Khronos Fast Forward

Neil Trevett
Khronos President and VP Developer Ecosystems at NVIDIA
Khronos Connects Software to Silicon

Open, royalty-free interoperability standards to harness the power of GPU, XR and multiprocessor hardware

3D graphics, augmented and virtual reality, parallel programming, inferencing and vision acceleration

Non-profit, member-driven standards organization, open to any company

Proven multi-company governance and Intellectual Property Framework

Founded in 2000
~ 200 Members | ~ 40% US, 30% Europe, 30% Asia
Khronos Standards for Spatial Computing

- **Embedded camera, sensor and ISP**
- **Sensor Stream**
- **Vision and inferencing acceleration**
- **Runtime Control**
- **Pose and scene data**
- **XR displays and devices**
- **Runtime Control**
- **Pixels**
- **3D rendering acceleration**
- **Runtime Control**
- **Augmentation 3D Assets**
- **OpenVX**
- **SYCL**
- **OpenCL**
- **OpenXR**
- **Vulkan**
- **ANARI**
- **WebGL**
**Khronos Active Standards**

**Khronos standards most relevant at SIGGRAPH**

- **3D Graphics**
  - Desktop, Mobile, Web
  - WebGL
  - ANARI
  - OpenGL
  - OpenGL ES
  - Vulkan

- **3D Asset**
  - Authoring/Delivery
  - COLLADA

- **Portable XR**
  - Augmented and Virtual Reality
  - OpenXR

- **Parallel Computation**
  - Vision, Camera, Inferencing, Machine Learning
  - OpenCL
  - Kamaros
  - OpenVX
  - NNEF
  - SYCL
  - SPIR

- **Safety Critical APIs**
  - Vulkan SC
  - OpenGL SC
  - SYCL SC

**NEW!** New Initiative under discussion - Slang Shading language
NVIDIA has proposed to bring Slang to Khronos

Slang is a fast-moving language designed specifically to meet the evolving needs of real-time rendering applications - including neural shaders

The Slang compiler also provides onramps for existing HLSL and GLSL code bases

Currently being discussed under Khronos’ New Initiative Process

‘Slang in Vulkan’ presentation at the Vulkan BOF - 3PM Wednesday

Multiple backends enable shader code to be written once to run almost anywhere

Slang
Open Source Compiler
github.com/shader-slang/slang

GLSL
HLSL 2020

HLSL
Direct3D

GLSL
OpenGL|ES

CUDA
Optix
C (CPU)
PyTorch
Metal Shading Language (in progress)
WGSL for WebGPU (coming soon)
OpenXR
Empowering Cross-Platform Immersive Experiences

Neil Trevett
NVIDIA
OpenXR Cross-Platform Portability

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.

Applications and engines can portably access any OpenXR-conformant hardware
## Conformant OpenXR Devices

<table>
<thead>
<tr>
<th>HoloLens and Mixed Reality Headsets. Hand and eye tracking extensions</th>
<th>Rift S, Quest 3, Quest 2 and Quest Pro Meta Deprecated own API for OpenXR</th>
<th>Vive Focus 3, Vive Cosmos, Vive XR Elite, Vive Wave Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valve Index. Valve Deprecated OpenVR APIs for OpenXR</td>
<td>All Varjo Headsets are fully compliant XR-3, XR-4</td>
<td>MREAL X1</td>
</tr>
<tr>
<td>Magic Leap 2</td>
<td>XREAL Air 2, Air 2 Pro, Air 2 Ultra</td>
<td>Qualcomm Snapdragon Spaces XR Development Platform</td>
</tr>
<tr>
<td>Spatial Labs Display Series</td>
<td>Neo 3 and Pico 4</td>
<td>Spatial Reality Displays</td>
</tr>
</tbody>
</table>
The OpenXR Story So Far...

