Press Briefing
SIGGRAPH 2015
Neil Trevett | Khronos President
NVIDIA Vice President Mobile Ecosystem
Khronos Connects Software to Silicon

Open Consortium creating ROYALTY-FREE, OPEN STANDARD APIs for hardware acceleration

Defining the roadmap for low-level silicon interfaces needed on every platform

Graphics, compute and vision processing

Rigorous specifications AND conformance tests for cross-vendor portability

Acceleration APIs BY the Industry FOR the Industry

Well over a BILLION people use Khronos APIs Every Day...
Khronos Open Standards for Graphics and Compute

1990’s | Workhorse cross-platform professional 3D apps & gaming

2000’s | Ubiquitous mobile gaming & graphics apps

2005 | Safety Critical Graphics

2008 | Heterogeneous parallel compute

Portable intermediate representation for graphics and parallel compute 2014

High-efficiency GPU graphics and compute for performance critical apps 2015

SIGGRAPH NEWS

New Extensions to enable latest desktop graphics capabilities

OpenGL ES 3.2 released today to bring AEP functionality to core

New Safety Critical Working Group - Call for Participation

OpenCL 2.0 specification update and C++ Headers released

Provisional Spec Update and significant open source activity

Adopted by Android and other platforms. Building ecosystem
OpenGL Extensions Released Today

- Expose cutting-edge desktop graphics
  - Paving the way for new core OpenGL when functionality pervasively available

- Streamlined sparse textures
  - Manage multisample sparse textures and uncommitted/unpopulated areas

- Enhanced shaders
  - Interlocks for multi-pass algorithms, 64-bit integer handling
  - Early fragment testing, enhanced atomic counters, 64-bit timing counter

- Multi-thread shader compilation
  - Faster application loading

- Modifiable locations of multi-samples within a pixel
  - Increase antialiasing quality

- OpenGL ES 3.2 compatibility
  - Use OpenGL to develop OpenGL ES 3.2 applications
OpenGL ES 3.2 Released Today

- Bringing advanced desktop-class graphics functionality into core OpenGL ES
  - Will make pervasive on large class of mobile, consumer and automotive hardware

- New functionality includes AEP (Android Extension Pack) for AAA games plus more
  - Tessellation and geometry shaders to improve add scene detail
  - ASTC Texture Compression - reduces memory footprint and bandwidth
  - Floating point render targets for higher precision compute
  - Debug and robustness for easier coding and secure execution
  - Enhanced blending operations

- Enables premium graphics effects
  - Deferred shading, Physically-based shading
  - High Dynamic Range tone mapping
  - Global Illumination and reflection
  - Smoke and particle effects

- Will be adopted by Android
  - And many other platforms

Epic’s Rivalry demo using full Unreal Engine 4
https://www.youtube.com/watch?v=jRr-G95GdaM
Safety Critical Working Group

Call for participation to create OpenGL SC 2.0 and future safety critical APIs!

2003
OpenGL ES 1.0
Fixed function graphics

2005
OpenGL SC 1.0
Fixed function graphics subset

2007
OpenGL ES 2/3
Programmable shader pipeline

2016 (planned)
OpenGL SC 2.0
Programmable shader pipeline subset

New Generation API for safety certifiable graphics AND compute

Many future safety critical use cases involve vision and compute acceleration (e.g. neural nets)
OpenCL 2.0 Updates Released Today

- Updates to OpenCL 2.0 specification
  - Bug fixes, clarifications
- OpenCL C++ Headers for OpenCL 2.0 also released today
  - Enables host side C++ code
- New OpenCL 1.2 implementations!
  - NVIDIA (Desktop), Texas Instruments (Embedded), Marvell (Mobile & Embedded)
- OpenCL 2.1 Provisional specification released in March 2015
  - OpenCL C++ kernel language based on a subset of C++14
  - Support for SPIR-V intermediate language in core
  - Runs on any OpenCL 2.0-capable hardware

---

Dec08  
OpenCL 1.0 Specification

18 months

Jun10  
OpenCL 1.1 Specification

18 months

Nov11  
OpenCL 1.2 Specification

24 months

Nov13  
OpenCL 2.0 Specification

16 months

Mar15  
OpenCL 2.1 Specification (Provisional)
Driving the SPIR-V Open Source Ecosystem

**SPIR-V**
- 32-bit Word Stream
- Extensible and easily parsed
- Retains data object and control flow information for effective code generation and translation

Khronos will open source these tools and translators

- SPIR-V Tools
  - SPIR-V Validator
  - SPIR-V (Dis)Assembler
  - LLVM

- OpenCL C
  - OpenCL C++

GLSL

Third party kernel and shader Languages

Community Open Source Projects Underway
- SPIR-V generators
- Bytecode/IL to SPIR-V
- SPIR-V Virtual Machines
- Domain specific languages
- Specification tools

Other Intermediate Forms

IHV Driver Runtimes

SPIR-V Provisional spec (V31) updated today!
SPIR-V Open Source Community Activity

