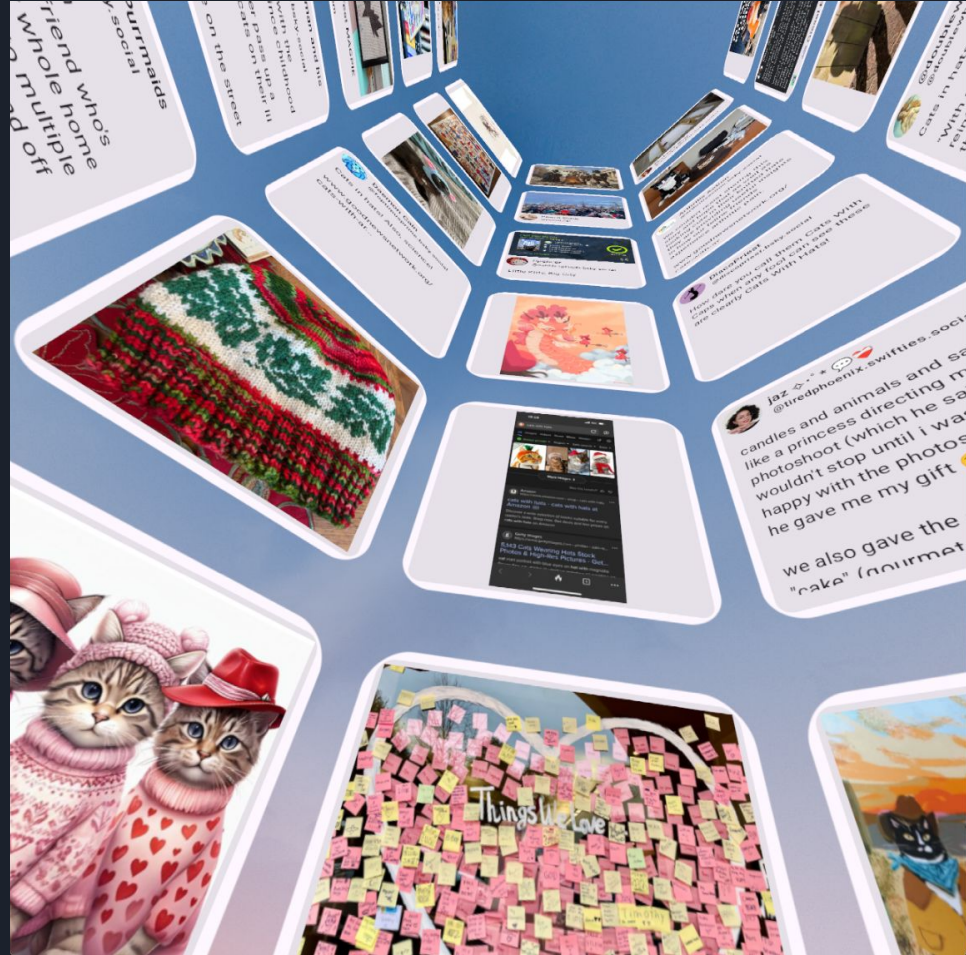
fun createBskyService(): BskyService = Retrofit.Builder()
    .baseUrl("https://api.bsky.app/")
    .addConverterFactory(
        Json { ignoreUnknownKeys = true }.asConverterFactory(
            "application/json".toMediaType()
        )
    )
    .build()
    .create<BskyService>(