Vendor Proprietary API fragmentation
Clear industry demand need for a cross-platform XR open standard

OpenXR Working Group Formed
2017

Establishing baseline XR functionality
Though industry consensus and contributed designs
OpenXR 1.0 specification drafted

2019

OpenXR 1.0 Released

OpenXR achieves wide industry adoption
OpenXR is foundation for experimentation
New functionality introduced through extensions

OpenXR Working Group Formed
2017

OpenXR 1.0 Released

OpenXR 1.1 Released
April 2024

OpenXR 1.1
Consolidates multiple extensions to streamline application development and reduce fragmentation
Adds new functionality with spec improvements

Increased focus on regular core spec updates
Balancing the need to ship new functionality AND consolidate widely proven technology

Leverage OpenXR’s flexible design to explore new use cases
e.g., body tracking and advanced spatial computing

Empowering Cross-platform Immersive Experiences

OpenXR® Group Inc. 2024 - Page 9
This work is licensed under a Creative Commons Attribution 4.0 International License
OpenXR 1.1 Key Extensions Promoted to Core

- **Local Floor Reference Space**
  - Gravity-aligned world-locked origin for standing-scale content
  - Estimated floor height built in
  - Recenter to current user position at the press of a button without a calibration procedure

- **Grip Surface**
  - Anchors visual content relative to the user’s physical hand
  - Can be tracked directly or inferred from a physical controller’s position and orientation

- **Stereo with Foveated Rendering for XR headsets**
  - Runtimes MAY optionally expose eye-tracked or fixed foveated rendering
  - Portable across multiple graphics rendering APIs

- **Additional enhancements**
  - Interaction Profile improvements
  - Spec language cleanup and clarifications
## Engines, Browsers, and Libraries with OpenXR

<table>
<thead>
<tr>
<th>Engine/Plugin</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unreal Engine</td>
<td>Unreal has been providing support since 4.24. UE 5.0 supports OpenXR</td>
</tr>
<tr>
<td>Unity</td>
<td>Unity’s OpenXR plugin available since 2020 LTS</td>
</tr>
<tr>
<td>Godot</td>
<td>Godot provides OpenXR support since March 2023 (Core 4.0 Alpha 4)</td>
</tr>
<tr>
<td>Autodesk VRED Library</td>
<td>OpenXR supported since VRED 2023.4</td>
</tr>
<tr>
<td>NVIDIA Omniverse and CloudXR Platforms</td>
<td>WebXR in Chrome, Edge, and Firefox uses OpenXR as the default backend</td>
</tr>
<tr>
<td>Collabora and Mono</td>
<td>Open-source OpenXR Implementation</td>
</tr>
<tr>
<td>Meta</td>
<td>A lightweight XR Meta XR Simulator to Speed Unity OpenXR Development</td>
</tr>
<tr>
<td>stereokit</td>
<td>Open-source mixed reality library for building HoloLens and VR applications</td>
</tr>
</tbody>
</table>
Khronos and W3C: Bringing XR to the Web

XR Applications and Engines use an API from both the 3D and XR Stacks

3D Stack
Driving GPUs to render scenes and augmentations

XR Stack
Handling XR Devices for creating UI

Engines
three.js  babylon.js  unity  Unreal Engine  GODOT

WebGL™  WebGPU

WebXR

W3C®

OpenGLES™  Vulkan®  OpenXR™
Coming Soon…

• Extending hand tracking
  - To include full body tracking

• Enhanced handling of spatial entities
  - Standardized methods to interact with the user’s environment
  - Support for advanced spatial computing applications

• Expanded haptics support
  - Support immersive experiences through PCM, vibrotractiles, and transients

• Controller render models (gltF)
  - Showing and animating a model of the user’s actual controller

‘Slang in Vulkan’ presentation at the Vulkan BOF - 3PM Wednesday
Vulkan: Forging Ahead!
(fast forward version)

Tom Olson (Arm), Vulkan Working Group chair
Ten years ago, in a city far, far away...

