Vulkan™ Tools

Filling the gap between driver and application

Khronos Munich Chapter Meeting - 04/08/2016

Norbert
Nopper

VulKan ToolS - Agenda

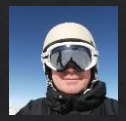
- In advance: Run Vulkan examples / show source code / open Blender in between

- One slide about me
- Motivation – Why VulKan ToolS (VKTS)
- VKTS Architecture
- Important VKTS modules and features
- VulKan ToolS - What's coming next?
- Questions & Answers

Norbert
Nopper

One slide about me

- Computer scientists with focus on computer graphics
  - Some computer games
  - More automotive applications
- Graphics engineer working @ NXP (Khronos Group Member)
  - Focus on Vulkan
- Private open source projects on GitHub:



https://github.com/McNopper/

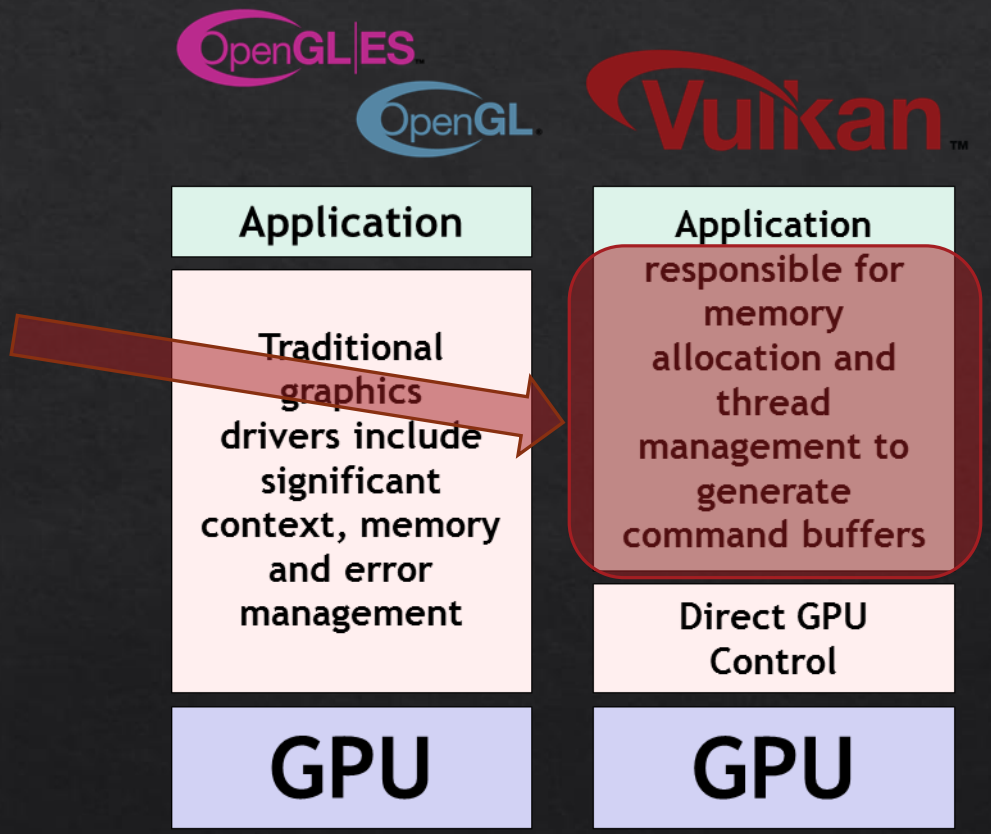
  - GLUS – part of the official OpenGL Software Development Kit
  - More than 40 core OpenGL 3 and 4 examples
  - VKTS – listed on the Khronos Vulkan website
- By the way … I'm a gamer 

Motivation – Why VKTS?

- "I want to use Vulkan in my application!"
- "Get a 3D Engine – but you will not *see* Vulkan."
- "No 3D Engine – I want to code!"
- "Beside the application code – you need this …"

- [Silence]

- "Let's create a wish list!"



Norbert Nopper

Motivation – Why VKTS? (cont.)

- Wish list
  - General
    - Easy to share code (no NDA with customers): The MIT License ✓
  - Focus
    - Learning Vulkan
    - Testing Vulkan drivers
    - Validate Vulkan application bugs
    - **Implement Vulkan demos**
  - Mandatory
    - Filling the “driver” gap

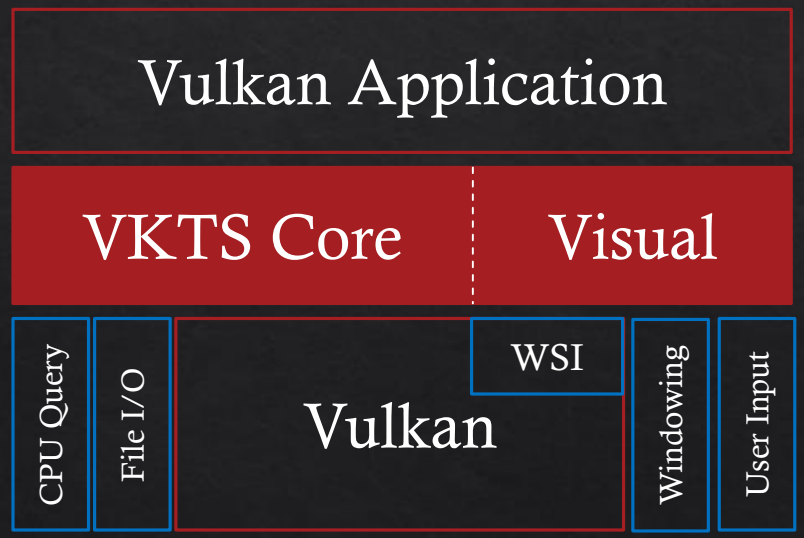
Motivation – Why VKTS? (cont.)

