Capture and Replay of Vulkan and OpenXR

Mark Young and John Zulauf
LunarG Senior Graphics Engineers

July 2025 – Siggraph Vancouver

Agenda

- GFXReconstruct (GFXR) Overview
- OpenXR Proof-of-Concept
- Challenges
- Using GFXR on Meta Quest 3/Pro
- Tool Output Examples
- Making GFXR Replay Work
- Verified Apps/Platforms
- Future Improvements

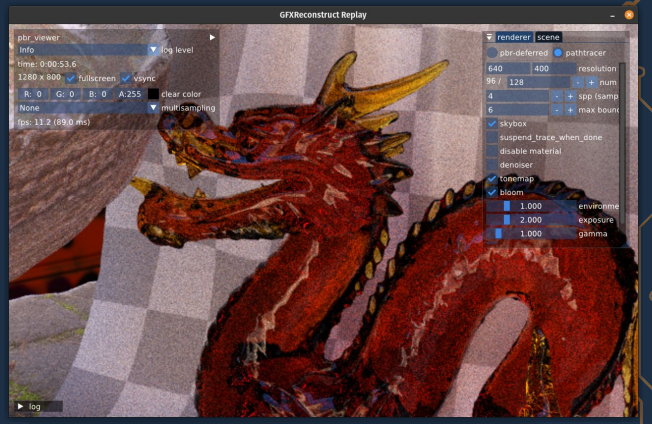
This is **not** going to be a detailed tutorial on using GFXReconstruct!

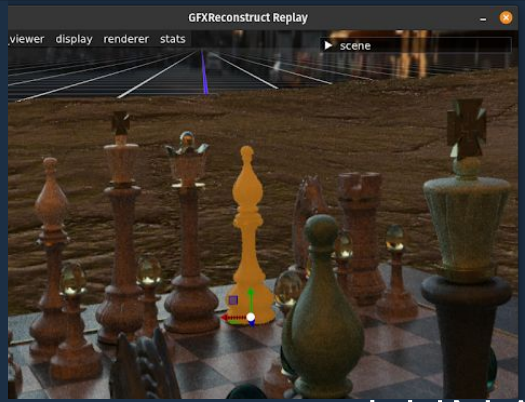
GFXReconstruct (GFXR) Overview

- GFXR started in 2019
  - Replacement for Vulkan VkTrace
- Captures application API usage (including bugs)
  - API Commands
  - Important Memory [Buffers/Images (Textures)]
  - OS-specific items as well
    - E.g. Android Hardware Buffers
- Content written to binary capture file (.gfxr)
  - Internally may use compression

GFXReconstruct Support

- Supports Vulkan and D3D12
  - Raytracing
  - Mesh Shading
- Regular Updates
  - Vulkan Header/Agility SDK (D3D12) releases
  - Code generation
- Github/Internal CI testing to maintain quality





KHRONOS GROUP

LUNARG POWER YOUR SUCCESS

GFXReconstruct (GFXR) Principles

FIDELITY	INTEGRITY	PORTABILITY	PERFORMANCE
Capture & Playback on the Same Device, with Identical Results	Optimizations that Stay True to Application Behavior	Playback Across a Broad Range of Devices, with Variable Fidelity	Deliver the Performance Required for Usability & Interactivity

➔ Trade offs are often required, but they should be evaluated against these principles

GFXReconstruct Usage

Image Generated
With Google Gemini

- Useful For:
  - Replaying / Sharing Bugs
  - Testing for Regressions
  - Analyzing API Usage
  - Performance Profiling

- Becoming the “engine” behind many proprietary GPU profiling/debugging tools
  - Used by HW and SW companies you know!

GFXR Is a Collection of Tools

- Capture Layers/Libraries
  - Captures API contents and stores necessary info in a capture file
- gfxrecon-replay
  - Replays a provided capture file
- gfxrecon-info
  - General high-level info about a capture file
- gfxrecon-convert
  - Output the contents of a capture file in a JSON readable format
- And more...