OpenGL / OpenGL ES BOF, SIGGRAPH 2014
Vulkan BOF Presentations

Vulkan: Looking back, and looking forward
- Tom Olson (Vulkan Working Group Chair / Arm)

Vulkan SDK: Where we started, and where we are going
- Karen Ghavam (LunarG)

Vulkan: Crash Diagnostic Layer
- Jeremy Gebben (LunarG)

Slang in Vulkan
- Hai Nguyen (NVIDIA)

EVOLVE - Next Generation Benchmarking
- Jasper Bekkers and Darius Bouma (Traverse Research)

Adding Vulkan to Pixar’s Hydra Storm Renderer
- Henrik Edstrom (Autodesk), Ashwin Bhat (Autodesk), and Caroline Lachanski (Pixar)
Vulkan Roadmap 2024 Profile

Represents the second milestone on the Vulkan Roadmap
- Captures expected feature set for “immersive graphics” 2024-2026+
Vulkan Roadmap 2026 Milestone

“You’ve got to be very careful if you don’t know where you’re going, otherwise you might not get there”
- Yogi Berra

Our plans for the third milestone on the Vulkan Roadmap
Vulkan BOF Presentations

Vulkan: Looking back, and looking forward
- Tom Olson (Vulkan Working Group Chair / Arm)

Vulkan SDK: Where we started, and where we are going
- Karen Ghavam (LunarG)

Vulkan: Crash Diagnostic Layer
- Jeremy Gebben (LunarG)

Slang in Vulkan
- Hai Nguyen (NVIDIA)

EVOLVE - Next Generation Benchmarking
- Jasper Bekkers and Darius Bouma (Traverse Research)

Adding Vulkan to Pixar’s Hydra Storm Renderer
- Henrik Edstrom (Autodesk), Ashwin Bhat (Autodesk), and Caroline Lachanski (Pixar)
Shading Languages: it’s complicated...

Vulkan is defined to accept shaders in the SPIR-V IR
- In theory, how you generate it is up to you
- But, the ecosystem needs standards and stability
Shading Languages: it’s complicated...

Vulkan is defined to accept shaders in the SPIR-V IR
- In theory, how you generate it is up to you
- But, the ecosystem needs standards and stability

GLSL is NOT going away - will live forever
- But no plan to evolve its syntax (templates, meta-programming, etc)
Shading Languages: it’s complicated...

Vulkan is defined to accept shaders in the SPIR-V IR
- In theory, how you generate it is up to you
- But, the ecosystem needs standards and stability

GLSL is NOT going away - will live forever
- But no plan to evolve its syntax (templates, meta-programming, etc

Some developers will always prefer HLSL
- Microsoft has been very welcoming and accommodating - thanks!
- Resourcing is a problem
Shading Languages: it’s complicated...

Vulkan is defined to accept shaders in the SPIR-V IR
- In theory, how you generate it is up to you
- But, the ecosystem needs standards and stability

GLSL is NOT going away - will live forever
- But no plan to evolve its syntax (templates, meta-programming, etc

Some developers will always prefer HLSL
- Microsoft has been very welcoming and accommodating - thanks!
- Resourcing is a problem

Slang is an option
- NVIDIA has offered to place it under community governance
- Khronos is one possible hosting consortium - under discussion
Vulkan BOF Presentations

Vulkan: Looking back, and looking forward
   - Tom Olson (Vulkan Working Group Chair / Arm)

Vulkan SDK: Where we started, and where we are going
   - Karen Ghavam (LunarG)

Vulkan: Crash Diagnostic Layer
   - Jeremy Gebben (LunarG)

Slang in Vulkan
   - Hai Nguyen (NVIDIA)

EVOLVE - Next Generation Benchmarking
   - Jasper Bekkers and Darius Bouma (Traverse Research)

Adding Vulkan to Pixar’s Hydra Storm Renderer
   - Henrik Edstrom (Autodesk), Ashwin Bhat (Autodesk), and Caroline Lachanski (Pixar)
Vulkan Video is a Thing!

- Vulkan Video expands Vulkan capabilities
  - Accelerated processing of streamed media into the Vulkan pipeline

Vulkan Video is increasingly providing cross-platform media framework acceleration

Status tracked at
https://blogs.igalia.com/vjaquez/vulkan-video-status/
Ray tracing in Mobile is a Thing too!

