## Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 mins</td>
<td>Overview about the AREA and the Interoperability and Standards Program</td>
</tr>
<tr>
<td>10 min</td>
<td>Paul Davies, The Boeing Company</td>
</tr>
<tr>
<td>10 min</td>
<td>Neil Trevett, President Khronos Group</td>
</tr>
<tr>
<td>25 mins</td>
<td>Brent Insko, Khronos Group OpenXR and Intel</td>
</tr>
<tr>
<td>10 mins</td>
<td>Questions from Participants</td>
</tr>
</tbody>
</table>
SPEAKERS

Paul Davies
The Boeing Company

Neil Trevett, NVIDIA
Khronos Group

Brent Insko, Intel
Khronos Open XR WG
The Comprehensive Ecosystem

Enterprises

Becoming aware of the benefits of AR through research, piloting or implementing projects

Providers

Developing services, software and hardware products/technology for the AR ecosystem

Non-commercial

Universities, public research institutes and agencies who deliver research and push the boundaries of AR technology
Interoperability and Standards Program
AREA
Schema of AR
Customer Needs

Build, operate/manage, use multi-vendor solutions without special engineering

Integration with existing IT for enterprise and operational management

Solutions (hardware and software)

Requirements/needs

Use cases (business processes)

Scenarios

Settings

Industries

Automotive
Aerospace
Industrial equipment
Smart cities
Medical

Engineering Development
Factory
Field Operations
User Operation
Sales & marketing

Field service operation
Engineering review
Interactive user manuals
Manufacturing assembly
Inspection

Remote assistance
Step by step instructions
Visual compare (inspection)
Multi-participant collaboration
Product status documentation

Statements of Pain Points/Problems that AR can Address

Bluetooth
Sensors
Audio
Video
RFID

Connectivity
Performance
Experience
Practicality
Safety
Security & Privacy

App creation
Scripting
UI creation

Authoring

Interaction
UX
Tracking
Sensor usage
Analytics
IoT

Apps

API needs
Languages

Toolkits

Formats
Content
Compression
Encryption
Conversion

Data

Business systems
PLM
ERP
SLM

Integration

Graphics
CPU
Tracking

Field of view
Mono/stereo
Resolution
Audio
Touch

Battery
Ergonomics
Wearability
Durability
Environmental

Battery
Ergonomics
Wearability
Durability
Environmental

Statements of Pain Points/Problems that AR can Address
Program Goals

- To increase knowledge about benefits and approaches to achieving interoperability
- To advance the development of standards or other approaches to interoperability through a set of AREA policies, procedures and programs
Tactics

… to target specific barriers to interoperability and/or promote increased awareness and use of existing (and/or development of new) standards for open, interoperable and safe AR solutions by

1. Develop/share thought leadership content
2. Support standards development
3. Review draft specifications
4. Co-organize/conduct events focusing on interoperability
5. Test/build reference implementations
Role of Standards in AR at Boeing

- Future-proofing
- Standardization across internal Boeing teams
- Supportability
- Interoperability
- Avoiding reliance on one particular vendor
Open Standards for Augmented and Virtual Reality

Neil Trevett
Khronos President
NVIDIA VP Developer Ecosystems
ntrevett@nvidia.com | @neilt3d
What is An Open Standard?

An INTEROPERABILITY STANDARD enables two entities to COMMUNICATE
E.g. Software <-> Hardware

Standards Grow Markets
By reducing consumer confusion and increasing capabilities and usability

Standards Reduce Costs
By sharing development between many companies and driving volume

Standards Speed Time to Market
With well-proven testing and interoperability

Standards Do Not Stifle Innovation
Companies can compete on implementation quality, performance, power etc. etc.