- Wish list (cont.)
  - Cross platform
    - Android, Linux, Windows – but also Integrity, QNX and maybe *Bare Metal* (same code everywhere)
  - Support for multi-threading
    - One of Vulkan design philosophy
  - **Implement sophisticated Vulkan demos**
    - Beside Phong also BSDF* materials
    - Node based and skeletal animations
  - Blender as primary 3D Tool
    - Need easy human readable and to parse scene file format

* https://en.wikipedia.org/wiki/Bidirectional_scattering_distribution_function

VKTS Architecture

- Filling the “driver” gap ✓
- Cross platform ✓



- Provides “filling the gap” modules e.g.
  - VkImage / VkBuffer staging
  - MIP map generation
- “True” cross platform
  - Same application code on any platform

  - C++11, GLM* and Vulkan – that’s it!
  - Encapsulated platform specific code you do not see

* http://glm.g-truc.net

Norbert Nopper

Important VKTS modules and features

- Multi-threaded “engine” ✓
  - Threaded applications (not a have to)
  - Job/task system (not a have to)
- Query amount of CPU core available
- Image loading and saving (TGA and HDR)
- Input handling
  - Gamepad, Keyboard and Mouse
- Display and window handling
  - Multi-window and full screen supported

Important VKTS modules and features

- Blender as primary 3D Tool ✓

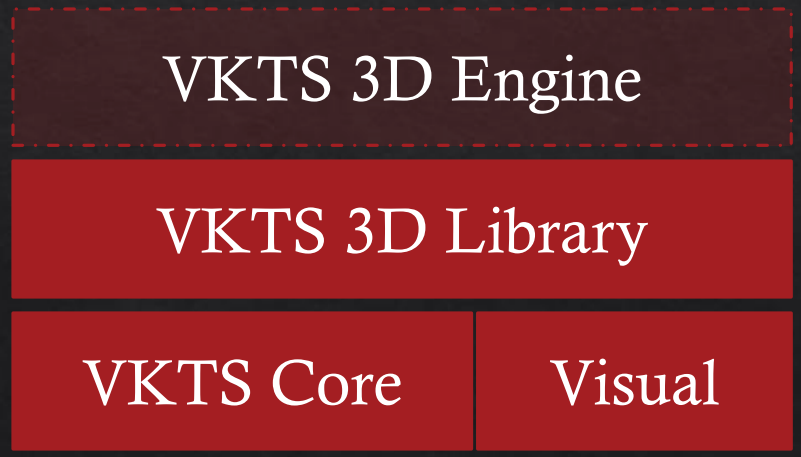
- Blender scene exporter
  - Node hierarchy
  - Meshes including sub meshes
  - Phong materials
  - Animations
  - Armatures
- Proprietary VKTS scene file format
  - Similar like Wavefront .obj files*

* https://en.wikipedia.org/wiki/Wavefront_.obj_file

Norbert Nopper

VulKan ToolS - What's coming next?

- **Implement sophisticated Vulkan demos** ☒

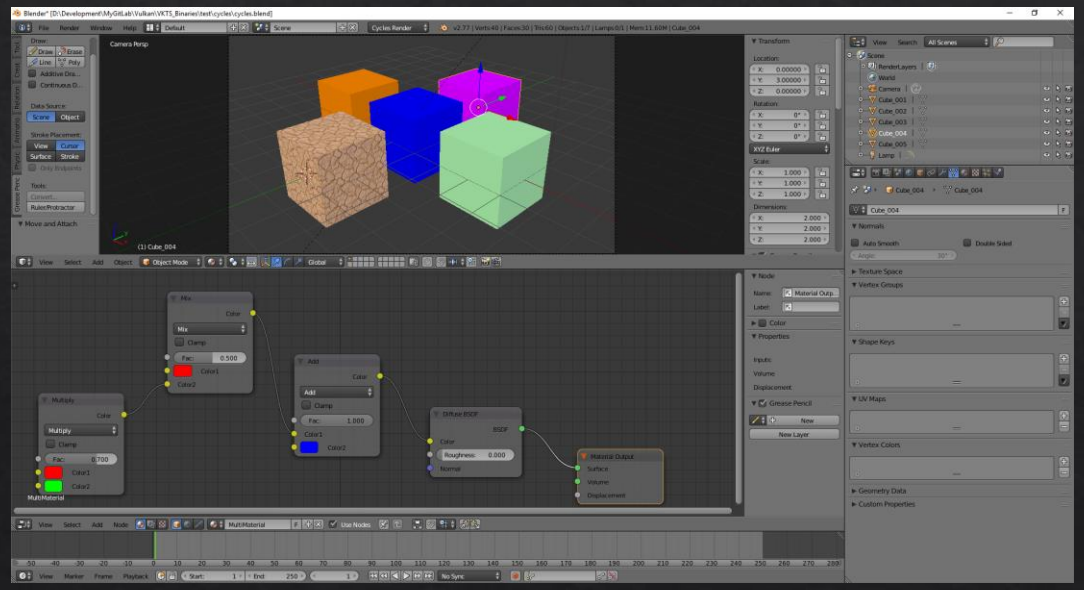


- VKTS currently has two layers
  - First can be used independently
  - Second depends on first
- New third layer
  - Third depends on second
  - Will setup "everything"

Norbert Nopper

VulKan ToolS - What's coming next?

- **Implement sophisticated Vulkan demos** ☒



- Exporting of Blender Cycles materials
  - Generating GLSL code as material
  - Using build SPIR as material
  - Using deferred rendering

Norbert Nopper

Questions & Answers

later …

much later …

@McNopper

Norbert Nopper