Vulkan BOF Presentations

Vulkan: Looking back, and looking forward
- Tom Olson (Vulkan Working Group Chair / Arm)

Vulkan SDK: Where we started, and where we are going
- Karen Ghavam (LunarG)

Vulkan: Crash Diagnostic Layer
- Jeremy Gebben (LunarG)

Slang in Vulkan
- Hai Nguyen (NVIDIA)

EVOLVE - Next Generation Benchmarking
- Jasper Bekkers and Darius Bouma (Traverse Research)

Adding Vulkan to Pixar’s Hydra Storm Renderer
- Henrik Edstrom (Autodesk), Ashwin Bhat (Autodesk), and Caroline Lachanski (Pixar)
Professional Rendering

Artwork by Emily Bisset, courtesy of Adobe

Vulkan Ray Tracing in Aurora:
An Open Source Real-Time Path Tracer

https://github.com/Autodesk/Aurora

Substance 3D Stager
Vulkan BOF Presentations

Vulkan: Looking back, and looking forward
- Tom Olson (Vulkan Working Group Chair / Arm)

Vulkan SDK: Where we started, and where we are going
- Karen Ghavam (LunarG)

Vulkan: Crash Diagnostic Layer
- Jeremy Gebben (LunarG)

Slang in Vulkan
- Hai Nguyen (NVIDIA)

EVOLVE - Next Generation Benchmarking
- Jasper Bekkers and Darius Bouma (Traverse Research)

Adding Vulkan to Pixar’s Hydra Storm Renderer
- Henrik Edstrom (Autodesk), Ashwin Bhat (Autodesk), and Caroline Lachanski (Pixar)
What’s New:
Documentation and Developer Support
Vulkan Documentation Project

Bring Vulkan documentation together in one place

- Specification, Vulkan Guide, Proposal documents, Samples...
- Easy navigation and cross-linking
- https://docs.vulkan.org
- Please report issues at https://github.com/KhronosGroup/Vulkan-Site

Resource Creation

Vulkan supports two primary resource types: buffers and images. Resources are views of memory with associated formatting and dimensionality. Buffers provide access to raw arrays of bytes, whereas images can be multidimensional and may have associated metadata.
Vulkan Samples Repository

A home for Vulkan sample code
- Intended to help you learn to use Vulkan effectively
- GPU, OS, and platform neutral, well tested
- On github in open source (Apache 2.0)
- Access via docs.Vulkan.org or at github/KhronosGroup/Vulkan-Samples

A community effort
- Khronos member ISVs, IHVs, contractors
- Interested community members
Some recently added samples

Sparse Image / virtual texture (Mobica)

OIT using per-pixel linked lists (community)

Mobile NeRF (Qualcomm)

Vulkanised!

First full-scale Vulkanised was held in February 2023
- Hosted by Google in Munich, Germany
- Three days of talks, panels, demos, and a Vulkan course
  - All on line at https://vulkan.org/learn#videos

Second in February 2024
- Hosted by Google in Sunnyvale, California
Vulkanised 2025

The 7th Vulkan Conference | Cambridge, UK | Feb 11-13, 2025
The Premier Vulkan Developer Conference

To be hosted by Arm in Cambridge, UK - submissions due Oct. 11
Thanks!
ANARI

Jeff Amstutz
NVIDIA
API Design: Balancing Opposing Forces

**API Uniformity**
- Handle-based Objects
- Generic Parameters + Arrays
- Object/Array Updates
- Scene Hierarchy
- Concurrency + Parallelism
- API Synchronization Semantics
- Graphics/Compute API Interop
- ...

**Feature Differentiation**
- Supported API Extensions
- Performance (Frame/Update Latencies)
- Supported Hardware Features
- Image Quality
- Scene Size (Memory overhead, LoD, Out-of-core, Distributed, etc…)
- …
API Design: Balancing Opposing Forces

API Uniformity
- Handle-based Objects
- Generic Parameters + Arrays
- Object/Array Updates
- Scene Hierarchy
- Concurrency + Parallelism
- API Synchronization Semantics
- Graphics/Compute API Interop
- ...

Feature Differentiation
- API Extensions
- Scene Size (Memory overhead, LoD, Out-of-core, Distributed, etc…)
- Graphics/Compute API Interop
- ...

only "what" and "when"

not "how"
ANARI Development Stack

3D Applications

Scene Graphs

3D Rendering Engines

VisRTX | VisGL | IndeX | OSPRay | Radeon ProRender | Cycles | ...

