**Khronos Group Releases Vulkan Ray Tracing**

**Set of provisional extension specifications publicly available today for industry feedback.**

**Beaverton, OR – March 17, 2020 – 6:00 AM PT –** Today, The Khronos® Group, an open consortium of industry-leading companies creating advanced interoperability standards, announces the ratification and public release of the Vulkan® Ray Tracing provisional extensions, creating the industry’s first open, cross-vendor, cross-platform standard for ray tracing acceleration. Primarily focused on meeting desktop market demand for both real-time and offline rendering, the release of Vulkan Ray Tracing as provisional extensions enables the developer community to provide feedback before the specifications are finalized. Comments and feedback will be collected through the [Vulkan GitHub Issues Tracker](https://khr.io/vkrayprovfeedback) and [Khronos Developer Slack](https://khr.io/slack). Developers are also encouraged to share comments with their preferred hardware vendors. The specifications are available today on the [Vulkan Registry](https://www.khronos.org/registry/vulkan/).

Ray tracing is a rendering technique that realistically simulates how light rays intersect and interact with scene geometry, materials, and light sources to generate photorealistic imagery. It is widely used for film and other production rendering and is beginning to be practical for real-time applications and games. Vulkan Ray Tracing seamlessly integrates a coherent ray tracing framework into the Vulkan API, enabling a flexible merging of rasterization and ray tracing acceleration. Vulkan Ray Tracing is designed to be hardware agnostic and so can be accelerated on both existing GPU compute and dedicated ray tracing cores if available.

“There has been strong developer demand for a truly cross-platform ray tracing acceleration API and now Vulkan Ray Tracing is here to meet that industry need,” said **Daniel Koch, senior graphics system software engineer at NVIDIA and Vulkan Ray Tracing task sub group chair at Khronos**. “The overall architecture of Vulkan Ray Tracing will be familiar to users of existing proprietary ray tracing APIs, which enables straightforward porting of existing ray traced content, but this framework also introduces new functionality and implementation flexibility.”

Vulkan Ray Tracing consists of a number of Vulkan, SPIR-V, and GLSL extensions, some of which are optional. The primary [VK\_KHR\_ray\_tracing](https://www.khronos.org/registry/vulkan/specs/1.2-extensions/html/vkspec.html#VK_KHR_ray_tracing) extension provides support for acceleration structure building and management, ray tracing shader stages and pipelines, and ray query intrinsics for all shader stages. [VK\_KHR\_pipeline\_library](https://www.khronos.org/registry/vulkan/specs/1.2-extensions/html/vkspec.html#VK_KHR_pipeline_library) provides the ability to provide a set of shaders which can be efficiently linked into ray tracing pipelines. [VK\_KHR\_deferred\_host\_operations](https://www.khronos.org/registry/vulkan/specs/1.2-extensions/html/vkspec.html#VK_KHR_deferred_host_operations) enables intensive driver operations, including ray tracing pipeline compilation or CPU-based acceleration structure construction to be offloaded to application-managed CPU thread pools.

Vulkan Ray Tracing shaders are SPIR-V binaries which use two new extensions. The [SPV\_KHR\_ray\_tracing](http://htmlpreview.github.io/?https://github.com/KhronosGroup/SPIRV-Registry/blob/master/extensions/KHR/SPV_KHR_ray_tracing.html) SPIR-V extension adds support for ray tracing shader stages and instructions; [SPV\_KHR\_ray\_query](http://htmlpreview.github.io/?https://github.com/KhronosGroup/SPIRV-Registry/blob/master/extensions/KHR/SPV_KHR_ray_query.html) adds support for ray query shader instructions. Developers can generate those binaries in GLSL using two new GLSL extensions, [GLSL\_EXT\_ray\_tracing](https://github.com/KhronosGroup/GLSL/blob/master/extensions/ext/GLSL_EXT_ray_tracing.txt) and [GLSL\_EXT\_ray\_query](https://github.com/KhronosGroup/GLSL/blob/master/extensions/ext/GLSL_EXT_ray_query.txt), which are supported in the open source [glslang](https://github.com/KhronosGroup/glslang/) compiler. Engineers at Khronos member companies, including NVIDIA, have also [added support for the SPIR-V](https://devblogs.nvidia.com/bringing-hlsl-ray-tracing-to-vulkan/) extensions to [DXC](https://github.com/microsoft/DirectXShaderCompiler), Microsoft's open source HLSL compiler, enabling Vulkan Ray Tracing SPIR-V shaders to be authored in HLSL using the syntax defined by Microsoft, with minimal modifications.