True OPEN Standards
Are not controlled by a single company - but by the industry - typically through an SDO
Well defined participation, governance and intellectual property frameworks

Widespread XR will take a constellation of industry standards
Khronos Connects Software to Silicon

Open interoperability standards to enable software to effectively harness the power of multiprocessors and accelerator silicon

>150 Members - 40% US, 30% Europe, 30% Asia

3D graphics, XR, parallel programming, vision acceleration and machine learning

Non-profit, member-driven standards-defining industry consortium

Open to any interested company

All Khronos standards are royalty-free

Well-defined IP Framework protects participant’s intellectual property
Khronos Standards for XR

Create 3D Assets
Download 3D object and scene data

Vision and sensor processing - including neural network inferencing for machine learning

Portable interaction with VR/AR sensor, haptic and display devices

High-performance, low-latency 3D Graphics

3DCommerce™

glTF™

SyCL™

OpenVX™

OpenCL™

OpenGL®

NNEF™

WebGL™

Vulkan®

OpenXR™
OpenXR is used with a 3D API

- High-performance, low-latency 3D rendering and composition*
  - Multiview
  - Context priority
  - Front buffer rendering
  - Tiled rendering (beam racing)
  - Variable rate rendering

- Display, composition and optical correction parameters

- Cross-platform access to XR
  - HMDs and sensors
  - XR application lifecycle
  - Input device discovery and events
  - Sensor tracking and pose calculation
  - Frame timing and display composition
  - Haptics Control

* OpenXR can be used with other 3D APIs such as Direct3D, OpenGL and OpenGL ES
The Metaverse will be the Web!

https://xkcd.com/1367/

In the future, installing things has gotten so fast and painless. Why not skip it entirely and make a phone that has every app "installed" already and just downloads and runs them on the fly?

I felt pretty clever until I realized I'd invented webpages.
Bringing XR to the Web

Native XR Apps

Native 3D Engines

Web XR Apps

Web XR Apps

The Web will Evolve into the Metaverse

Lifting OpenXR functionality into the Web stack

Close cooperation between WebXR and OpenXR

WebGL

Web 3D Engines

Khronos provides the foundation for native and Web-based 3D/XR
OpenXR and Edge Server Applications with 5G

Wireless mobile device with display and sensors

OpenXR APIs hide the 5G round trip from applications

Sensor handling

Generated Augmentations & Scenes

5G

Low latency Sensor Data

MEC (Multi-access Edge Computing) Server
1. Processes sensor data, including machine learning for environmental lighting, occlusion, scene semantics, object reconstruction and UI
2. Generates imagery from 3D models, including stereo, foveal rendering, ray-tracing, optics pre-distortion, varifocal processing

Needed assets loaded to edge server

Location-aware Content Requests

Apps and 3D Assets
glTF 2.0 Scene Description Structure

- **.gltf (JSON)**
  - Node hierarchy, PBR material textures, cameras

- **.bin**
  - Geometry: vertices and indices
  - Animation: key-frames
  - Skins: inverse-bind matrices

- **.png**
- **.jpg**
- **.ktx2**
- Textures

Mandatory Metallic-Roughness Materials

Optional Specular-Glossiness Materials
3D Commerce - Four Initial Areas of Focus

**Asset Creation Guidelines**
For tools and product designers to create assets with consistent data to be used through the 3D Commerce pipeline

**Product Configuration**
Universal product configurability data and guidelines on how to drive consistent product display

**Metadata**
Structured metadata definitions and examples to consistently carry product information through the retail pipeline

**Viewer Validation and Certification**
Test models, reference viewer, display analysis tools and capability specifications to guarantee a consistent and accurate end user experience

**First Goals**
Helping evolve glTF to meet the needs of 3D Commerce
- Next generation PBR
- Advanced metadata

Industry cooperation to urgently develop guidelines and tools to address priority problem areas
How May Standards Help Your Business?

- Seek friction points in your industry caused by lack of agreed communication protocols
  - Between people, companies, products, subsystems, components …
- Find or create a safe space for industry cooperation to define interoperability standards
  - With well defined governance model, IP framework, and proven processes
- Neil Trevett
  - ntrevett@nvidia.com | @neilt3d

Benefits of Khronos membership [www.khronos.org]
SPEAKERS

Paul Davies
The Boeing Company

Neil Trevett, NVIDIA
Khronos Group

Brent Insko, Intel
Khronos Open XR WG
OpenXR in Depth

(in 30 min or less)

Brent E. Insko, PhD
Lead XR Architect at Intel &
OpenXR Working Group Chair
Agenda

- What is OpenXR?
- A Brief History of the Standard
- What are the Problems we are trying to Solve
- OpenXR Timeline of Development
- Brief Overview
- What’s Next?
- Recap
What is OpenXR?