Now for OpenXR
KHRONOS GROUP
LUNARG
POWER YOUR SUCCESS

OpenXR Proof-of-Concept

- Initial OpenXR 1.1 support paid for by a generous Client
- Limited Support
  - Works only with Vulkan graphics API
  - Small set of **verified** extensions
  - Single threaded apps
- Not enabled in SDK builds of GFXR
  - But enabled by default building from GitHub

Challenges

- Atoms and opaque values (not just handles)
- Parent/Child structures (*BaseHeader)
- Different ways to create Vulkan Instance/Devices
- Enabling API layers on the headset could be easier
- Capture Session lifecycle is encoded in the event stream
- Validation layers are insufficient to detect issues

Challenges (cont)

- Re-Entrance Problem

Application

```
Vulkan Call

Vulkan Call

OpenXR Call




Vulkan Call

Vulkan Call
```

OpenXR Runtime

```
OpenXR Call  {
    Vulkan Call

    Vulkan Call
}
```

GFXR Capture File

```
Vulkan Call

Vulkan Call

OpenXR Call

Vulkan Call

Vulkan Call

Vulkan Call

Vulkan Call
```

Challenges (cont)

- Re-Entrance Problem

Application

```
Vulkan Call

Vulkan Call

OpenXR Call




Vulkan Call

Vulkan Call
```

OpenXR Runtime

```
OpenXR Call  {
    Vulkan Call

    Vulkan Call
}
```

GFXR Capture File

```
Vulkan Call

Vulkan Call

OpenXR Call
```

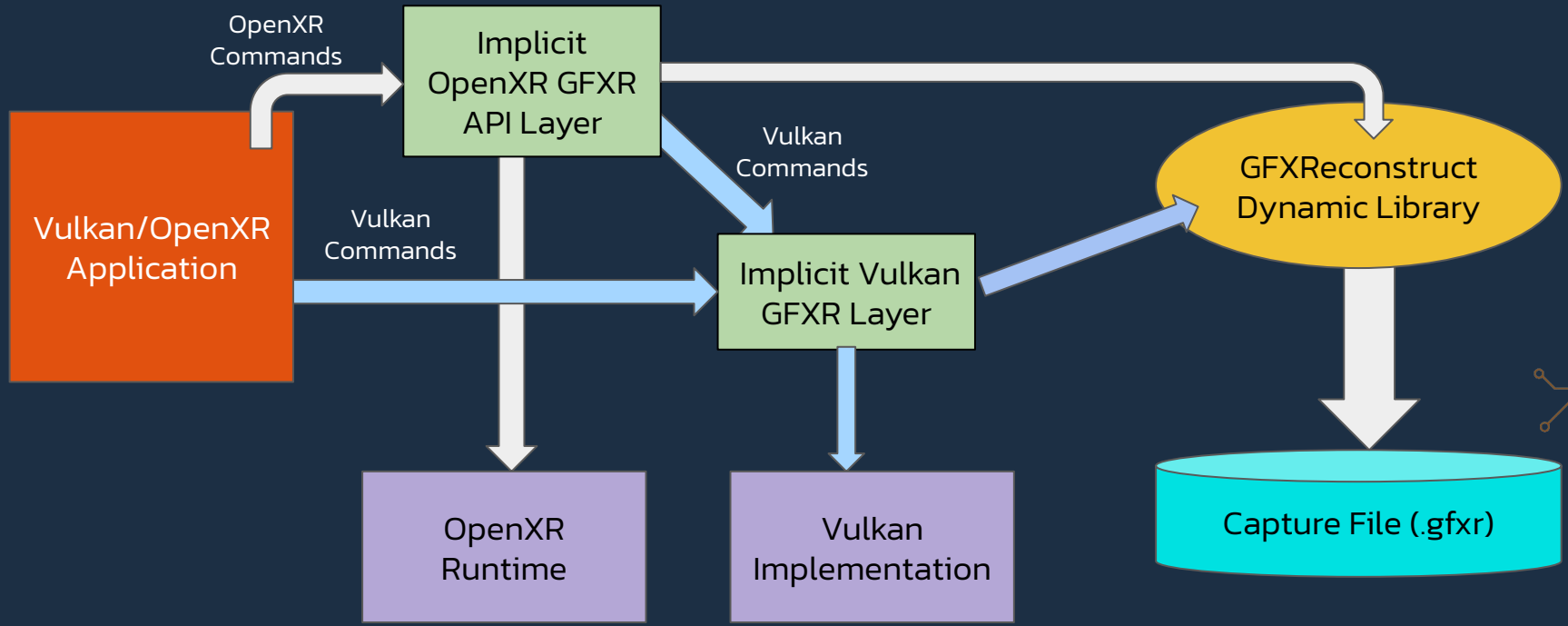
We Don't Want This During Replay

```
Vulkan Call

Vulkan Call
```