Graphics + Compute APIs

OptiX | DirectX | Vulkan | OpenGL | Embree | Metal | ...

Hardware

GPUs | CPUs | ...

C99 | C++ | Python | ...

This work is licensed under a Creative Commons Attribution 4.0 International License
Open-source SDK includes Conformance Test code

All specification, SDK and Conformance Test development work done publicly on GitHub
Data Parallel Rendering

Standardized Data-Parallel Rendering Using ANARI

Ingo Wald\textsuperscript{1} NVIDIA
Stefan Zeilmann\textsuperscript{1} University of Cologne
Jefferson Amstutz\textsuperscript{1} NVIDIA
Qi Wu\textsuperscript{1} University of California, Davis
Kevin Griffin\textsuperscript{1} NVIDIA
Milan Jaros\textsuperscript{2} IT4Innovations, VSB – Technical University of Ostrava
Stefan Wesner\textsuperscript{2} University of Cologne

Figure 1: Several examples of large sci-vis data being rendered using the data-parallel ANARI paradigm proposed in this paper. From left to right: a) Roughly one billion color-mapped spheres, rendered using HidStack and BANARI. b) The roughly 50GB exa data set, with volume path tracing on 128 GPUs, also using HidStack and BANARI. c) An iso-surface rendered during an in-situ Ascent session, while attached to an SSD simulation. d) ParaView performing data-parallel rendering on the airp2lass data set, using our data-parallel ANARI integration in preserver.

ABSTRACT
We propose and discuss a paradigm that allows for expressing data-parallel rendering with the classically non-parallel ANARI API. We propose this as a new standard for data-parallel sci-vis rendering, describe two different implementations of this paradigm, and use multiple sample integrations into existing apps to show how easy it is to adopt this paradigm, and what can be gained from doing so.

1 INTRODUCTION
Visualization is about more than rendering, but rendering nevertheless plays a large role in many vis tools. Rendering is hard: it was already a hard problem when all such tools could rely on a single common API (e.g. OpenGL); today it is further complicated involved in rendering, such as cameras or data arrays containing geometry, materials, colors, etc. These objects ultimately represent a generic interface to the private implementation of the back-end, where the mechanics of rendering frames is left up to the implementation.

ANARI is not a silver bullet, though. Even with a single agreed-upon API, different implementations can and will still differ in what features exactly they will support (and in which form). Thus, applications still need to be aware of which specific implementation they may be running on—and either adopt a least common denominator approach, or have some application features only available from specific ANARI vendors. Still, this standardization is encouraging as ANARI is already seeing adoption even in VTK and VTK-m, and some vendors even support ANARI in their commercial products.
Updates Since SIGGRAPH 2023

- **New Adopters Program**
  - Help secure future ANARI SDK development through official conformance!
  - Improved Conformance Test Suite (with more on the way!)

- **Many SDK Improvements + Additions:**
  - Initial version of an OpenUSD Hydra plugin ‘hdAnari’ now available
  - New Blender add-on - contributions welcome!
  - Application debug layer can be enabled without code changes
  - Helium now provides a generic host-side array implementation

- **New Implementations and Integrations coming online**
  - New applications: OVITO, Ascent | Improved integrations: VTK, VTK-m, ParaView, VisIt
  - New devices: Visionaray (CPU/CUDA), Barney (OptiX + MPI parallel), Cycles (prototype),...

Lots more detail at the ‘Exploring ANARI’ BOF
10AM Wednesday
glTF & 3D Commerce

Alexey Medvedev
Meta
glTF - 3D Asset Transmission Format

AUTHORING FORMATS

USD, Material X, FBX, MDL ...

AUTHORING TOOLS

Interchange

Import & Export

Remix / Publish

AUTHORING TOOLS

Transmission

Exchange

VIEWERS & ENGINES

Optimization & Verification

Testing and Editing

Distill

Blender
3DS MAX
Maya
Adobe ...

glTF TOOLS

glTF’s focus is on enabling optimized run-time delivery of 3D Assets
Increasingly Foundational for Other Standards

- **glTF as an ISO standard**: solidifies global recognition and adoption as a 3D asset format.