OpenXR is a royalty-free, open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR)—collectively known as XR—platforms and devices.
A Brief History of OpenXR

- Among the first VR hardware available 2016

- Need applications...
  - Each platform provided an SDK to interface with the hardware
  - Each was different from the other
XR Ecosystem Fragmentation

- Increased development time and therefore cost.
- Increased validation overhead and therefore cost.
- Time and resources spent developing one title, impacts developers’ ability to create more titles.
## Major XR Runtimes

<table>
<thead>
<tr>
<th></th>
<th>Virtual Reality</th>
<th>Augmented Reality</th>
<th>Console VR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PC</td>
<td>AIO</td>
<td>Mobile</td>
</tr>
<tr>
<td>Oculus Rift</td>
<td>Facebook</td>
<td>SteamVR</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Mixed Reality</td>
<td>Microsoft</td>
<td>Daydream</td>
<td>Apple</td>
</tr>
<tr>
<td>Oculus Go</td>
<td>Google</td>
<td>GearVR</td>
<td>Google</td>
</tr>
<tr>
<td></td>
<td>Samsung Oculus</td>
<td>Hololens</td>
<td>Sony</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ML1</td>
<td>PlayStation</td>
</tr>
</tbody>
</table>

| Company          | OS support       | ARKit             | ARCore     | PSVR       |
|------------------|------------------|-------------------|------------|
| Facebook         | Linux            |                   |            |
| Valve            | Windows          |                   |            |
| Microsoft        | Windows          |                   |            |
| Facebook         | Android          |                   |            |
| Google           |                  |                   |            |
| Samsung Oculus   |                  |                   |            |
| Microsoft        |                  |                   |            |
| Magic Leap       |                  |                   |            |
| Apple            |                  |                   |            |
| Google           | iPad             |                   |            |
| Sony             |                  |                   |            |
OpenXR

- Recognizing the problem, several companies got together in late 2016 / early 2017 and formed the OpenXR working group in Khronos.
OpenXR - Solving XR Fragmentation
OpenXR - Solving XR Fragmentation

Before OpenXR
XR Market Fragmentation

After OpenXR
Wide interoperability of XR apps and devices
VR Software Stack (Example)

VR APIs
- Application interface
- Driver(HW) interface

VR Runtime
- Server
- Compositor

VR Application
- Game Engine

Browser
- WebXR

VR APIs
- Vulkan
- Direct3D
- OpenGL (+ other system APIs)

VR Hardware
- Windows
- Linux
- Android
OpenXR Architecture

OpenXR does not replace XR Runtime Systems!
It enables any XR Runtime to expose CROSS-VENDOR APIs to access their functionality.

- Current Device State
  - Controller/Peripheral State
  - Normalized Poses
  - Input Events

- Outgoing Requests
  - Pre-distortion image to display Haptics
  - Post-distortion image to display Haptics

- XR Apps and Engines

- OpenXR API

- XR Vendor Runtime System
  - Distortion Correction and Display Output
  - Coordinate System Unification and Prediction

- OpenXR Device Plugin Extension
  - (Optional - Coming soon)

- Vendor-supplied Device Drivers

- XR Devices
The Structure

- Application
  - OpenXR Application Layer
    - Runtime A
      - VR / AR Hardware
    - OpenXR Device Plugin Extension
      - Runtime B
        - VR / AR Hardware
      - VR / AR Hardware
OpenXR Philosophies

1. Enable both VR and AR applications
   The OpenXR standard will unify common VR and AR functionality to streamline software and hardware development for a wide variety of products and platforms.

2. Be future-proof
   While OpenXR 1.0 is focused on enabling the current state-of-the-art, the standard is built around a flexible architecture and extensibility to support rapid innovation in the software and hardware spaces for years to come.

3. Do not try to predict the future of XR technology
   While trying to predict the future details of XR would be foolhardy, OpenXR uses forward-looking API design techniques to enable engineers to easily harness new and emerging technologies.