Using GFXR on Meta Quest 3/Pro

- **Recursively** pull down GFXReconstruct source from GitHub
- Follow instructions in HOWTO_meta_quest.md
  - Requires Gradle changes to your project
    - Add GFXR layer as dependency
      - OpenXR API Layer Manifest (.json)
      - GFXR library (.so)
  - Release app, **must** be made debuggable

Capturing Process



gfxrecon-info

```
Exe info:
    Application exe name: org.khronos.OpenXRTutorialChapter4
    Application version: 0.0.0.0
    Application Company name:
    Product name:
```

```
File info:
    Compression format: LZ4
    Total frames: 241
```

```
Vulkan physical device info:
    Device name: Adreno (TM) 650
    Device ID: 0x6050002
    Vendor ID: 0x5143
    Driver version: 2150780928 (0x80325000)
    API version: 4198695 (1.1.295)
```

gfxrecon-info (cont)

```
Vulkan device memory allocation info:
    Total allocations: 4
    Min allocation size: 4096
    Max allocation size: 28672
```

```
Vulkan pipeline info:
    Total graphics pipelines: 1
    Total compute pipelines: 0
    Total raytracing pipelines: 0
```

```
OpenXR info:
    Header Version: 1.1.40
    Application name: OpenXR Tutorial Chapter 4
    Application version: 1
    Target API version: 34 (0.0.34)
```

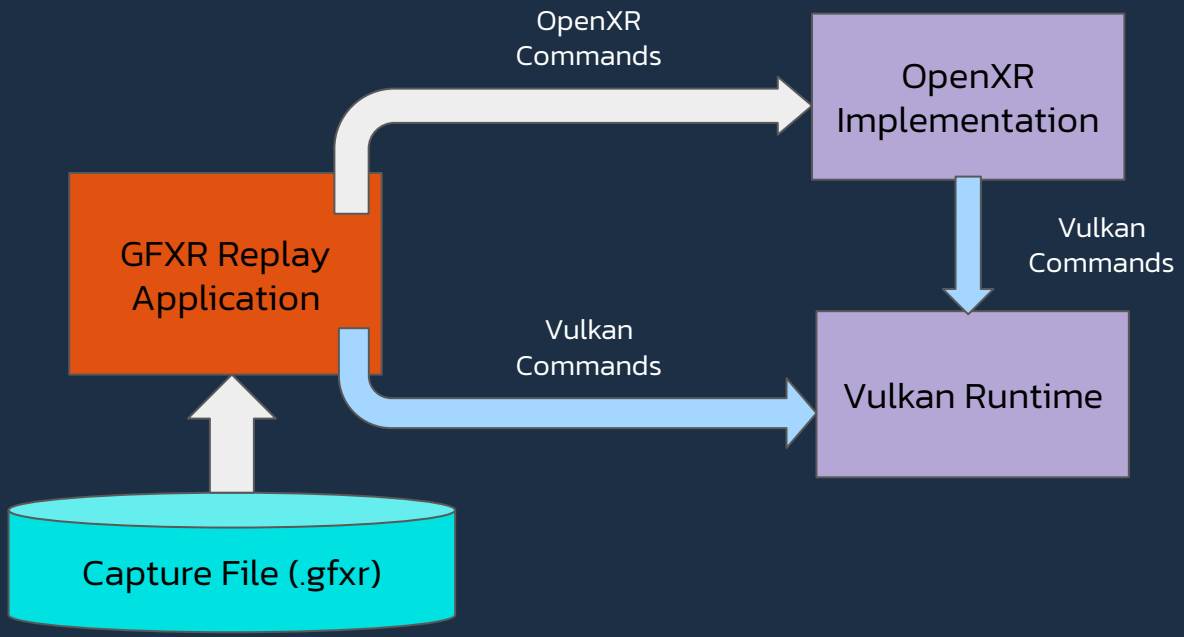
gfxrecon-convert Example

```
{
  "index": 145,
  "function": {
    "name": "xrPollEvent",
    "thread": 1,
    "return": "XR_SUCCESS",
    "args": {
      "instance": 1,
      "eventData": {
        "type":
"XR_TYPE_EVENT_DATA_REFERENCE_SPACE_CHANGE_PENDING",
        "session": 34,
        "referenceSpaceType":
"XR_REFERENCE_SPACE_TYPE_STAGE",
        "changeTime": 14834752945132,
        "poseValid": true,
        "poseInPreviousSpace": {
          ...
```