- **Customized interactive 3D avatar format based on glTF + extensions**: (.vrm extension)

- **Streamlined streaming and rendering large-scale 3D geospatial datasets**: uses glTF + extensions (.b3dm and .i3dm extensions)

- **ISO/IEC 23090-14:2023 (MPEG-I for immersive media experience)** uses glTF + extensions as its scene graph (.mp4 extension)

- **ISO/TS 32007 brings glTF 2.0 as a supported 3D asset into PDF**

- **ISO/IEC IS 19775-1:2023 (X3D)** uses glTF + extensions as its scene graph (.mp4 extension)
Cross Standards Cooperation

Cooperation between glTF and USD ecosystems is a significant industry benefit.

Asset format to enable 3D content to be pervasively delivered and displayed on a wide diversity of native and web viewers, applications and engines.

Extensible framework and ecosystem for describing, composing, simulating, and collaboratively navigating and constructing 3D scenes.

Multiple open-source projects including OpenPBR and MaterialX.

Metaverse Standards BOF
Presentation by glTF/USD Interoperability Working Group
Tuesday 11:30AM Room 710
glTF PBR Materials Roadmap

Incremental consolidation and meticulous specification of proven and accepted industry practice

Subsurface
In development
Khronos glTF PBR available in MaterialX

- glTF’s PBR material is available as a node graph in MaterialX since 2022
  - Being updated for 2024

- Next step: feed MaterialX as a set of procedural texture inputs into glTF PBR
  - Enable much higher detail in smaller assets
  - Remain compatible with existing PBR shaders
  - Optional texture atlas fallbacks for compatibility
  - Extension in development
Khronos PBR Neutral Tone Mapper

- True-to-Life Color Rendering of 3D Products
  - Released in May 2024
  - Specification and sample implementation

- 1:1 match for colors up to a certain maximum value
  - The remainder of color space used as headroom for compressed highlights

- Wide adoption and support by 3D tools and engines
  - <model-viewer>, Autodesk, Babylon.js, Blender, Dassault, Filament
  - London Dynamics, Phasmatic, Three.js, and ThreeKit
glTF Viewer for iOS

- Khronos Releases Open-Source iOS App for Viewing glTF Files
  - Available on the Apple App Store and supports AR mode
  - Source code available on GitHub under the Apache 2.0 license
glTF Spatial Computing Roadmap

Interactivity

Physics

Complex Scenes

Audio
glTF Interactivity Extension

- Uses behavior graphs to add logic and behaviors to glTF assets
  - Interactive assets portable across eCommerce sites, applications, XR experiences etc.
  - Focus on safety, portability and ease of implementation

- Distillation of engine accepted practice
  - Unity (Visual Scripting), Unreal (Blueprints), Nvidia Omniverse (Action Graph)

- Invitation for Public Comments issued!
  - Draft Specification on GitHub | Khronos webinar on interactivity
  - Feedback on GitHub pull request | glTF Interactivity Graph Authoring Tool (WIP DCC React App)
glTF Physics

• Express the physics properties of assets in a platform independent way
  - Provides procedural animation
  - Makes scenes more interesting, believable, and dynamic

• Enables scene understanding
  - Possible with render geometry, but much more efficient with physics

• Rigid Bodies
  - Collision geometry | Rigid bodies
  - Motions | Materials
  - Joints | Filters

Distillation of widely adopted physics engines practices

Specification
feedback welcome!
glTFX : glTF eXternal References

- Meet user requirement to reference multiple glTF assets
  - Complex scenes, Level-of-Detail, streaming, smart loading, scene change...

