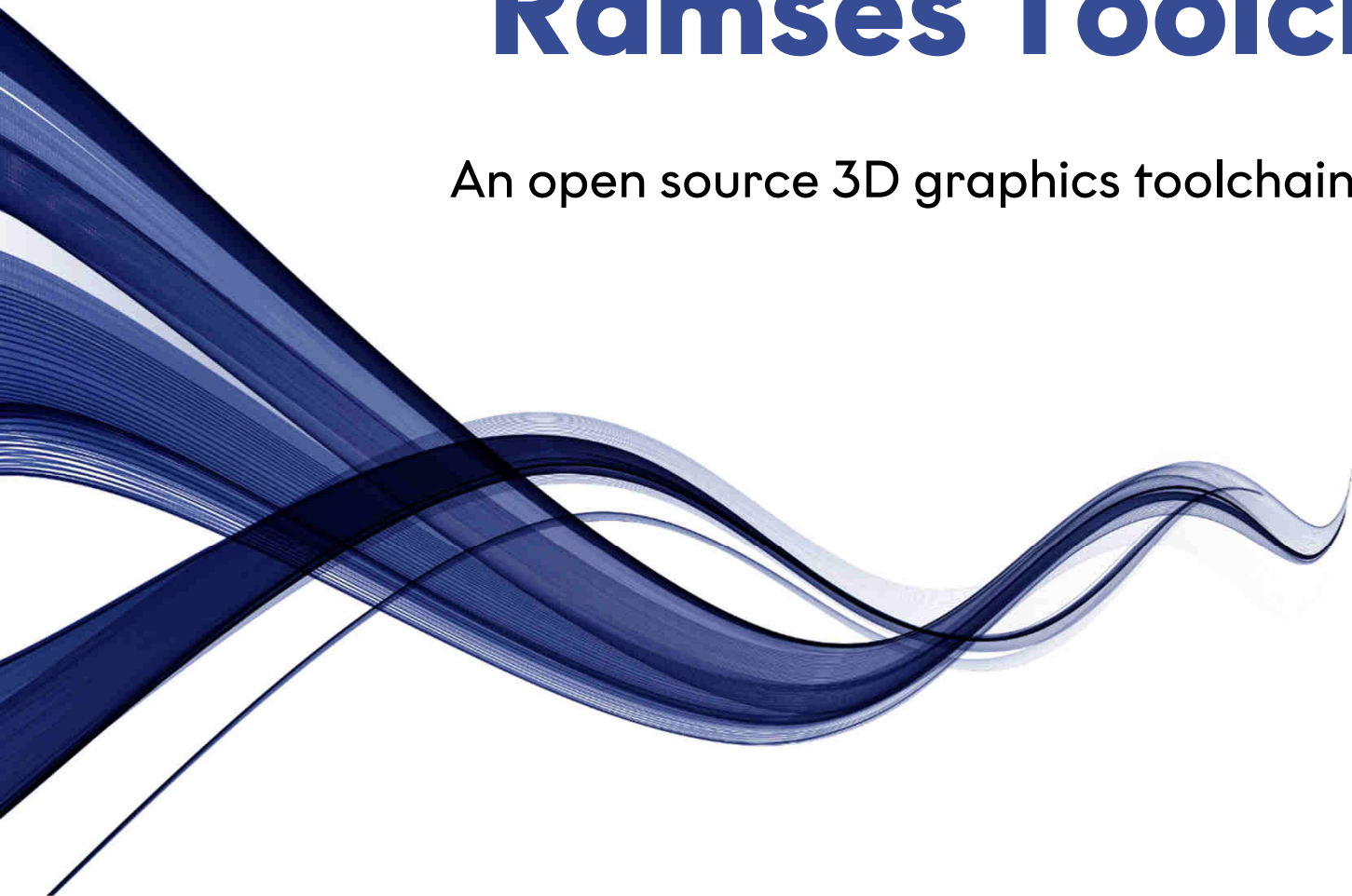




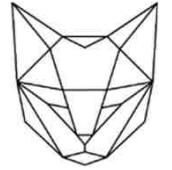
PARADOXCAT

Ramses Toolchain

An open source 3D graphics toolchain for Android



Introducing Ramses

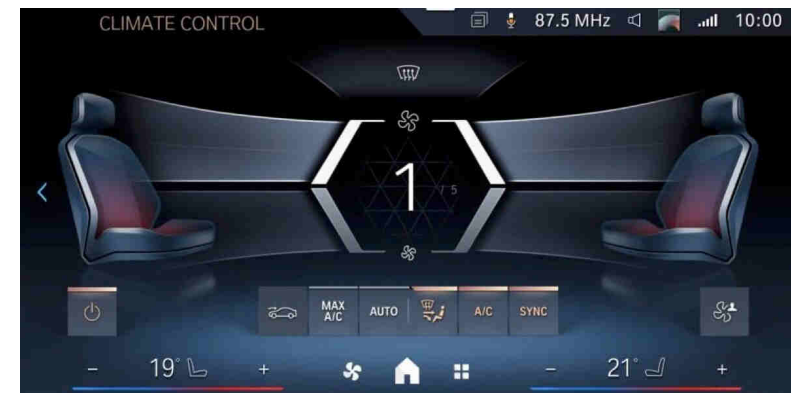
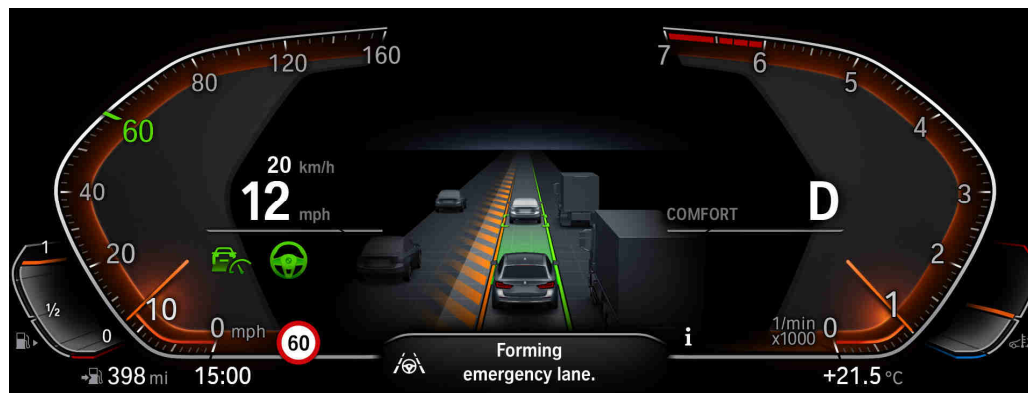


- Toolchain for graphics challenges on embedded/limited hardware