        BskyService::class.java
    )
}
```

XPR



SkyXR - An extended app

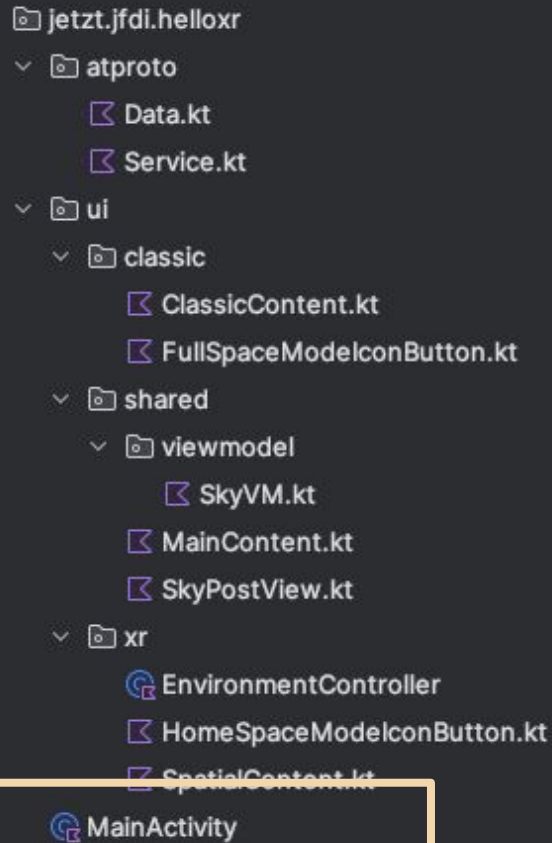
- Search BlueSky for a term
- List result in a cylinder!
- same logic as 2d
- same views
- differently organized





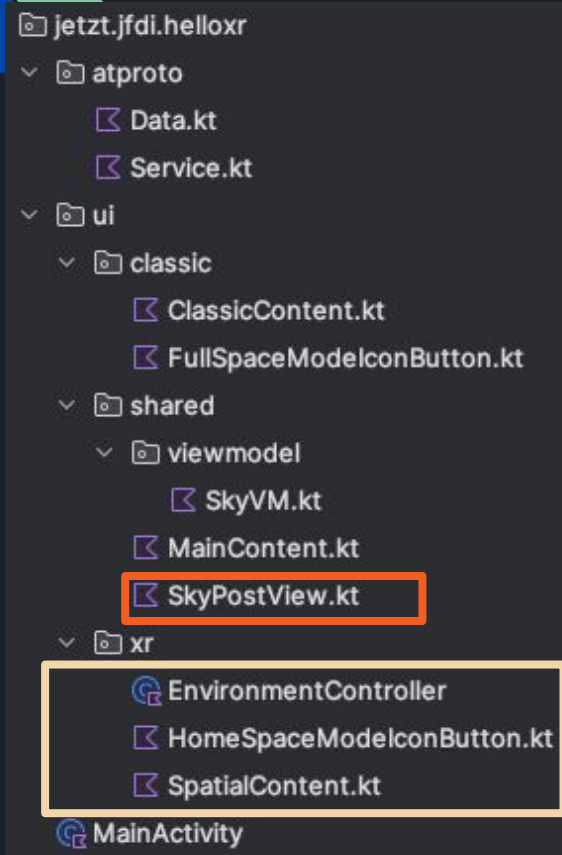
BYLINES CYMRU

Structure