4. Unify performance-critical concepts in XR application development
   Developers can optimize to a single, predictable, universal target rather than add application complexity to handle a variety of target platforms.
Where are we on the timeline?

<table>
<thead>
<tr>
<th>Call for Participation / Exploratory Group Formation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall F2F, October 2016: Korea</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement of Work / Working Group Formation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter F2F, January 2017: Vancouver</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Defining the MVP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall F2F, September 2017: Chicago</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Public Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDC, March 2018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Public Demonstrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGGRAPH, August 2018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Release Provisional Specification!</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDC, March 2019</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolving Issues / Incorporating feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring F2F, April 2019: Singapore</td>
</tr>
<tr>
<td>Summer F2F, July 2019: Bellevue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Release 1.0 Specification!</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGGRAPH, July 2019</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conformance Tests and Adopters Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feedback</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Finalize Implementations</th>
</tr>
</thead>
</table>

| Enable Conformant Implementations to Ship    |

OpenXR 1.0 Conformance Tests and Adopters Program Development
OpenXR API High Level Concepts
OpenXR Application Flow Diagram
Form Factors

- Currently two XrFormFactors in the specification:
  - XR_FORM_FACTOR_HANDHELD_DISPLAY
  - XR_FORM_FACTOR_HEAD_MOUNTED_DISPLAY

<table>
<thead>
<tr>
<th>Camera Passthrough AR</th>
<th>Stereoscopic VR / AR</th>
<th>Projection CAVE-like</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Camera Passthrough AR" /></td>
<td><img src="image" alt="Stereoscopic VR / AR" /></td>
<td><img src="image" alt="Projection CAVE-like" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>One View</th>
<th>Two View (one per eye)</th>
<th>Twelve Views (six per eye)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XR_FORM_FACTOR_HANDHELD_DISPLAY</td>
<td>XR_FORM_FACTOR_HEAD_MOUNTED_DISPLAY</td>
<td>(future support)</td>
</tr>
</tbody>
</table>

Photo Credit: Dave Pape
View Configurations

- Currently two XrViewConfigurations in the specification:
  - `XR_VIEW_CONFIGURATION_TYPE_PRIMARY_MONO`
  - `XR_VIEW.Configuration_TYPE_PRIMARY_STEREO`

<table>
<thead>
<tr>
<th>Camera Passthrough AR</th>
<th>Stereoscopic VR / AR</th>
<th>Projection CAVE-like</th>
</tr>
</thead>
<tbody>
<tr>
<td>XR_FORM_FACTOR_HANDHELD_DISPLAY</td>
<td>XR_FORM_FACTOR_HEAD_MOUNTED_DISPLAY</td>
<td>(future support)</td>
</tr>
<tr>
<td>XR_VIEW_CONFIGURATION_TYPE_PRIMARY_MONO</td>
<td>XR_VIEW_CONFIGURATION_TYPE_PRIMARY_STEREO</td>
<td>(future support)</td>
</tr>
</tbody>
</table>
Input and Haptics
Input and Haptics

When user clicks button “a” it results in the user teleporting
**Input and Haptics**

Input in OpenXR goes through a layer of abstraction built around Input Actions (XrActions). These allow application developers to define input based on resulting action (e.g. “Grab”, “Jump,” “Teleport”) rather than explicitly binding controls.

While the application can suggest recommended bindings, it is ultimately up to the runtime to bind input sources to actions as it sees fit (application’s recommendation, user settings in the runtime’s UI, etc).
Input and Haptics - Interaction Profiles

- Collections of input and output sources on physical devices
- Runtimes can support multiple interaction profiles

ControllerCorp’s Fancy_Controller:
- /user/hand/left
- /user/hand/right
- /input/a/click
- /input/b/click
- /input/c/click
- /input/d/click
- /input/trigger/click
- /input/trigger/touch
- /input/trigger/value
- /output/haptic
Input and Haptics - Predefined Interaction Profiles

- Interaction profiles for many current products are predefined in the OpenXR specification including:
  - Google Daydream* controller
  - HTC Vive and Vive Pro* controllers
  - Microsoft* Mixed reality motion controllers
  - Microsoft* Xbox controller
  - Oculus Go* controller
  - Oculus Touch* controllers
  - Valve Index* controllers
Input and Haptics - Runtime Binding Decision - Why?