- Lean engine based on OpenGL ES 3.2
- Well-integrated into android
- Scales well for complex graphical assets
- You probably never heard about it 😊



Origin

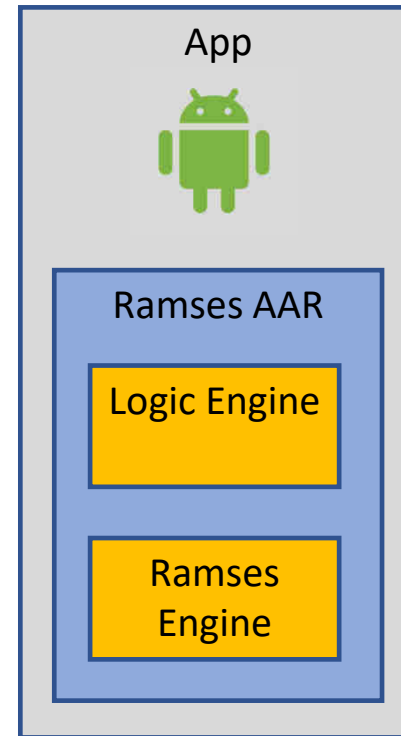
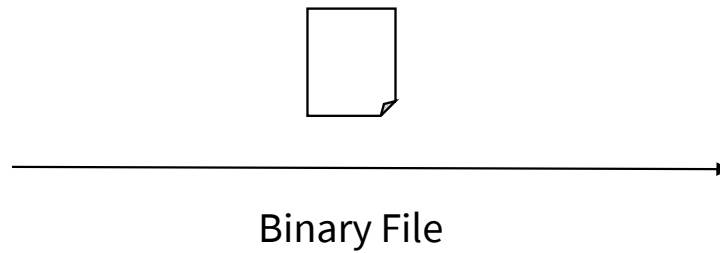
- Developed by BMW for in-car use
- Published as open source
- Powering vehicle HMIs and clusters since 2018



