Standards update and liaison report
January 2019

Original Slides are written by Neil Trevett | Khronos President
Modified and Presented by Hwanyong Lee | Khronos Liaison Representative

www.khronos.org
Over 140 members worldwide
Any company is welcome to join
Topics

• OpenXR first public demonstrations
  - StarVR and Microsoft Windows Mixed Reality headsets

• NNEF 1.0 released at SIGGRAPH
  - Neural Network Exchange Format for machine learning

• Khronos Educators Program launch
  - Shared creation and refinement of course materials

• 3D API ecosystem progress
  - Vulkan 1.1, OpenGL 4.6, OpenGL ES 3.2, WebGL 2.0
  - Porting Vulkan apps to closed platforms

• glTF Widespread Industry Adoption
  - Working on Texture Transmission extension

• Liaison Report
OpenXR - Solving VR Fragmentation

Before OpenXR
VR Market
Fragmentation

After OpenXR
Wide interoperability of
VR apps and devices
Companies Publicly Supporting OpenXR

OpenXR is a collaborative design
1) For cross-platform XR portability - VR in V1.0, then add AR
2) Integrating many lessons from proprietary ‘first-generation’ API designs
OpenXR Development Process

Call for Participation / Exploratory Group Formation
Fall F2F, October 2016: Korea

Statement of Work / Working Group Formation
Winter F2F, January 2017: Vancouver

Specification Work
Spring F2F, April 2017: Amsterdam
Interim F2F, July 2017: Washington

Defining the MVP
Fall F2F, September 2017: Chicago

Resolving Implementation Issues
Winterim F2F, November 2017: Washington
Winter F2F, January 2018: Taipei

First Public Information
GDC, March 2018

First Public Demonstrations!
SIGGRAPH, August 2018

Release Provisional Specification
Conformance Tests and Adopters Program
Finalize Implementations
Ratify and release Final Specification and Enable Conformant Implementations to Ship

Present Day
Coming Soon

Implementations Underway!
Specifications will incorporate implementation experience

Much more detailed specification overview and GDC session videos:
https://www.khronos.org/developers/library/2018-gdc

Feedback

© Copyright Khronos™ Group 2018 - Page 6
Input and Haptics

- Input uses abstracted Input Actions
  - E.g. “Move,” “Jump,” “Teleport”
- Many advantages
  - Existing content can easily use new devices
  - Mix-and-match multiple input sources to create a unified UI
  - Easy optional feature support (e.g. eye and body tracking)
  - Future-proofing for innovation in input devices and form factors
OpenXR Viewport Configurations

- Applications can:
  - Query for runtime supported Viewport Configurations
  - Applications can then set the Viewport Configurations that they plan to use
  - Select and change their active configuration over the lifetime of the session

<table>
<thead>
<tr>
<th>Camera Passthrough AR</th>
<th>Stereoscopic VR / AR</th>
<th>Projection CAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Image of Camera Passthrough AR]</td>
<td>[Image of Stereoscopic VR / AR]</td>
<td>[Image of Projection CAVE]</td>
</tr>
</tbody>
</table>

- One Viewport
  - /viewport_configuration/ar_mono/magic_window
- Two Viewports (one per eye)
  - /viewport_configuration/vr/hmd
- Twelve Viewports (six per eye)
  - /viewport_configuration/vr_cube/cave_vr

Photo Credit: Dave Pape
Layered XR Ecosystems

Native XR Apps
- Vuforia
- Unity
- Unreal Engine
- 3D Engines
- System-exposed AR Capabilities

Web XR Apps
- WebXR
- ARCore
- Future versions of OpenXR will include cross-platform extended AR functionality
- Three.js
- 3D Engines
- Close ongoing collaboration between WebXR and OpenXR

