Vulkan: One API for all Platforms

Kevin Sun
Lead Developer Support Engineer, APAC PowerVR Graphics
Introduction

Who am I?

- Kevin Sun
  - Working at Imagination Technologies
    - Take responsibility for PowerVR technical support in APAC region
Introduction

This series

- Detailed look at Vulkan
- No new information about the API
  - Khronos’ job!
- Adding context to existing information
  - Primarily from SIGGRAPH
  - https://www.youtube.com/v/quNsdYfWXfM
- This week: What does “One API” mean?
What does “One API” mean?
What does “One API” mean?

- Vulkan is consistently available
  - A wide range of targets
  - Multiple platforms
  - Multiple GPU vendors

- Consistent Experience
  - Same API everywhere

- Differentiation of targets
  - Describe the platform, allow informed choices
Consistencies

*Coding to the API*

- Same library of core functions
  - Link to the same library
  - Use the same header

- Write the same code
  - No preprocessor needed
  - Write one code path
Consistencies

*Performance*

- Few ways to do things
  - Less scope for “picking the slow path”
  - All vendors can optimise each path

- Same fast paths
  - One fast way to achieve a goal
    - E.g. Clearing via Render Pass load ops

- Hardware-specific optimisations less necessary
  - Still needed for last few %
  - Useful for cutting edge techniques
Differences

Capabilities

- Some functionality is optional
  - Power, efficiency, or area reasons
  - Other reasons may exist

- Optional means just that
  - Can still create fantastic apps without them

- Explicitly enable features
  - At device creation time
  - Avoids accidental use
Differences

**Limits**

- Various things have limits
  - E.g. maximum texture size

- Minimum values for all of these
  - Consistent values available
  - Scope to expose higher values
Differences

Extensions

- Extensions enable additional functionality
  - Expose features not in core
  - Fantastic for prototyping

- Similar to other Khronos APIs
  - Vendor
  - Multi-vendor
  - Khronos ratified

- Must be explicitly enabled
  - Same as optional features
Differences

Feature Sets

- Packages of functionality
  - May guarantee higher minimum limits
  - May guarantee extension support

- Not explicitly part of the API
  - Considerations being made though
  - Khronos not committed to any yet
  - External parties may define them

- Profiles intended as a guide
  - Efficiency vs. Comprehensive
Differences

**Platform Integration**

- Platform integration will be defined as an extension
  - Not necessary for conformant Vulkan

- Few core extensions
  - Augmented with platform-specific functionality

- No obligation to use these extensions
  - But designed to work consistently

- See [Alon Or-bach’s talk at the SIGGRAPH 2015 BoF](#)
Platform Support

- Designed to run on any platform
- Input specifically for these platforms:

- Android
- Tizen
- Red Hat Linux
- SteamOS
- Ubuntu
- Windows
Conclusion

- Not limited to subset of platforms
  - Potential for very wide adoption
  - Thus – large market share

- Lots of interest across the board
  - Platforms vendors, GPU vendors, Developers
  - Will develop a large ecosystem over time

- Vulkan set to make a big impact on the industry
  - Will cover why and how in next few talks
Follow us on social media

Twitter and blog

- Developer SDK and tools + free support
  - www.powervrinsider.com

- Check out our blog posts on Vulkan
  - http://blog.imgtec.com/tag/vulkan

- WeChat
  - Imgtec

- Follow us on Twitter for the latest updates
  - @ImaginationTech
  - @PowerVRInsider
Vulkan: One API for all Platforms

Kevin sun
Lead Developer Support Engineer, APAC PowerVR Graphics