Toolchain Components



Workstation



Android device

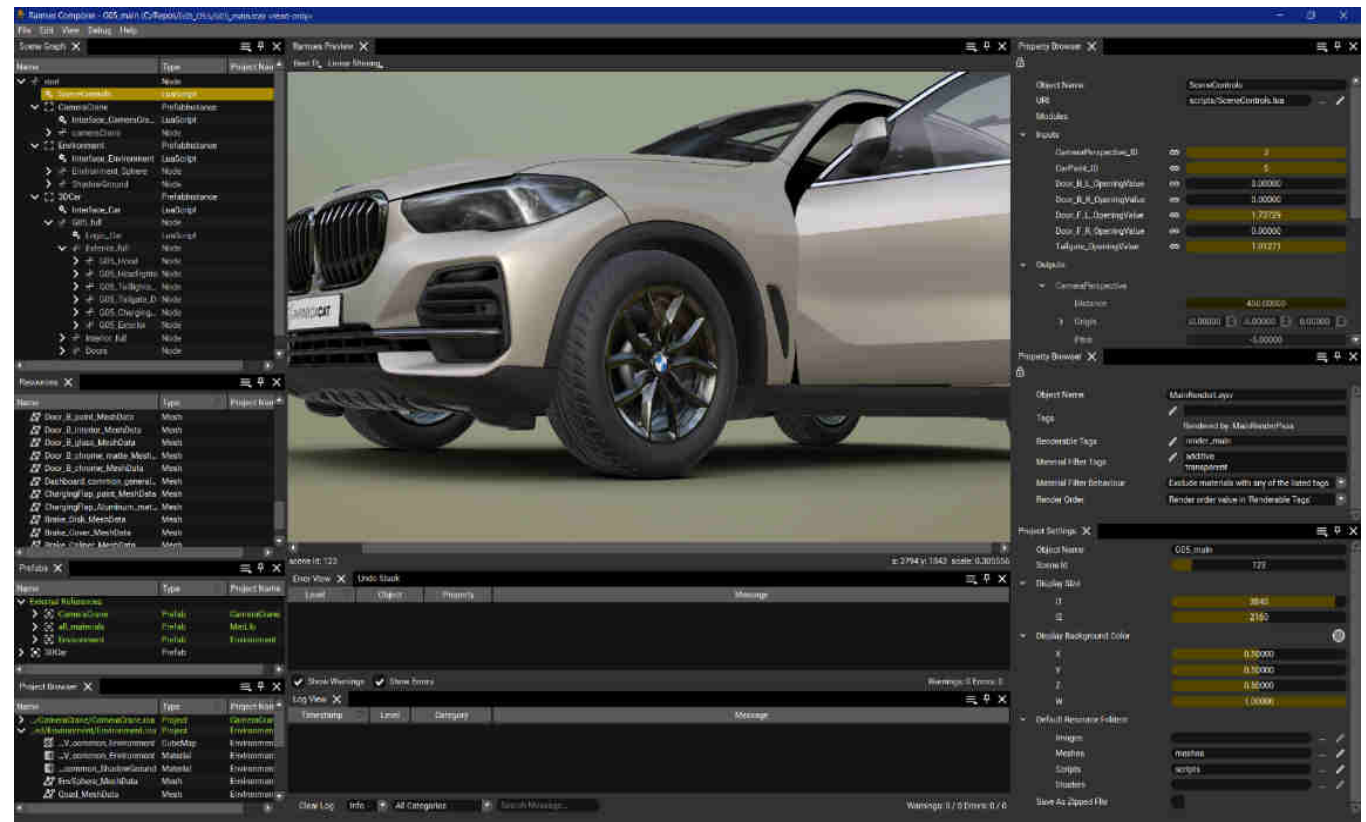
Interaction,
Animations,
Scripts

Scene graph,
Geometry

Ramses Composer

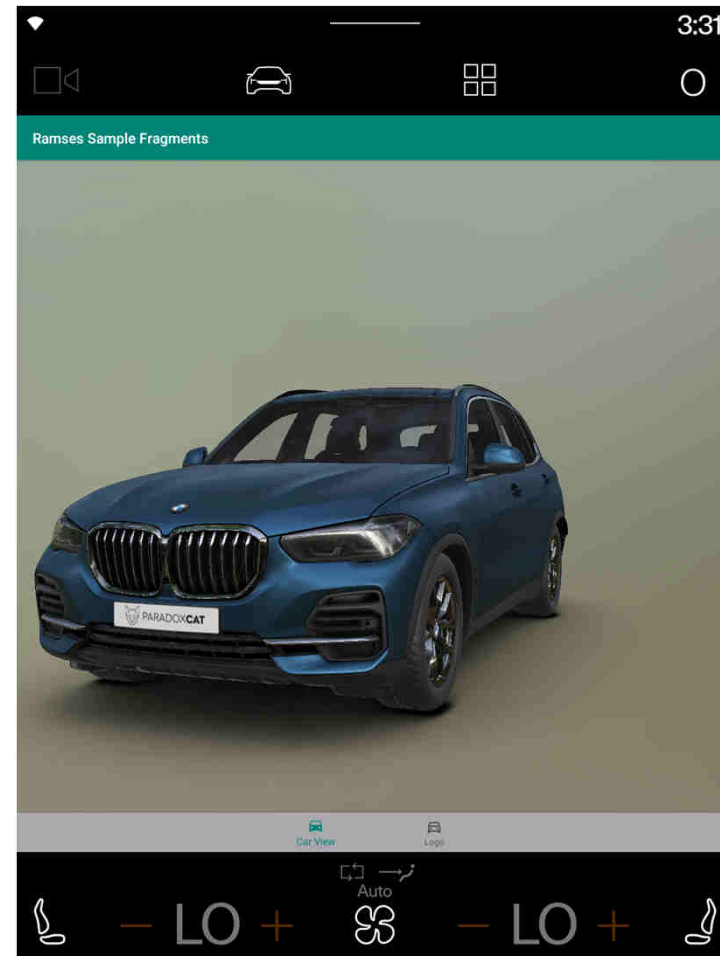


- Visual editor to create and manage Ramses content
- Compose complex 3D or 2D scenes
- Embraces open formats like glTF
- What-you-see-is-what-you-get (WYSIWIG)
- Preview rendered with Ramses: One engine – one truth



Concept

- Build your system with android
- Integrate 3D elements and complex assets with Ramses





Android Integration

- Add one dependency to build.gradle:

```
dependencies {  
    implementation "io.github.bmwcarit:ramses-aar:1.1.0"  
}
```

- Disable compression for Ramses assets:

```
android {  
    androidResources {  
        noCompress 'ramses', 'rlogic'  
    }  
}
```

Android Integration



- Create a SurfaceView or a TextureView to host RAMSES scene:

```
<SurfaceView
  android:id="@+id/ramsesSurfaceView"
  ...
/>
```


Android Integration



- Load scene from binary assets:

```
RamsesThread.initRamsesThreadAndLoadScene(requireActivity().assets, "G05.ramses", "G05.rlogic")
```