WebGL
- Khronos providing the foundation for 3D and XR in the Web

Vision and Inferencing libraries and run-times
Neural Network Workflow

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Framework Specific Formats

Compilation and Optimization

Trained Networks

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices

Applications Using Embedded Vision and Inferencing

WinML
Neural Networks API

Vision and Neural Net Inferencing Runtimes

Diverse Inferencing Acceleration Hardware

Desktop and Cloud Hardware

Neural Net Training Frameworks

Training = Desktop / Cloud

Deployment on Embedded Devices
NNEF - Solving Neural Net Fragmentation

Before NNEF - NN Training and Inferencing Fragmentation

NN Authoring Framework 1\n\nNN Authoring Framework 2\n\nNN Authoring Framework 3\n\nInference Engine 1\n\nInference Engine 2\n\nInference Engine 3

Every Tool Needs an Exporter to Every Accelerator

With NNEF - NN Training and Inferencing Interoperability

NNEF

Optimization and processing tools

NN Authoring Framework 1\n\nNN Authoring Framework 2\n\nNN Authoring Framework 3

Inference Engine 1\n\nInference Engine 2\n\nInference Engine 3
NNEF Ecosystem

NNEF = Neural Network Exchange Format

NNEF V1.0 released at SIGGRAPH!!
After positive industry feedback on Provisional specification released in December 2017

Comparing Neural Network Exchange Industry Initiatives

<table>
<thead>
<tr>
<th>NNEF</th>
<th>ONNX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined Specification</td>
<td>Open Source Project</td>
</tr>
<tr>
<td>Stability for hardware deployment</td>
<td>Software stack flexibility</td>
</tr>
<tr>
<td>Multi-company Governance</td>
<td>Initiated by Facebook</td>
</tr>
<tr>
<td>Flexible Precision</td>
<td>32-bit Floating Point only</td>
</tr>
<tr>
<td>Flat and Compound Ops</td>
<td>Flat Ops Only</td>
</tr>
</tbody>
</table>

NNEF open source projects hosted on Khronos NNEF GitHub repository
Apache 2.0 license
https://github.com/KhronosGroup/NNEF-Tools

Live

Imminent

TensorFlow and Caffe Exporters

Syntax Parser/Validator

Google NNAPI Converter

OpenVX Ingestion & Execution

NNEF Files

TensorFlow and Caffe2 Importer / Exporters
Vulkan and New Generation GPU APIs

Non-proprietary, royalty-free open standard ‘By the industry for the industry’
Portable across multiple platforms - desktop and mobile
Modern architecture | Low overhead | Multi-thread friendly
EXPLICIT GPU access for EFFICIENT, LOW-LATENCY, PREDICTABLE performance

Vulkan Porting Tools

Vulkan is available on Android 7.0+
Pervasive Vulkan

Major GPU Companies supporting Vulkan for Desktop and Mobile Platforms

http://vulkan.gpuinfo.org/

Platforms

Desktop

Mobile (Android 7.0+)

Media Players

Consoles

Virtual Reality

Cloud Services

Embedded

Game Engines

Valve

id

CryEngine

Unity

EPIC Games

Croteam

Xenko
Vulkan 1.1 Launch and Ongoing Momentum

Strengthening the Ecosystem
- Improved developer tools (SDK, validation/debug layers)
- More rigorous conformance testing
- Shader toolchain improvements (size, speed, robustness)
- Shading language flexibility - HLSL and OpenCL C support
- Vulkan Public Ecosystem Forum

Building Vulkan’s Future
- Deliver complete ecosystem - not just specs
- Listen and prioritize developer needs
- Drive GPU technology

Released Vulkan 1.1 Extensions
- KHR_draw_indirect_count
  - Source draw count parameter from a buffer in GPU-writable memory for greater flexibility for GPU-generated work
- KHR_8bit_storage
  - 8-bit types in uniform and storage buffers for improved compute support in apps such as inferencing and vision
- EXT_descriptor_indexing
  - Dynamically non-uniform (aka bindless) resource access
    - Required by some modern game engine architectures

Discussions in Flight
- Reduced precision arithmetic types
  - FP16 and int8 arithmetic for reduced power and improved performance
- Detailed driver property queries
  - Query vendor (e.g. IHV vs open source), conformance status
- Variable-resolution rendering
  - E.g. foveated rendering for VR / AR
  - ...and many others under investigation
- Perf counter access, memory management, depth/stencil resolve, ray tracing, video, new sync primitives...