- Adds a new glTF file type (glTFX)
  - Contains eXternal reference to glTF files
  - New file does not specify any meshes, animations, materials, etc. directly

- Draft spec: [khr.io/127](khr.io/127)
Khronos 3D Commerce

Making 3D Pervasive - in the Real World

Build Once, Use Everywhere

Developing tools and techniques for 3D assets to be reliably and consistently used and displayed across diverse platforms and engines

Multiple Projects Underway

Render Showcase - evolve and expand Render Fidelity Site

Tone Mapping (PBR Neutral), exposure and lighting

Apparel: Skeletal & Facial Anchoring, Virtual Try-On, Stitching / detailing, Simulation

Content Developers test and validate assets

Sample Viewer
Asset Display Exemplar

The equivalent of an API
Conformance Test Suite
(Vulkan has >3M tests)

Asset Library
Showcase & Test Assets

Public comparison of multiple engines
1) Early warning of inconsistencies
2) Incentive to implement functionality

Validator
Asset Correctness

Asset Generator
Generate unit tests

Render Showcase
Web-based display comparisons

Sample images of assets and comparisons are shown on the right.

This work is licensed under a Creative Commons Attribution 4.0 International License
Two glTF BOFs to Come and Learn More!

Innovations in 3D Content Delivery and Use
Tuesday, July 30: 8:30 am - 11:30 AM
Room 710

<table>
<thead>
<tr>
<th>Start</th>
<th>Session</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30</td>
<td>glTF Intro and Roadmap</td>
<td>Alexey Medvedev, Meta</td>
</tr>
<tr>
<td>8:45</td>
<td>Interactivity</td>
<td>Peter Martin , Adobe</td>
</tr>
<tr>
<td>9:00</td>
<td>Physics</td>
<td>Eoin McIoughlin, Microsoft</td>
</tr>
<tr>
<td>9:10</td>
<td>Audio</td>
<td>Chintan Shah, Meta</td>
</tr>
<tr>
<td>9:20</td>
<td>Complex Scenes</td>
<td>Leonard Daly, Daly Realism</td>
</tr>
<tr>
<td>9:30</td>
<td>AEC &amp; Geospatial</td>
<td>Sean Lilly and Adam Morris, Cesium</td>
</tr>
<tr>
<td>9:45</td>
<td>Q&amp;A</td>
<td>All</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start</th>
<th>Title</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td>glTF Intro and Roadmap</td>
<td>Alexey Medvedev, Meta</td>
</tr>
<tr>
<td>1:05</td>
<td>Interactivity</td>
<td>Peter Martin , Adobe</td>
</tr>
<tr>
<td>1:20</td>
<td>Physics</td>
<td>Eoin McIoughlin, Microsoft</td>
</tr>
<tr>
<td>1:30</td>
<td>Audio</td>
<td>Chintan Shah, Meta</td>
</tr>
<tr>
<td>1:40</td>
<td>Complex Scenes</td>
<td>Leonard Daly, Daly Realism</td>
</tr>
<tr>
<td>1:50</td>
<td>AEC &amp; Geospatial</td>
<td>Sean Lilly and Adam Morris, Cesium</td>
</tr>
<tr>
<td>2:05</td>
<td>Q&amp;A</td>
<td>All</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start</th>
<th>Title</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:25</td>
<td>Image Compression</td>
<td>Stephanie Hurlbut &amp; Rich Geldrich, Binomial</td>
</tr>
<tr>
<td>2:40</td>
<td>PBR Materials</td>
<td>Alexey Medvedev, Meta</td>
</tr>
<tr>
<td>2:50</td>
<td>Tone Mapping</td>
<td>Emmett Lalish, Google</td>
</tr>
<tr>
<td>3:00</td>
<td>Content Creation</td>
<td>Eric Chadwick, DGG</td>
</tr>
<tr>
<td>3:25</td>
<td>Q&amp;A</td>
<td>All</td>
</tr>
</tbody>
</table>

Slightly more in Depth  
Part of the Khronos BOF Series
WebGL + WebGPU

Ken Russell
Google
WebGL Update

- Khronos is fully supporting development of WebGPU at W3C
  - Working for a smooth transition for developers between WebGL and WebGPU
  - WebGPU brings GPU Compute to the Web using Vulkan/DX12/Metal backends