Driver release updates and the status of Vulkan ecosystem components will be posted on the [Vulkan Ray Tracing Provisional Release Tracker](https://khr.io/vkrayprovrelease). A Vulkan SDK that includes support for Vulkan Ray Tracing will become available once all the necessary ecosystem components are upstreamed; check [this link](https://vulkan.lunarg.com/) to watch for its availability. An introductory launch presentation on Vulkan Ray Tracing is here, and further technical details can be found in this [blog post](https://khr.io/vkrayprovblog).

**Industry Support for Vulkan Ray Tracing Provisional Specification**

"Standardizing ray tracing in Vulkan is an important step towards making ray tracing available across a wide range of devices, as well as enabling developers to use this technology to its full advantage. AMD intends to provide support for all of the major features in this extension, including ray shading, ray queries, and CPU acceleration structure management. We will be working with developers to ensure great performance from our Vulkan Ray Tracing implementation; these efforts will help us to provide end users with even more visually stunning graphics on AMD Radeon™ GPUs,” said **Andrej Zdravkovic, senior vice president, software development, AMD.**

“EA is happy to see the release of the provisional ray tracing extension for Vulkan. Realtime ray tracing is already an important part of game development and it will continue to be in the future. Allowing ray queries from any shader stage is a great feature, which will both simplify integrations and open up the possibility for new techniques while multithreaded host-side building of acceleration structures has the potential to reduce latency and improve the performance of our upcoming game titles,” said **Sebastian Tafuri, senior rendering engineer at Frostbite, EA.**

“Epic Games has been an active member of the Vulkan Ray Tracing group from the beginning, and we are happy to see the ray tracing extension released to the public. We at Epic Games continue to wholeheartedly support Khronos's efforts on creating open standards to enhance the end-user experience,” said **Yuriy O’Donnell, rendering engineer, Epic Games.**

"Imagination Technologies are very happy to see ray tracing become a standard part of Vulkan, helping the overall ray tracing ecosystem to grow substantially due to Vulkan's wide reach across many platforms and devices," said **Rys Sommefeldt, senior director of product, Ray Tracing and High Performance Graphics, Imagination Technologies.** "We are very supportive of the standard, which will help us deliver the efficient, fast, and focused hardware solution we are developing for our customers."

"The Intel Xe architecture roadmap includes support for hardware accelerated ray tracing, and we're excited to work with Khronos to implement full support into Vulkan," said **Joshua Barczak, graphics software architect at Intel.**

“NVIDIA ships beta drivers today with support for the provisional standardized ray tracing functionality in Vulkan,” said **Morgan McGuire, research director at NVIDIA.** “Bringing accelerated ray tracing to the Vulkan cross-platform, open standard API is another significant step towards enabling the highest quality of visual realism for real-time games and applications everywhere.”

"We are very excited about having hardware ray tracing support baked into Vulkan. Vulkan Ray Tracing enables us to research high-end rendering solutions, while also having support for all supporting vendors and platforms with minimal overhead,” said **Jules Urbach, CEO, OTOY.**

**About Vulkan**

Vulkan is an open, royalty-free API for high-efficiency, cross-platform access to modern GPUs, with widespread adoption in leading engines, cutting-edge games, and demanding applications. Vulkan is supported in a diverse range of devices from Windows and Linux PCs, consoles, and the cloud, to mobile phones and embedded platforms.

**About The Khronos Group**

The Khronos Group is an open, non-profit, member-driven consortium of over 150 industry-leading companies creating advanced, royalty-free, interoperability standards for 3D graphics, augmented and virtual reality, parallel programming, vision acceleration and machine learning. Khronos activities include Vulkan®, OpenGL®, OpenGL® ES, WebGL™, SPIR-V™, OpenCL™, SYCL™, OpenVX™, NNEF™, OpenXR™, 3D Commerce™, ANARI™, and glTF™. Khronos members drive the development and evolution of Khronos specifications and are able to accelerate the delivery of cutting-edge platforms and applications through early access to specification drafts and conformance tests.

###

Khronos® and Vulkan® are registered trademarks, and SPIR-V™ is a trademark of The Khronos Group Inc. OpenXR™ is a trademark owned by The Khronos Group Inc. and is registered as a trademark in China, the European Union, Japan and the United Kingdom. OpenCL™ is a trademark of Apple Inc. and OpenGL® is a registered trademark and the OpenGL ES™ and OpenGL SC™ logos are trademarks of Hewlett Packard Enterprise used under license by Khronos. All other product names, trademarks, and/or company names are used solely for identification and belong to their respective owners.

Khronos Group Press Contact:

Alex Crabb, Caster Communications Inc.

alex@castercomm.com

+1-401-792-7080