Vulkan 1.0 Extensions
- Maintenance updates plus additional functionality
- Explicit Building Blocks for VR: e.g. multiview
- Explicit Building Blocks for Homogeneous Multi-GPU
- Enhanced Windows System Integration
- Increased Shader Flexibility:
  - 16 bit storage, Variable Pointers
  - Enhanced Cross-Process and Cross-API Sharing

Vulkan 1.1
- Integration of 1.0 Extensions.
- New Technology into Core e.g. Subgroup Operations

Widening Platform Support
- Pervasive GPU vendor native driver availability
- Open source drivers - ANV (Intel) and RADV and AMDVLK (AMD)
- Port Vulkan apps to macOS/iOS and DX12

February 2016
Vulkan 1.0
Explicit Access to GPU Acceleration

March 2018
Vulkan 1.1
Integration of 1.0 Extensions.
New Technology into Core e.g. Subgroup Operations
Content is shipping on desktop...

Vulkan-only AAA Titles on PC

Dota 2 on PC and macOS

AAA titles on Linux
...and Mobile

Plus....
Lineage 2 Revolution
Heroes of Incredible Tales
Dream League Soccer...
Vulkan Developer Activity - SDK and GitHub

LunarG Vulkan SDK
Download rate increases every year since launch
http://vulkan.lunarg.com

SIGGRAPH 2016
SIGGRAPH 2017
SIGGRAPH 2018

Vulkan Related GitHub Repos
Vulkan Portability Initiative

Enabling and accelerating the creation of tools and run-time libraries for Vulkan applications to run on platforms supporting only Metal or Direct3D

**Porting Research**
Determining what % of Vulkan can be EFFICIENTLY supported, with high-performance, over various versions of D3D and Metal

**Vulkan Portability Extension**
Enabling application to query what Vulkan features are not supported by a particular library/platform combination

**Portability Layers**
- **DevSim Layer** - develop and debug with the features of a Portability Library on a full Vulkan driver
- **Validation Layer** - enforces use of Portability features

**Conformance Tests**
Enabling Vulkan CTS to test only the available functionality
Subsets cannot be conformant but functionality that is present must work!

Implementation and testing experience

Very little functionality not supported today:
- Triangle fans, Separate stencil reference masks
- Vulkan Events, Allocation callbacks
- Some texture-specific swizzles
Bringing Vulkan Apps to Apple Platforms Today

Dota 2 running on Mac up to 50% faster than native OpenGL

First productions apps using MoltenVK already shipping on macOS and iOS

Open source SDK to build, run, and debug applications on macOS including validation layer support

Beta release - but working to pass all applicable conformance tests

Previously a paid product
Now released into OPEN SOURCE
Completely free to use - no fees or royalties - including commercial applications

SPIRV-Cross
Convert SPIR-V shaders to platform source formats

macOS / iOS Run-time
Maps Vulkan to Metal

MoltenVK for macOS and iOS
For macOS 10.11, iOS 9.0 and up

Vulkan macOS SDK
Valve - Vulkan Dota 2 on macOS

Shipping Now. Vulkan delivering up to 50% performance increase over native OpenGL

<table>
<thead>
<tr>
<th>Model</th>
<th>OpenGL FPS</th>
<th>Vulkan FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD FirePro D500 Mac Pro</td>
<td>75.3</td>
<td>102.8</td>
</tr>
<tr>
<td>NVIDIA GT 650M Macbook Pro</td>
<td>35.9</td>
<td>53.9</td>
</tr>
<tr>
<td>Intel Iris Pro Macbook Pro</td>
<td>42.2</td>
<td>47.7</td>
</tr>
</tbody>
</table>

