Portable VR with Khronos OpenXR

Kaye Mason, Nick Whiting, Rémi Arnaud, Brad Grantham
July 2017
Outline for the Session

- Problem, Solution
- Panelist Introductions
- Technical Issues
- Audience Questions
- Lightning Round
- How to Join
- Party!
Problem

Without a cross-platform standard, VR applications, games and engines must port to each vendors’ APIs.

In turn, this means that each VR device can only run the apps that have been ported to its SDK. The result is high development costs and confused customers - limiting market growth.
Solution

The cross-platform VR standard eliminates industry fragmentation by enabling applications to be written once to run on any VR system, and to access VR devices integrated into those VR systems to be used by applications.
Architecture

OpenXR defines two levels of API interfaces that a VR platform’s runtime can use to access the OpenXR ecosystem. Apps and engines use standardized interfaces to interrogate and drive devices. Devices can self-integrate to a standardized driver interface. Standardized hardware/software interfaces reduce fragmentation while leaving implementation details open to encourage industry innovation.
OpenXR Participant Companies
Panelist Introduction

- Kaye Mason

Dr. Kaye Mason is a senior software engineer at Google and the API Lead for Google Daydream, as well as working on graphics and distortion. She is also the Specification Editor for the Khronos OpenXR standard. Kaye has a PhD in Computer Science from research focused on using AIs to do non-photorealistic rendering. She spent ten years as a AAA video game developer -- shipping over a dozen titles. The first was Lord of the Rings: Return of the King, and the last was Sims 4. At Google, she worked on Google Earth before her work on the original Cardboard launch led to a move to Daydream. Her goal is to make VR universally accessible and useful.
Panelist Introductions

- Rémi Arnaud

Dr. Rémi Arnaud serves as Chief Architect Officer at Starbreeze, leading developments such as the StarVR SDK. Involved early on with real-time image generation in Paris where he did his Ph.D., he then relocated to California and since has worked on many projects including Silicon Graphics IRIS Performer, Keyhole’s Earth Viewer, Intrinsic Graphics’ Alchemy, Sony’s PS3 SDK, Intel’s Larrabee Game Engine, Screampoint’s 5D City, Fl4re’s game engine. Collaborated to various Khronos groups including OpenGL ES, COLLADA, glTF, WebGL, webCL, and OpenXR.
Panelist Introductions

• Brad Grantham

Brad Grantham currently specializes in workload analysis for GPU architecture in ARM Inc.’s Media Processing Group. He has been an OpenGL developer since 1997 and has presented at SIGGRAPH and ARM TechCon. Brad previously worked on IRIS Performer and the Fahrenheit Scene Graph at Silicon Graphics; benchmarking, samples, and applications engineering at ATI (now AMD); and driver design and implementation at ARM. He recently wasted a lot of personal time writing an IrisGL-compatible transformation and lighting library for a handheld retrocomputer.
Panelist Introductions

• Nick Whiting

Dr. Nick Whiting is currently overseeing the development of the award-winning Unreal Engine 4's virtual and augmented reality efforts, including shipping the recent "Robo Recall", "Bullet Train," "Thief in the Shadows," "Showdown," and "Couch Knights" VR titles. Previously, he has helped shipped titles in the blockbuster "Gears of War" series, including "Gears of War 3" and "Gears of War: Judgment." He is also currently serving as the chair of the Khronos OpenXR initiative, working to create a standard for VR and AR platforms and applications.
Introductory Questions

- What is your personal involvement in OpenXR?
- Why are you or your company involved?
- How will having a standard affect your work?
- How do you think having a standard will benefit the VR community?
Mobile XR

- Brad Grantham

~1.5B smartphones shipped in 2016

Mobile game revenue exceeded console and PC in 2016 (DFC Intelligence)

Constrained

~1 Watt Thermal throttling

Managing power use is critical

Developer tools to measure and manage power & temp are essential
Large Field of View

- Rémi Arnaud
Colo(u)r Spaces

- Nick Whiting

Today’s HMDs use a wide variety of panels and drivers, which produce a wide range of responses. The same content can look completely different on two different devices!

How can a standard help us get a more consistent output between multiple devices?
Performance Levels

- Kaye Mason

Sustained Performance API

Performance can fluctuate dramatically for long-running apps, because the system throttles system-on-chip engines as device components reach their temperature limits. This fluctuation presents a moving target for app developers creating high-performance, long-running apps.

To address these limitations, Android 7.0 includes support for sustained performance mode, enabling OEMs to provide hints about device-performance capabilities for long-running apps. App developers can use these hints to tune apps for a predictable, consistent level of device performance over long periods of time.


Power State Notification and Mitigation Strategy

The mobile SDK provides power level state detection and handling.

Power level state refers to whether the device is operating at normal clock frequencies or if the device has risen above a thermal threshold and thermal throttling (power save mode) is taking place. In power save mode, CPU and GPU frequencies will be switched to power save levels. The power save levels are equivalent to setting the fixed CPU and GPU clock levels to (0, 0). If the temperature continues to rise, clock frequencies will be set to minimum values which are not capable of supporting VR applications.

Once we detect that thermal throttling is taking place, the app has the choice to either continue operating in a degraded fashion or to immediately exit to the Oculus Menu with a head-tracked error message.

Oculus Developer Center (Power Management) (https://developer.oculus.com/documentation/mobilesdk/1.0.3/concepts/mobile-power-overview/)
Location-Based Entertainment

• Rémi Arnaud
Mobile Video

- Kaye Mason

Secure texture video playback

Android 7.0 supports GPU post-processing of protected video content. This allows using the GPU for complex non-linear video effects (such as warps), mapping protected video content onto textures for use in general graphics scenes (e.g., using OpenGL ES), and virtual reality (VR).

Android Open Source Project Documentation (Surface Texture)
https://source.android.com/devices/graphics/arch-st
Future Proofing

• Nick Whiting

We have a great group of contributors to the workgroup, so we have a pretty good idea of what kind of devices will be coming down the pipe in the near future.

If the past five years have taught us nothing, it’s that we have no idea what HMDs and input devices will look like even a few years into the future! How do we make sure applications developed today can run on future hardware?
Compositing Layers

- Brad Grantham

Reprojection and correction

Many types of image content
  3D content, UI, video, etc

Implications on system performance
  GPU, CPU, video, display

Balance between ubiquity and flexibility
  Define baseline
  Allow innovation with extensions
Frame Scheduling and Blocking

- Kaye Mason

16.66ms

App Submit (CPU)

App Render (GPU)

Reprojection Submit (CPU)

Reprojection Render (GPU)

Hardware Compositor

Submit Buffer

Render Buffer

Latch buffer

Read L

Read R

User Input

Up to 62.5ms (@ 60 Hz)

VSync

VSync

16.66ms

16.66ms

16.66ms

Chart Credits: Adam Gousetis @ Google
Audience Questions
Lightning Round

- Eye tracking?
- Loader and Mobile?
- Hardware Detection?
  - Coordinate Systems
- Runtime and Device Selection
  - how do you allow control at the application and / runtime level?
- Inter-process communication
  - should it be part of the standard?
- Spacewarp / frame interpolation / framerate upsampling?
Participate

- Have an opinion?

https://www.khronos.org/members/
Thank You
BoF Blitz
After-Party

Come Back at 5:45 for the After-Party

Sponsored by:

CESIUM
NVIDIA

WWW.KHRONOS.ORG