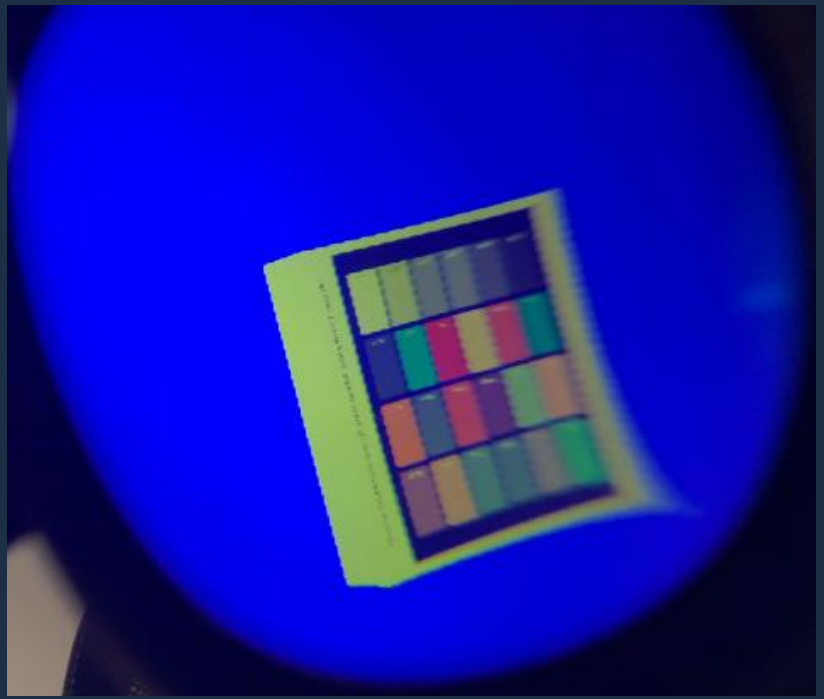
KHRONOS GROUP

LUNARG POWER YOUR SUCCESS

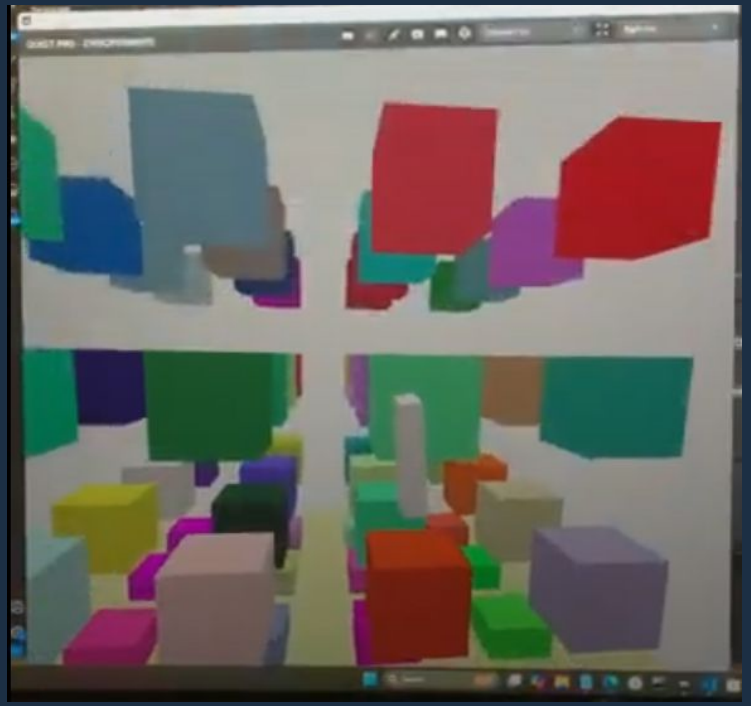
gfxrecon-replay



Seeing Results



In Headset



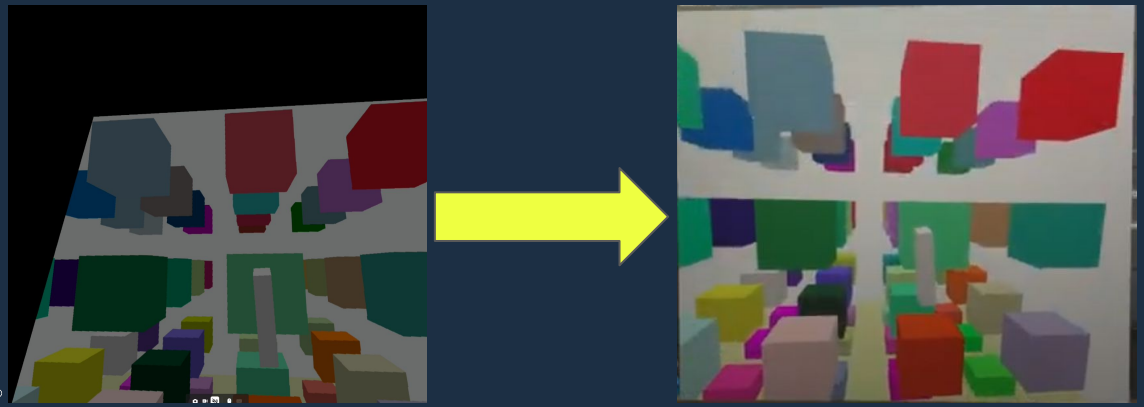
Desktop using Casting

Making GFXR Replay Work

- Track/replace handles, atoms, opaque values
- Fill in memory (after allocation) [Metadata]
- Replace `xrCreateApiLayerInstance` with `xrCreateInstance`
- Virtual Swapchain
  - Swapchain images can be returned in different order

Making GFXR Replay Work (2)

- Update times in certain commands (`xrWaitFrame`/ `xrEndFrame`)