Dota 2 OpenGL vs Vulkan macOS

© Copyright Khronos™ Group 2018 - Page 21
OpenGL and OpenGL ES

January 2018
OpenGL 4.6 conformance test suite released in open source
Intel and NVIDIA released conformant OpenGL 4.6 drivers
April 2018
OpenGL 4.6.0.1 CTS bugfix update released in April

June 2018
OpenGL ES CTS 3.2.5.0 released in open source
Raises the quality bar for OpenGL 3.2 implementations

Working Group Meetings Merged under one Chairperson for Improved Efficiency
GLSL and ESSL specs merged and migrated from LibreOffice to AsciiDoctor to improve maintainability and reduce divergence
OpenGL 4.6, OpenGL ES 3.2, GLSL 4.60 and ESSL 3.20 specs June 2018
Lots of bug fixes - many leveraged from open GitHub projects

OpenGL ES still the most prevalent 3D API (billions of units!)
More conformant products added
OpenGL ES 3.2 adoption increasing

Google data - July 2018

© Copyright Khronos™ Group 2018 - Page 22
OpenGL ES and WebGL Evolution

Pervasive OpenGL ES 2.0
OpenGL and OpenGL ES ships on every desktop and mobile OS. 3D on the Web is enabled!

Desktop Graphics
Textures: NPOT, 3D, Depth, Arrays, Int/float
Objects: Query, Sync, Samplers
Seamless Cubemaps, Integer vertex attributes
Multiple Render Targets, Instanced rendering
Transform feedback, Uniform blocks
Vertex array objects, GLSL ES 3.0 shaders

Mobile Graphics
Programmable Vertex and Fragment shaders

Compute Shaders

Advanced Graphics
Tessellation and geometry shaders
ASTC Texture Compression
Floating point render targets
Debug and robustness for security

Mobile and Desktop Graphics

2007
OpenGL ES 2.0

2012
OpenGL ES 3.0

2014
OpenGL ES 3.1
Compute Shaders

2015
OpenGL ES 3.2

WebGL

2011
WebGL 1.0

4 years

2014
Compute Shaders

5 years

March 2017
WebGL 2.0

Work in Progress

Conformance Testing is vital for Cross-Platform Reliability
WebGL 2.0 conformance tests are very thorough 10x more tests than WebGL 1.0 tests
WebGL Momentum - WebGL 2.0 is Here!

WebGL 2.0 brings Desktop-class graphics to the Web
The time to create a new class of Web-based 3D Apps is now!

http://caniuse.com/#feat=webgl
### Ecosystem = API + File Format

<table>
<thead>
<tr>
<th>Ecosystem Area</th>
<th>Run-time APIs</th>
<th>File Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Graphics</td>
<td>Vulkan, WebGL</td>
<td>glTF</td>
</tr>
<tr>
<td>Heterogenous Compute (Parallel Processing)</td>
<td>SYCL, Vulkan, OpenCL</td>
<td>SPIR, Vulkan</td>
</tr>
<tr>
<td>VR and AR Vision and Inferencing</td>
<td>OpenVX, OpenXR</td>
<td>NNEF</td>
</tr>
</tbody>
</table>
JSON (ISO/IEC 21778:2017, ECMA 404)

- X3D, glTF already use it
glTF for IS

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

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

.png
.jpg
...
Textures

Geometry
Texture based
PBR materials
glTF - Cross-Platform 3D Asset Transmission

<table>
<thead>
<tr>
<th>Audio</th>
<th>Video</th>
<th>Images</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3</td>
<td>H.264</td>
<td>JPEG</td>
<td>glTF</td>
</tr>
</tbody>
</table>

- New market opportunities for 3D content creation and deployment!