- Runtime ultimately decides the bindings
  - “dev teams are ephermal, games last forever”
  - More likely the runtime is updated than individual games

- Reasons for selecting different bindings:
  - 1. this runtime does not have ControllerCorp’s fancy_controller currently attached, but it knows how to map the inputs and outputs to the controllers that *are* attached
  - 2. Some runtimes can support user mapping of inputs such that the controls per game can be customized by the user, such as swapping trigger and button ‘a’, this enables customization without the original application knowing about it
  - 3. Some future controller is developed but the application is not updated for it, a new interaction profile can help map the actions to the new inputs
  - 4. Accessibility devices can be used and input mapped appropriately
Input and Haptics

XrAction: “Teleport”

OpenXR Runtime

- .../input/trigger/click
  - Explode
- .../input/a/click
  - Teleport

/user/hand/left/input/a/click
(/interaction_profile/ControllerCorp/fancy_controller/input/a/click)
Input and Haptics

Haptics build upon the same XrAction system, and have their own Action Type: XR_HAPTIC_VIBRATION. Just like other XrActions, they can be used with XrActionSets, but unlike inputs, they are activated with xrApplyHapticFeedback.

Currently, only XrHapticVibration is supported:
- Start Time
- Duration (s)
- Frequency (Hz)
- Amplitude (0.0 - 1.0)

xrStopHapticFeedback can also be called to immediately end haptic feedback.

We expect that many more haptic types will be added through extensions as the technology develops.
Deeply Pipelined Multithreaded Example
(Unreal Engine 4 with Vulkan, DX12, Metal)
Extensions

Core Standard

Core concepts that are fundamental to the specification for all use cases

Examples: Instance management, tracking, frame timing

KHR Extensions

Functionality that a large class of runtimes will likely implement

Examples: Platform support, Graphic API Extensions

EXT Extensions

Functionality that a few runtimes might implement

Examples: Performance Settings, Thermals, Debug Utils

Vendor Extensions

Functionality that is limited to a specific vendor

Examples: Device specific functionality
Current KHR Extensions

- **Platform-specific support:**
  - KHR_android_create_instance
  - KHR_android_surface_swapchain
  - KHR_android_thread_settings

- **Graphics API support:**
  - KHR_D3D11_enable
  - KHR_D3D12_enable
  - KHR_opengl_enable
  - KHR_opengl_es_enable
  - KHR_vulkan_enable
  - KHR_vulkan_swapchain_format_list

- **Support for specific XR layer types:**
  - KHR_composition_layer_cube
  - KHR_composition_layer_cylinder
  - KHR_composition_layer_depth
  - KHR_composition_layer_equirect

- **Performance improvement by masking non-visible portions of the display:**
  - KHR_visibility_mask

- **Time Conversion functions:**
  - KHR_convert_timespec_time
  - KHR_win32_convert_performance_counter_time
Eye Tracking Extension

- Foveated rendering
- Gaze selection/feedback/Aiming
- Social interaction
- Headset position guidance
Hand Tracking Extension

- Standardize the definition of a common set of hand joints, e.g. naming, location and orientations.
- Five fingers are named as: thumb, index, middle, ring and little.
- Each finger has 5 joints: tip, distal, intermediate, proximal, metacarpal, except thumb lacks intermediate joint.
- Wrist joint is added as the pivot point of hand motion.
- Palm joint is added as the center of hand and easy access to palm orientation.
- This extension fits both VR and AR experiences.
There are a list of things to consider for 1.1 and for additional extensions

- Many of the items on the list are obvious next steps in the progress of AR and VR development

- Won’t list them here 😊

- But feel free to send us your list of requests in via the feedback channels we’ll provide in a sec
OpenXR 1.0 Release is Here!

The OpenXR standard could lead to loads of cross-platform AR and VR applications. The OpenXR specification standard addresses some of the most difficult challenges of cross-platform AR and VR development.