- WebGL is pervasive and will be used by many applications for many years
  - Khronos is evolving the WebGL specification and supporting multiple implementations
  - ANGLE's Metal backend supports WebGL 2.0 in Safari on macOS/iOS
    - Shipping in Chrome on Mac/ARM; coming to Mac/Intel
  - Firefox's WebGL implementation is similarly advancing

WebGL 2.0 is available on >96% of browsers
Pixel Local Storage Extension

- Programmable blending and other use cases
  - Developed by Chris Dalton from Rive with significant contributions from Alexey Knyazev

- Specification being finalized
  - In Draft in Chrome Canary
  - Implementation in ANGLE is tracking the spec and can ship soon afterward
  - Specification | Source Code | Live demo implements blend_equation_advanced
  - Enable WebGL draft extensions in about:flags
New WebGL Extensions

- Multiple useful extensions have been added to WebGL
  - Ported from OpenGL ES to WebGL
- These are now shipping in browsers
  - Enhance, speed up, and simplify applications
  - Always test for the presence of the extension and include fallback paths

- **EXT_clip_control**
- **EXT_conservative_depth**
- **EXT_depth_clamp**
- **EXT_polygon_offset_clamp**
- **EXT_render_snorm**
- **EXT_texture_mirror_clamp_to_edge**
- **NV_shader_noperspective_interpolation**
- **OES_sample_variables**
- **OES_shader_multisample_interpolation**
- **WEBGL_blend_func_extended**
- **WEBGL_clip_cull_distance**
- **WEBGL_polygon_mode**
- **WEBGL_render_shared_exponent**
- **WEBGL_stencil_texturing**
WebGPU Updates

- The WebGPU ecosystem is solidifying!
- Firefox Nightly and Safari Technology Preview have WebGPU support today
  - Try your content and ensure it works across browsers!
- WebGPU specification is nearing Candidate Recommendation!
  - Aim to transition to Living Standard afterward
- Lots of ecosystem progress including Three.js’s WebGPU backend

More exciting news in the Khronos WebGL + WebGPU BOF 9AM Wednesday morning!
## Khronos Group Sessions at SIGGRAPH

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Session Type / Title</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon, Jul 29</td>
<td>3:30pm - 4:30pm</td>
<td>Khronos Fast Forward</td>
<td>3D Commerce, ANARI, glTF, OpenXR, Vulkan, WebGL</td>
</tr>
<tr>
<td>Tue, Jul 30</td>
<td>8:30am - 11:30pm</td>
<td>glTF: Innovations in 3D Content Delivery and Use</td>
<td>3D Commerce, glTF</td>
</tr>
<tr>
<td>Wed, Jul 31</td>
<td>9:00am - 9:30am</td>
<td>Advancements in WebGL and WebGPU ...</td>
<td>WebGL &amp; WebGPU</td>
</tr>
<tr>
<td>Wed, Jul 31</td>
<td>10:00am - 11:00am</td>
<td>Exploring ANARI ...</td>
<td>ANARI</td>
</tr>
<tr>
<td>Wed, Jul 31</td>
<td>11:00am - 12:00pm</td>
<td>OpenXR: Transforming the Future of Cross-Platform XR</td>
<td>OpenXR</td>
</tr>
<tr>
<td>Wed, Jul 31</td>
<td>1:00pm - 3:00pm</td>
<td>glTF: Transforming 3D Asset Delivery for Real-Time Graphics</td>
<td>OpenXR</td>
</tr>
<tr>
<td>Wed, Jul 31</td>
<td>3:00pm - 6:00pm</td>
<td>Vulkan, Forging Ahead (including Slang talk)</td>
<td>Vulkan</td>
</tr>
<tr>
<td>Wed, Jul 31</td>
<td>6:00pm - 9:00pm</td>
<td>Social: Khronos Group Networking Reception</td>
<td>All</td>
</tr>
<tr>
<td>Thu, Aug 1</td>
<td>11:50pm - 12:15pm</td>
<td>Siggraph Talk: Neutral Tone Mapping for PBR Color Accuracy</td>
<td>glTF</td>
</tr>
</tbody>
</table>
Thank You! Have a Great Show!

More Information
www.khronos.org
memberservices@khronosgroup.org