- **Python byte code to SPIR-V Convertor**
  - Write shaders or kernels in Python, Encode and decode SPIR-V in Python
  - Dis(Assembler) with high level human readable assembler syntax
- **.NET IL to SPIR-V Convertor**
  - Write and debug shaders or kernels using C#, SPIR-V interpreter
- **Shade SPIR-V virtual machine**
  - Test and debug SPIR-V binaries for binary correctness in human readable format
- **Otherside SPIR-V virtual machine**
  - Academic software rasterizer project to produce C code from SPIR-V
- **Rust (Dis)Assembler**
  - Encode and decode SPIR-V binaries in Rust
- **Go (Dis)Assembler**
  - Encode and decode SPIR-V in Go, SPIR-V represented in Go data structures
- **Haskell EDSL**
  - SPIR-V like language embedded in Haskell with significantly relaxed layout constraints
- **Lisp SPIR-V Specification**
  - Lisp readable SPIR-V specification
- **JSON SPIR-V specification**
  - Conversion of HTML SPIR-V specification to JSON format
- **This is just the start....**
Next Generation GPU APIs

Ground-up API designs for driving high-efficiency graphics and compute on GPUs

Platforms with announced Vulkan support:
- Android will adopt Vulkan
- Windows 7, 8 and 10, SteamOS, Tizen
- Many Linux distributions including Ubuntu and Red Hat
Vulkan Tools Architecture

• Layered design for cross-vendor tools innovation and flexibility
  - IHVs plug into a common, extensible architecture for code validation, debugging and profiling during development without impacting production performance

• Common Loader used to enable use of tools layers during debug
  - Finds and load drivers, dispatches API calls to correct driver and layers

• Vulkan supports intercepting or hooking API entry points
  - Multiple layers may be chained together to cascade their functionality
**LunarG Open Source SDK Layers**

- Will be released in open source in parallel with Vulkan specification

<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIDump</td>
<td>Print API calls and their parameters and values</td>
</tr>
<tr>
<td>DrawState</td>
<td>Validate the descriptor set, pipeline state and dynamic state</td>
</tr>
<tr>
<td>Image</td>
<td>Validate texture formats and render target formats</td>
</tr>
<tr>
<td>MemTracker</td>
<td>Track &amp; validate GPU memory, its binding to objects &amp; command buffers</td>
</tr>
<tr>
<td>ObjectTracker</td>
<td>Track all Vulkan objects and flag invalid objects and object memory leaks</td>
</tr>
<tr>
<td>ParamChecker</td>
<td>Validate API parameter values</td>
</tr>
<tr>
<td>ShaderTracker</td>
<td>Validate the interfaces between SPIR-V modules and the graphics pipeline</td>
</tr>
<tr>
<td>Threading</td>
<td>Check validity of multi-threaded API usage</td>
</tr>
</tbody>
</table>
Vulkan Feature Sets

- Vulkan supports hardware with a wide range of hardware capabilities
  - Mobile OpenGL ES 3.1 up to desktop OpenGL 4.5 and beyond

- One unified API for desktop, mobile, console, and embedded
  - No "Vulkan ES" or "Vulkan Desktop"

- Vulkan precisely defines a set of "fine-grained features"
  - Features are specifically enabled at device creation time (similar to extensions)

- Vulkan provides the mechanism but does not mandate policy
  - Market or platform specific profiles can be defined by Khronos, platform vendors, or other interested parties
Vulkan Window System Integration (WSI)

- Explicit control for acquisition and presentation of images
  - Designed to fit the Vulkan API and today’s compositing window systems
  - Cleanly separates device creation from window system

- Standardized extensions - unified API for multiple window systems
  - Works across Android, Mir, Windows (Vista and up), Wayland and X (with DRI3)
  - Platforms can extend functionality, define custom WSI stack, or have no display at all

- Platform provides an array of persistent presentable images = Vulkan Swapchain
  - Device exposes which queues support presentation
  - Application explicitly controls which image to render and present
Vulkan Open Source Conformance Tests

Khronos and Android leveraging and merging Vulkan tests with the Android Open Source Project (AOSP) and DrawElements Quality Program (dEQP) framework

Khronos Generated Tests

AOSP/dEQP Framework and Tests

Open source Vulkan conformance test suite hosted on GitHub

Vulkan Conformance Test Suite

Implementers must pass Test Suite

Community Test Contributions

Enable developers to provide direct feedback and contributions to help resolve cross-vendor inconsistencies

Khronos administered Vulkan Adopters Program. Implementations that pass test process may use Vulkan trademark
Vulkan Status

- Rapid progress since project start in June 2014
  - Significant proposals and IP contributions received from members
- Participants come from all segments of the graphics industry
  - Including an unprecedented level of participation from game engine ISVs
- Initial specs and implementations expected this year
  - Will work on any GPU hardware that supports OpenGL ES 3.1/OpenGL 4.X and up
  - Can ship on any OS - including Windows XP/7/8/10
Summary

- Khronos is advancing the scope of the 3D and compute API ecosystem
  - Existing and new generation APIs in parallel
- Advanced capabilities AND cross-platform portability
  - Key to many emerging markets such as Augmented Reality and Virtual Reality
- Any company welcome to join for a voice and vote in creating these standards
  - $15K annual membership fee for access to all Khronos API working groups

More Information
- www.khronos.org
- ntrevett@nvidia.com
- @neilt3d