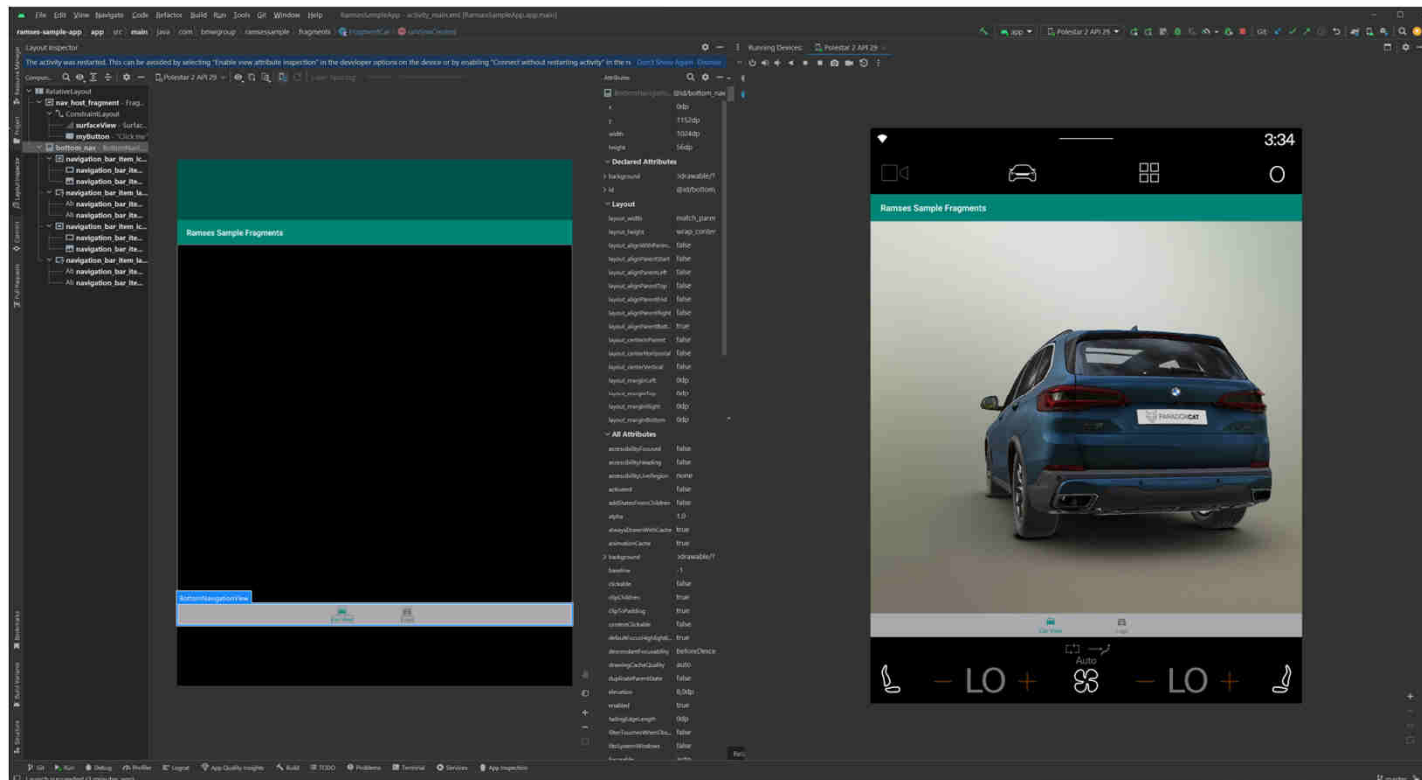
- Render to SurfaceView:

```
RamsesThread.createDisplayAndShowScene(surface)  
RamsesThread.startRendering()
```