VR Industry News

OpenXR 1.0 Specification Release Carries Wide Industry Support

An effort to standardize certain aspects of VR and AR applications gains wide industry support today with the release of version 1.0 of the OpenXR specification.

https://www.engadget.com/2019/07/30/openxr-1-launch/
https://uploadvr.com/openxr-specification-standard/
https://pcper.com/2019/07/the-khronos-group-ratifies-and-releases-openxr-1-0/
"OpenXR 1.0 release is a huge milestone and AMD is proud to have been a member in its creation. The expanding XR industry and ecosystem continues to be a key focus for AMD and we are excited by the potential for market growth that OpenXR 1.0 enables. As always, AMD is a proponent of open industry standards," said Daryl Sartain, Director of XR at AMD.

"Arm is focused on developing technology innovations that power the next generation of untethered, standalone AR/VR devices. The release of the OpenXR 1.0 specification will further enable us to break barriers for cross-platform XR applications, while bringing the performance and efficiency required to support these complex, immersive use cases," said Roger Barker, director of IP solutions, Immersive Experience Group, Arm.

"As part of its unwavering commitment to open source and open standards, Collabora is proud to be part of bringing OpenXR 1.0 to life. We are pioneering the Monado open source runtime for OpenXR to ensure the future of XR is truly open and accessible to all hardware vendors. As the OpenXR specification editor, I am grateful for the diligent efforts of the working group, as well as the community feedback that shaped this release," said Ryan Pavlik, OpenXR Specification Editor, XR Principal Software Engineer at Collabora.

"OpenXR 1.0 steers us toward an alignment of many crucial emerging interface platforms. CTRL-labs is excited to contribute to this important step forward and to give developers the tools they need to explore neural interfaces." said Attila Maczak, Software Architect at CTRL-labs.

"We're thrilled to support the OpenXR 1.0 release, along with all of the Khronos Group members who have worked tirelessly to create the standard. Unreal Engine led the way with support for the OpenXR 0.9 provisional specification, and we're excited to move the 1.0 revision forward in collaboration with our hardware partners releasing at the same time. Epic believes that open standards are essential to driving technology and bridging the gaps between digital ecosystems," said Jules Blok, Epic Games.

"Facebook and Oculus continue to believe in the value the OpenXR standard delivers to users and developers. We plan to provide runtime support for apps built on OpenXR 1.0 on the Rift and Quest platforms," said Nate Mitchell, Oculus Co-founder and head of VR product, Facebook.

"WebXR relies on APIs like OpenXR to provide the communication layer between browsers and virtual reality or augmented reality devices. The Immersive Web Working Group is excited for OpenXR support to become widely available because it has the potential to reduce the development and maintenance burden for WebXR implementers while increasing the range of supported hardware," said Brandon Jones, WebXR Spec Editor.

"HTC VIVE is committed to creating a viable ecosystem for the XR industry which is why we are proud to support OpenXR," said Vinay Narayan, vice president, platform strategy, HTC. "Bringing the community together to help define standards and best practices, allows all of us to move forward, together."

"The mobile era of computing was defined and ultimately constrained by closed ecosystems. With mixed reality, the next wave of computing must be and will be open," said Don Box, Technical Fellow at Microsoft. "Today, Microsoft is proud to release the first OpenXR 1.0 runtime that supports mixed reality, for all Windows Mixed Reality and HoloLens 2 users. We are excited to now work with the OpenXR community to design the key extensions that will bring mixed reality to life, with full support by end of year for HoloLens 2 hand tracking, eye tracking, spatial mapping and spatial anchors."

"Congratulations to the OpenXR team on this important 1.0 release. The flexible, extensible design of OpenXR 1.0 will support innovative, next-generation graphics technologies that accelerate XR applications and even enable new XR use-cases," said David Weinstein, Director of Virtual Reality at NVIDIA.

"OpenXR provides a solid foundation for developers to more easily support a broader range of platforms and devices. 1.0 is an exciting milestone and it is just the beginning. This release will open new doors for developers to experiment and extend the capabilities, pushing the VR and AR industry into the future." said Jared Cheshier, CTO at Pluto VR.

"It’s great to see this important milestone finally completed and we are excited by the promise OpenXR holds for lowering the barrier for creating XR applications across a wide array of devices. Now that the stable 1.0 release of OpenXR is out and we are beginning to see industry adoption, Tobii will work diligently to unlock support for eye tracking through OpenXR extensions for eye gaze interaction and foveated rendering." said Denny Rönngren, architect at Tobii.