  - Relative diff between predicted and display times
- Remap Views during replay to current headset View



Making GFXR Replay Work (3)

- Replay OpenXR Session Lifecycle Handling
  - Events may occur in different orders
  - New events may occur, other events may never occur
- GFXR Quest Replay Android manifest enables a lot of features

Image Generated
With Google Gemini

Re-entrance Multithreading Issue

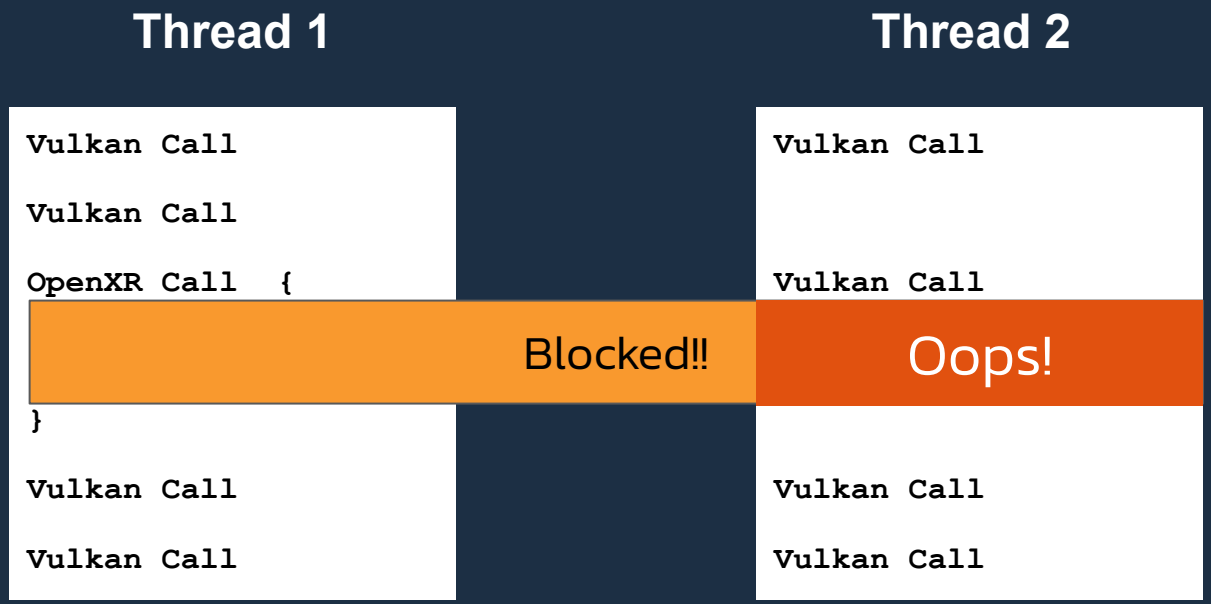
Thread 1

```
Vulkan Call

Vulkan Call

OpenXR Call  {
    Vulkan Call

    Vulkan Call
}

Vulkan Call

Vulkan Call
```