Android Integration



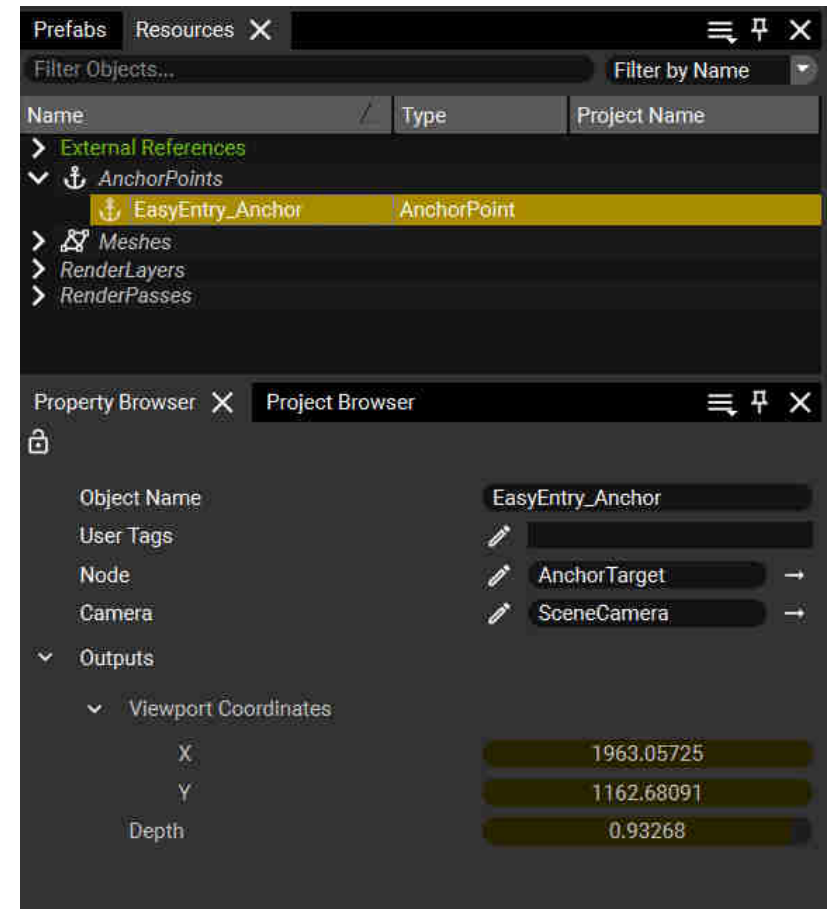
- A little more lifecycle management and layout creativity, and boom:



Anchor points



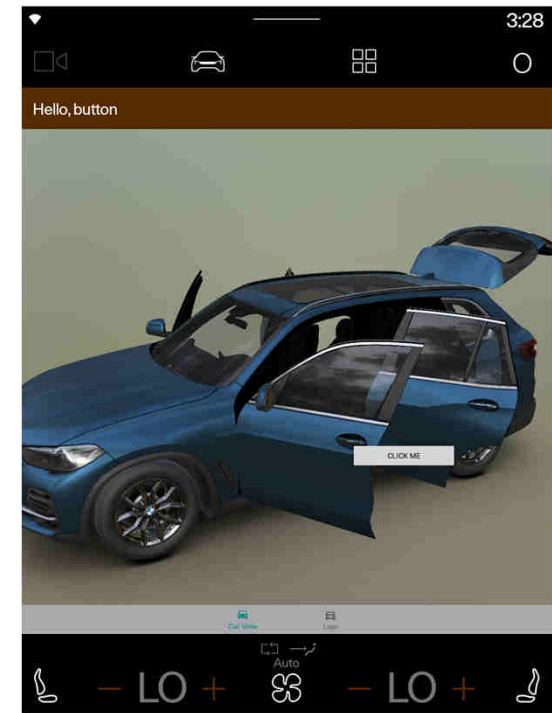
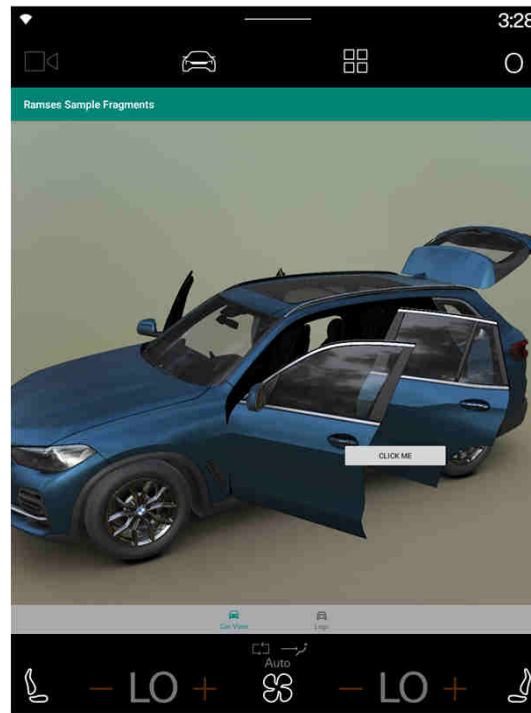
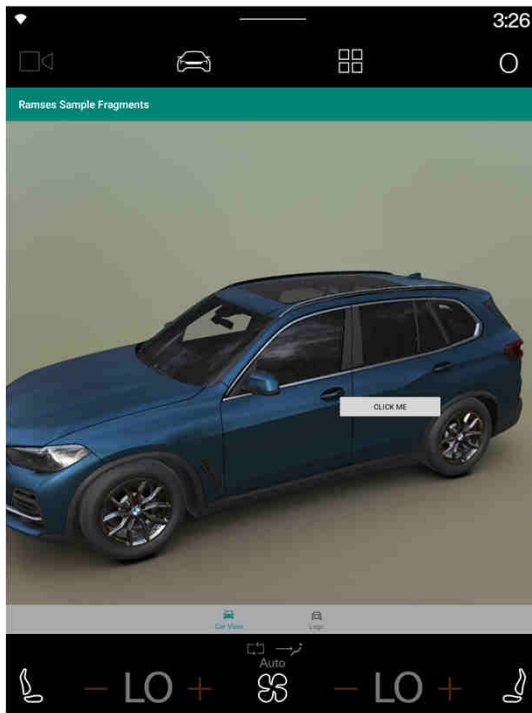
- Integrate Android elements on top of 3D objects, e.g.
 - Icons or labels
 - Arrows pointing at parts of a 3D model
- Anchor points are locations of 3D objects
- Available as screen coordinates
- Synced with movement or animations