"Unity is committed to being an open and accessible platform and we remain supportive of open standards for XR applications and devices," said Ralph Hauwert, vice president of platforms at Unity Technologies. "To that end, we're excited about OpenXR and believe this is a significant step towards a more open ecosystem."

"OpenXR is an important milestone for VR. This API will allow games and other applications to work easily across a variety of hardware platforms without proprietary SDKs. Valve is happy to have worked closely with other VR industry leaders to create this open standard, and looks forward to supporting it in SteamVR," said Joe Ludwig, programmer at Valve.

"Varjo is creating the world’s most groundbreaking VR/AR/XR hardware and software by merging the real and digital worlds seamlessly together in human-eye resolution. We’re excited for the release of the OpenXR 1.0 specification because it ensures the enterprise community has compatible, easy access to the best XR technology on the market today while removing barriers to future innovation," said Rémi Arnaud, principal architect at Varjo.
What Resources Are Available?
What Resources Are Available?

200+ page specification
What Resources Are Available?

- 200+ page specification
- Reference Pages
What Resources Are Available?

- 200+ page specification
- Reference Pages
- Overview Guide
What Resources Are Available?

- **https://github.com/KhronosGroup/OpenXR-Docs**

  - Contains the source for generating the specification document and reference pages, scripts to be added soon

- Contains the openxr header files
What Resources Are Available?

- [https://github.com/KhronosGroup/OpenXR-Registry](https://github.com/KhronosGroup/OpenXR-Registry)

- Contains the specification, reference pages, and overview guide

---

**OpenXR-Registry**

The OpenXR-Registry repository contains the OpenXR™ API and Extension Registry, including generated specifications and reference pages, and reference cards. The sources for these documents are found in the separate [https://github.com/KhronosGroup/OpenXR-Docs](https://github.com/KhronosGroup/OpenXR-Docs) repository; this repository is used as a backing store for the web view of the registry at [https://www.khronos.org/registry/OpenXR/](https://www.khronos.org/registry/OpenXR/). Commits to the master branch of OpenXR-Registry will be reflected in the web view.

Interesting files in this repository include:

- `specs/1.0/` - OpenXR 1.0 API specifications and reference pages.
- `specs/0.90/` - OpenXR 0.90 Provisional API specifications and reference pages and API reference card.
- `index.php` - toplevel index page for the web view of [https://www.khronos.org/registry/OpenXR/](https://www.khronos.org/registry/OpenXR/). This relies on PHP include files found elsewhere on [www.khronos.org](http://www.khronos.org) and so is not very useful in isolation.
What Resources Are Available?

- [https://github.com/KhronosGroup/OpenXR-SDK-Source](https://github.com/KhronosGroup/OpenXR-SDK-Source)

- Contains the source for:
  - Loader
  - Some basic API layers
  - Test sample

- For the current best example code, see: src/tests/hello_xr
What Resources Are Available?

- [https://github.com/KhronosGroup/OpenXR-SDK](https://github.com/KhronosGroup/OpenXR-SDK)

- Contains Generated Files

- Use for building on Windows and Linux

- Embed this in your projects
Primary Resources

- **OpenXR Landing Page - Specification, Reference Pages, Sample Code, Overview**
  - [https://www.khronos.org/openxr](https://www.khronos.org/openxr)

- **OpenXR Forum and Slack Channel**
  - Forum: [https://khr.io/openxrfedback](https://khr.io/openxrfedback)
  - Discussion: [https://khr.io/slack](https://khr.io/slack)

- **Vendor preview implementations and integration**
  - Collabora: open source implementation [http://monado.dev](http://monado.dev)
  - Microsoft: OpenXR runtime for Windows Mixed Reality headsets [https://aka.ms/openxr](https://aka.ms/openxr)
  - Epic Games: Unreal Engine 4.24 with OpenXR 1.0 plugin [https://www.unrealengine.com](https://www.unrealengine.com)