Thread 2

```
Vulkan Call



Vulkan Call


Vulkan Call


Vulkan Call

Vulkan Call
```

Re-entrance Multithreading Issue



Future Improvements

- Test against more applications (more extensions)
- Test against more headsets (AndroidXR, …)
- Add screenshot capture for headsets
- Implement trimming support
  - Enables fine-tuning a capture down to a few frames

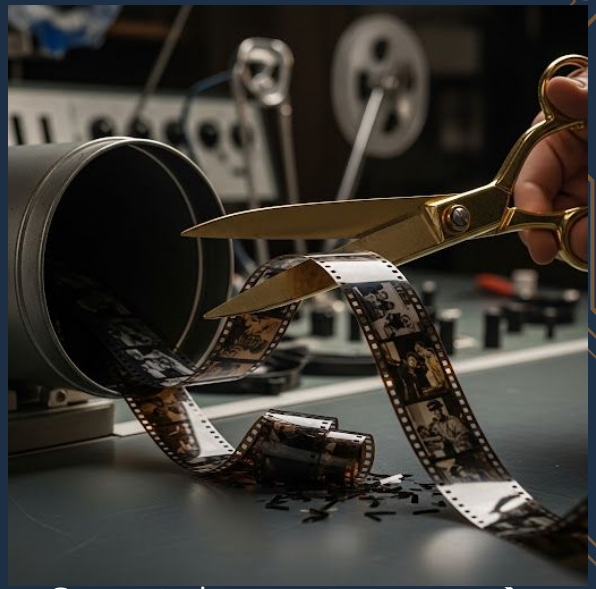


Image Generated With Google Gemini

Verified Apps and Platforms

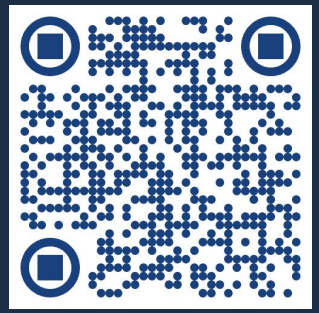
- Applications
  - HelloXR Sample
  - OpenXR Tutorials (up to Chapter 4)
    - Chapter 5 failed to compile for me
  - OpenXR Provider_v2
  - Meta's IGL OpenXR Sample

- Using the following runtimes
  - Meta Quest Pro/3
  - Monado (Windows/Linux)

OpenXR Provider_v2 by Rune Berg

GFXReconstruct Resources

- More Detailed Videos (Brad Grantham):
  - XDC 2022 / Vulkanised 2023
- Source in GitHub



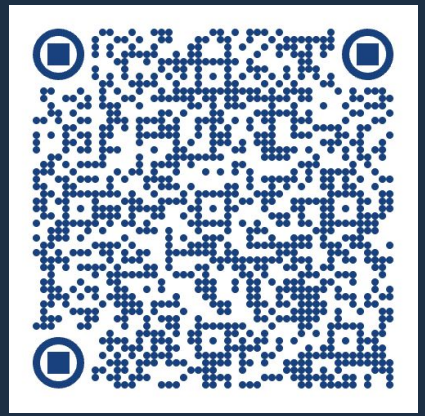
- HOWTO_meta_quest.md

Summary

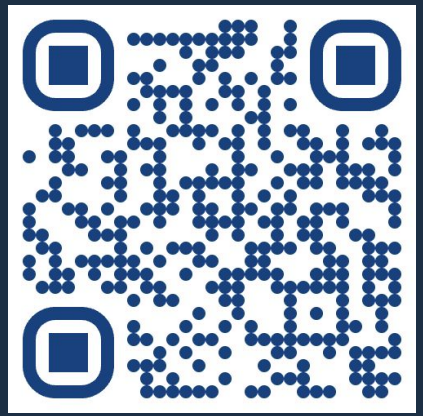
- GFXReconstruct continues to grow
- PoC capture and replay of OpenXR + Vulkan is there now
  - May not handle all cases right now
- Help us verify functionality by using GFXR

THANK YOU!

Download LunarG Presentations



LunarG Website



KosmicKrisp BLOG Post

KHRONOS GROUP

LUNARG POWER YOUR SUCCESS