All glTF spec development on open GitHub: https://github.com/KhronosGroup/glTF

- Compact to Transmit
- Fast to Load
- Describes Full Scenes
- Runtime Neutral
- Open and Extensible

OpenGL Transmission Format
Efficient transmission of 3D scenes and assets

glTF 1.0 - Primarily for WebGL
Uses GLSL for materials
Released December 2015

glTF 2.0 - Physically Based Rendering
Metallic-Roughness and Specular-Glossiness Materials
Rendering API independence
Released at Web3D 2017
glTF Ecosystem

Creation Tools
- blender
- SketchUp
- PIXYZ
- Archilogic
- Adobe
- SUBSTANCE PAINTER
- Minecraft
- Sony 3D Creator
- Oculus
- Modo
- medium
- Paint 3D
- instant3Dhub
- SIMPLYGN
- Assimp
- SAFE SOFTWARE

Repositories
- TURBOSQUID
- Sketchfab
- Remix 3D
- Poly
- poly.google.com

Discover

Create

Experience

Drive Demand

Apps and Engines
- Unreal
- unity
- Office
- three.js
- babel
- babylon.JS
- Godot
- Game engine
- Windows Mixed Reality Home
- Cesium
- Autodesk Forge
- A-Frame
- React VR
- JanusVR
glTF Recent Highlights

• TurboSquid adds glTF to StemCell - 60K+ 3D artists and 700K 3D models
  - https://www.khronos.org/blog/turbosquid-adds-gltf-to-supported-formats-for-its-stemcell-initiative

• Open Geospatial Consortium 3D Tiles standard proposal references glTF
  - Designed for streaming massive heterogeneous 3D geospatial datasets
  - http://www.opengeospatial.org/pressroom/pressreleases/2829

• Widespread Adoption
  - Microsoft makes glTF files as usable as JPGs in Windows 10
  - Facebook supports drag and drop for glTF models to your feed
  - Adobe Dimension using glTF for delivery of 3D marketing assets
  - Mozilla integrating glTF into A-FRAME
  - Sketchfab repository has over 150K glTF models

• Google Draco Mesh Compression
  - Extension is shipping in tools and engines

• Careful roadmap developments
  - Unlit materials and texture transforms
  - Texture Transmission format...
glTF Texture Transmission Extension

Transcodable, supercompressed textures for efficient transmission
- 25% size of the equivalent native GPU encoding.
- Rate-distortion optimization (RDO) for fine-grain control over quality vs bitrate.
- Optional LZ/ANS lossless codec stage for maximized compression efficiency.
- Support for both low precision formats for transcoding to ETC1, and high precision (128-bit block) format for full ASTC and BC6H/BC7, HDR etc.

Extension in design - looking for industry feedback

https://github.com/KhronosGroup/glTF/issues/1051
Google Draco

- Open Source Project of Google
- Compression (Lossy) of 3D Asset based on glTF
- Possible to apply on 3D Scanning data (point cloud)
New Activities

- **Initiative for Heterogeneous Communication**
  - Abaco Systems Takyon API is proposed.
  - P2P communication API between GPU’s and MPU’s

- **“Safety Critical” - New WG (Advisory Panel changed into official WG)**
  - OpenVX, OpenCL, NNEF, Vulkan, OpenGL etc.
  - Mainly targeted to Automotive Application

- **Liaisons**
  - SC 29 MPEG - for NNEF
  - OGC - for OpenXR
  - GENIVI - Open Source Infotainment in Automotive
Liaison Report

• Khronos Group hope to build “official” liaison between SC 24
  - Past board meeting, liaison request is approved.
  - JTC 1/ SC 24 Liaison Category A
  - Letter will be delivered to the secretary of SC 24, Soon
  - Topics - glTF for PAS, Collaboration in OpenXR

• Liaison between JTC 1/WG 12 (SC 3 → WG 12)
  - Past board meeting, liaison request is approved.
  - Liaison Category C
  - 3D point cloud data based on glTF for 3D Scanning
Thank you

• Please contact
  - Hwanyong LEE, hwanyong.lee@gmail.com