- **Khronos SIGGRAPH Sessions - including OpenXR Presentation and demos**
  - [https://www.khronos.org/events/2019-siggraph](https://www.khronos.org/events/2019-siggraph)
Additional Resources

- OpenXR Software Development Kit (SDK) Project
- OpenXR Software Development Kit (SDK) Sources Project
- OpenXR API Documentation Project
- OpenXR sample code for simple VR application
- OpenXR Registry (links to specification, documentation, reference guide and more...)
- Microsoft: Getting started with OpenXR
- Microsoft: OpenXR Samples for Mixed Reality Developers
- Oculus: OpenXR Mobile SDK
- Monado: A free, open-source XR platform
- Microsoft: OpenXR app best practices
- Collabora: What's new in OpenXR 1.0 & Monado?
- Collabora: OpenXR Masterclass from Laval-Virtual 2020
- OpenXR Youtube Playlist
- Varjo: OpenXR Developer Site
- Video: A Look at OpenXR - SIGGRAPH 2019 BOF Sessions
- Video: OpenXR - State of the Union - Khronos GDC 2019
What’s Next?
### What’s Next?

<table>
<thead>
<tr>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call for Participation / Exploratory Group Formation</td>
<td>Fall F2F, October 2016: Korea</td>
</tr>
<tr>
<td>Statement of Work / Working Group Formation</td>
<td>Winter F2F, January 2017: Vancouver</td>
</tr>
<tr>
<td>Defining the MVP</td>
<td>Fall F2F, September 2017: Chicago</td>
</tr>
<tr>
<td>First Public Information</td>
<td>GDC, March 2018</td>
</tr>
<tr>
<td>First Public Demonstrations</td>
<td>SIGGRAPH, August 2018</td>
</tr>
<tr>
<td>Release Provisional Specification!</td>
<td>GDC, March 2019</td>
</tr>
<tr>
<td>Resolving Issues / Incorporating feedback</td>
<td>Spring F2F, April 2019: Singapore</td>
</tr>
<tr>
<td>Release 1.0 Specification!</td>
<td>SIGGRAPH, July 2019</td>
</tr>
<tr>
<td>Conformance Tests and Adopters Program</td>
<td>Feedback to Finalize Implementations</td>
</tr>
<tr>
<td>Enable Conformant Implementations to Ship</td>
<td></td>
</tr>
</tbody>
</table>

**Finish Conformance Suite and Release Conformant Implementations**
Open Source Adoption

Chromium - browser project

- Google Chrome and Microsoft Edge support WebXR through OpenXR by default

Blender - 3D creation suite

- Upcoming 2.83 release will include native VR support through OpenXR
Thanks!

- To these companies for enabling their engineers to dedicate time to OpenXR!
Thanks to the Engineers!

Recap

- What is OpenXR?
- A Brief History of the Standard
- What are the Problems we are trying to Solve
- OpenXR Timeline of Development
- Brief Overview
- What’s Next?
- Recap
OpenXR Win-Win-Win

XR Vendors
Can bring more applications onto their platform by leveraging the OpenXR content ecosystem

XR ISVs
Can easily ship on more platforms for increased market reach

XR End-Users
Can run the apps they want on their system - reducing market confusion and increasing consumer confidence
Khronos APIs for XR

High-performance, low-latency 3D rendering and composition*
- Multiview
- Context priority
- Front buffer rendering
- Tiled rendering (beam racing)
- Variable rate rendering

Display, composition and optical correction parameters

Cross-platform access to XR
- HMDs and sensors
- XR application lifecycle
- Frame timing and display composition
- Sensor tracking and pose calculation
- Input device discovery and events
- Haptics Control

* OpenXR can be used with other 3D APIs such as Direct3D, OpenGL and OpenGL ES
Before we go...
Before we go... as always give us Feedback!

Tell us:

- What should be in the spec

- What shouldn’t be in the spec

- How things need to be added for your application/runtime/hardware/OS/...
Join Khronos!

- Get more involved
- Have direct impact on the direction of the API
- Be part of the effort to deliver OpenXR 1.1!
Thank You!
SPEAKERS

Paul Davies
The Boeing Company

Neil Trevett, NVIDIA
Khradnos Group

Brent Insko, Intel
Khradnos Open XR WG