```
if (LocalSpatialCapabilities.current.isSpatialUiEnabled) {  
    Subspace {  
        SpatialContent(  
            vm = vm,  
            onRequestHomeSpaceMode = {  
                environmentController?.requestHomeSpaceMode()  
            }  
        )  
    }  
} else {  
    ClassicContent(  
        vm = vm,  
        onRequestFullSpaceMode = {  
            environmentController?.requestFullSpaceMode()  
        }  
    )  
}
```


JetpackXR: Jetpack but Extended



```
SpatialColumn(  
    modifier = SubspaceModifier.fillMaxSize()  
) {  
    SpatialPanel(  
        modifier = SubspaceModifier.weight(1f)  
    ) {  
        Surface(  
            modifier = Modifier.fillMaxSize()  
        ) {  
            SkyPostView(  
                modifier = Modifier  
                    .width(postWidth.dp)  
                    .height(postHeight.dp),  
                post = post  
            )  
        }  
    }  
}
```

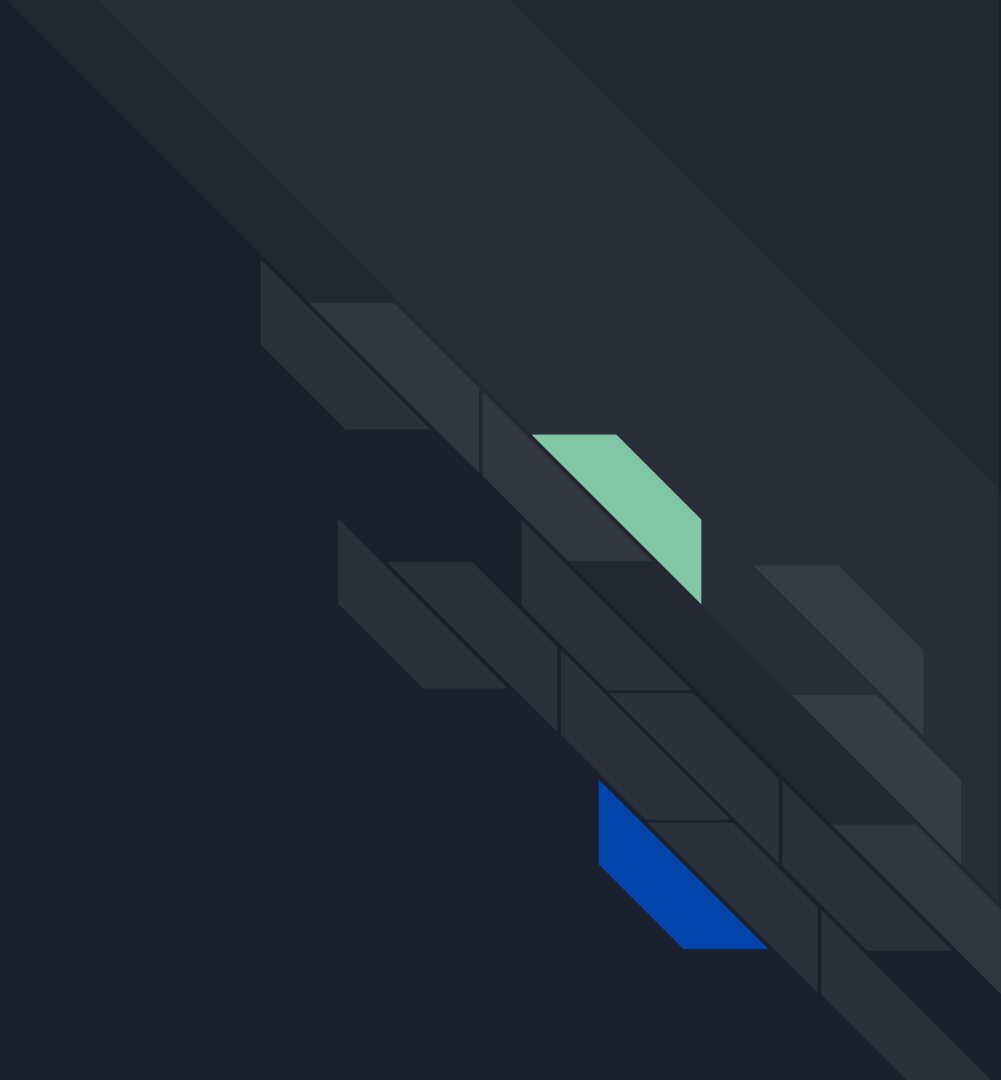
MATH

```
for (index in posts.indices) {  
    val angle: Float = (index % postsPerRound) * (360f / postsPerRound)  
  
    val x = normalizedAngleCos(angle / 360f) * postRadius  
    val z = normalizedAngleSin(angle / 360f) * postRadius  
    val y = (index / postsPerRound) * (postHeight + postPadding)
```

```
SpatialPanel(  
    SubspaceModifier  
        .offset(  
            x.dp, y.dp, z.dp  
        )  
        .rotate(  
            axisAngle = Vector3(0f, 1f, 0f),  
            rotation = 270f - angle  
        )  
        .padding(20.dp),
```

```
) {
```

Gotchas





 jfdi.jetzt/skyxr