OpenXR fast forward SIGGRAPH 2022

Denny Rönngren
OpenXR: On an HMD near you now!

Denny Rönngren
XR developer at Tobii AB & OpenXR outreach officer, Khronos
Email: denny.ronngren@tobii.com
What is OpenXR?

OpenXR is a royalty-free, open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR) platforms and devices.
Widespread Industry Support

OpenXR is a collaborative design that integrates many lessons from proprietary ‘first-generation’ XR APIs to create a new generation API with cutting-edge capabilities and a flexible, extensible, future-proof architecture.
Widespread industry adoption

- **Microsoft**: HoloLens and Mixed Reality Headsets, includes hand and eye tracking extensions
- **Microsoft**: Rift S, Quest and Quest 2 deprecated own APIs in favor of OpenXR
- **Meta**: Vive Focus 3, Vive Cosmos, Vive Wave Runtime
- **Valve**: Depreciated OpenVR APIs in favor of OpenXR
- **Varjo**: All Varjo Headsets are fully conformant
- **Collabora**: open-source implementation fully conformant from version 21.0.0
- **Snapdragon**: Snapdragon Spaces XR Developer Platform
- **Magic Leap**: End of year → prototype Fully conformant → early ‘23
Game Engine Adoption

- Unity Engine OpenXR Plugin in 2020.2+
- OpenXR added to core 4.0 Alpha 4, March 2022
OpenXR Applications

- Microsoft Flight Simulator
- Minecraft
- Zombieland VR: Headshot Fever
- War Thunder
- OpenBrush
- Echo VR (mobile)
- Horizon Workrooms
- Cubism
- Blender (VR Scene inspection)
- WebXR backend in Chromium
Current Extensions
(+28 since last year)

XR_KHR_android_create_instance
XR_KHR_android_surface_swapchain
XR_KHR_android_thread_settings
XR_KHR_binding_modification
XR_KHR_composition_layer_color_scale_bias
XR_KHR_composition_layer_cube
XR_KHR_composition_layer_cylinder
XR_KHR_composition_layer_depth
XR_KHR_composition_layer_equirect
XR_KHR_composition_layer_equirect2
XR_KHR_convert_timespec_time
XR_KHR_D3D11_enable
XR_KHR_D3D12_enable
XR_KHR_loader_init
XR_KHR_loader_init_android
XR_KHR_opengl_enable
XR_KHR_opengl_es_enable
XR_KHR_swapchain_usage_input_attachment_bit
XR_KHR_visibility_mask
XR_KHR_vulkan_enable
XR_KHR_vulkan_enable2
XR_KHR_vulkan_swapchain_format_list
XR_KHR_win32_convert_performance_counter_time

XR_EXT_conformance_automation (new)
XR_EXT_debug_utils
XR_EXT_dpad_binding (new)
XR_EXT_eye_gaze_interaction
XR_EXT_hand_joints_motion_range
XR_EXT_hand_tracking
XR_EXT_hpmixed_reality_controller
XR_EXT_palm_pose (new)
XR_EXT_performance_settings
XR_EXT_samsung_odyssey_controller
XR_EXT_temporal_query
XR_EXT_vulkan_enable2
XR_KHR_FB_swapchain_update_state
XR_KHR_FB_swapchain_update_state_android_surface
XR_KHR_FB_swapchain_update_state_opengl_es
XR_KHR_FB_swapchain_update_state_vulkan
XR_KHR_triangle_mesh (new)
XR_KHR_triangle_mesh2 (new)
XR_KHR_visibility_mask
XR_KHR_vulkan_enable
XR_KHR_vulkan_swapchain_format_list
XR_KHR_win32_appcontainer_compatible
XR_AMD_digital_lens_control (new)
XR_AMD_view_configuration_fov
XR_FB_android_surface_swapchain_create
XR_FB_color_space
XR_FB_composition_layer_alpha_blend
XR_FB_composition_layer_image_layout
XR_FB_composition_layer_secure_content
XR_FB_composition_layer_settings (new)
XR_FB_display_refresh_rate
XR_FB_foveation
XR_FB_foveation_configuration
XR_FB_foveation_vulkan
XR_FB_hand_tracking_aim (new)
XR_FB_hand_tracking_capsules (new)
XR_FB_hand_tracking_mesh (new)
XR_FB_keyboard_tracking (new)
XR_FB_passthrough (new)
XR_FB_passthrough_keyboard_hands (new)
XR_FB_render_model (new)
XR_FB_scene (new)
XR_FB_space_warp (new)
XR_FB_swapchain_image_storage
XR_FB_swapchain_image_storage_list
XR_FB_swapchain_image_storage_list2
XR_FB_swapchain_image_storage_list3
XR_KHR_XML_vulkan_image_storage
XR_KHR_XML_vulkan_image_storage2
XR_KHR_XML_vulkan_image_storage3
XR_KHR_XML_vulkan_image_storage4
XR_KHR_XML_vulkan_image_storage5
Advanced UI Cross-Vendor OpenXR Extensions

Developers can build cross-platform applications that use advanced UI solutions from different technology vendors. OpenXR API layers can be used to implement extensions.

**Hand Tracking**

- 26 unique joints per hand for fully articulated hands visible to the user

*Ultraleap API layer* available

**Eye Tracking**

- Eye gaze interaction for intuitive interfaces
- 2-Step Interaction
- Hand-eye coordination
- Natural aiming

Ultraleap

Microsoft

Meta

HTC

Qualcomm

Tobii

Microsoft
Multimodality (cross-vendor extensions)

Hand tracking *and* hand controller
- Customer feedback on challenges
  - Hand controller and hands used simultaneously
  - Use hands only in app designed for hand controllers
- Future exploration in the working group

Hand tracking *and* eye tracking
- Microsoft released Toolkit for Unity that helps with designing UI in multimodal way
  - Gaze and pinch (eye + hand)
  - Leveraging the OpenXR extensions!

MRTK3 transition to OpenXR - support multiple runtimes / HMDs
What is in the future for OpenXR?

- Clarifying and fixing a few things from OpenXR 1.0
- Expanding support for AR and MR
  - Scene understanding
- Additional User input/output capabilities
  - Expanded haptics support
  - Expanded body tracking support
  - Expanded controller and device support
  - Multimodality
- Extending the existing conformance suite and providing more sample code
Additional resources

- Collabora: OpenXR Masterclass from Laval-Virtual 2020
- Collabora: OpenXR easy-to-read example for OpenGL/Linux
- Holochip: OpenXR Samples
- HTC: OpenXR
- Microsoft: Getting started with OpenXR
- Microsoft: OpenXR app best practices
- Microsoft: OpenXR performance
- Microsoft: OpenXR troubleshooting
- Microsoft: Getting started with OpenXR
- Microsoft: OpenXR Samples for Mixed Reality Developers
- Ultraleap: OpenXR Hand Tracking in Unity
- Oculus: OpenXR for Oculus
- Unity: OpenXR Plugin
- Unreal: Developing with OpenXR