Access Anchor Points



- How to make an android button follow the anchor point?





Access Anchor Points

- How to make an android button follow the anchor point?

```
override fun onSceneLoaded() {
    mMyFirstAnchorPoint = getLogicNodeRootOutput("EasyEntry_Anchor")
}

override fun onLogicUpdated() {
    val newPos = mMyFirstAnchorPoint?.getChild(0).vec2f
    // do something with newPos, e.g. expose a LiveData<PointF> from this class,
    // observe it from Fragment/Activity, and update button position
}

private var mMyFirstAnchorPoint: Property? = null
```



Alternatives

- Why not use existing open source project?
- Filament
 - No scripting, no editor
- QT 3D Studio
 - LGPL license problematic for e.g. automotive use
- Game engines
 - Typically too heavy for non-game apps
 - Often limited to building whole apps



When to use

- You need complex 3D graphics on an embedded device with performance and/or memory limits
- You need to create a large number or a set of exchangeable 3D objects
- You want to combine graphics originating from different apps or devices over network



When not to use

- You mostly have 2D UI/UX
- You need a standalone 3D application
 - Ramses is relying on Android for input events, layout and text rendering
- Hardware limitations are not a problem for you

Conclusion

- Ramses is a quite powerful 3D graphics toolchain
- Integrates well with Android
- Available as Open Source
 - Mozilla Public License (MPL 2.0)
- In active development
 - Used in HMI of production vehicles
 - Community feedback is welcome



<https://ramses3d.org/>



[Why choose Ramses](#) [When to use](#) [Take a look](#) [Contributors](#)

[GETTING STARTED](#)

RAMSES
3D GRAPHICS FOR CARS AND IOT

Ramses is a low-level rendering engine which is optimized for embedded hardware
- mobile devices, automotive ECUs, IoT electronics.

[GETTING STARTED](#)

The banner features a dark background with a blurred image of a car's infotainment screen. The screen displays various UI elements: a "ConnectedDrive" status bar at the top, a speedometer showing "49.5 km/h", a notification area with "Notifications" and "7:15", and a large "0" on the right side. The text "RAMSES" and "3D GRAPHICS FOR CARS AND IOT" is overlaid in white and orange. A red "GETTING STARTED" button is positioned at the bottom center.