Logical Devices

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

Logical Devices

Vulkan: Overall Block Diagram

Application

Instance

Physical Device

Logical Device

Command Buffer

Vulkan: a More Typical (and Simplified) Block Diagram

Application

Instance

Physical Device

Logical Device

Command Buffer

const char * myDeviceLayers[ ] =
{
// "VK_LAYER_LUNARG_api_dump",
// "VK_LAYER_LUNARG_core_validation",
// "VK_LAYER_LUNARG_image",
"VK_LAYER_LUNARG_object_tracker",
"VK_LAYER_LUNARG_parameter_validation",
// "VK_LAYER_NV_optimus"
};

const char * myDeviceExtensions[ ] =
{
"VK_KHR_surface",
"VK_KHR_win32_surface",
"VK_EXT_debug_report"
// "VK_KHR_swapchains"
};

// see what device layers are available:
uint32_t  layerCount;
vkEnumerateDeviceLayerProperties(PhysicalDevice, &layerCount, (VkLayerProperties *)nullptr);

VkLayerProperties * deviceLayers = new VkLayerProperties[layerCount];
result = vkEnumerateDeviceLayerProperties( PhysicalDevice, &layerCount, deviceLayers);

Looking to See What Device Layers are Available

// see what device extensions are available:
uint32_t  extensionCount;
vkEnumerateDeviceExtensionProperties(PhysicalDevice, deviceLayers[i].layerName, &extensionCount, (VkExtensionProperties *)nullptr);

VkExtensionProperties * deviceExtensions = new VkExtensionProperties[extensionCount];
result = vkEnumerateDeviceExtensionProperties(PhysicalDevice, deviceLayers[i].layerName, &extensionCount, deviceExtensions);

What Device Layers and Extensions are Available

4 physical device layers enumerated:
0x00400033   1  "VK_LAYER_LUNARG_core_validation"  'LunarG Validation Layer'
0 device extensions enumerated for 'VK_LAYER_LUNARGCoreValidation'
0x00400303   1  "VK_LAYER_LUNARG_object_tracker"  'LunarG Validation Layer'
0 device extensions enumerated for 'VK_LAYER_LUNARGObjectTracker'
0x00400303   1  "VK_LAYER_LUNARG_parameter_validation"  'LunarG Validation Layer'
150 device extensions enumerated for 'VK_LAYER_LUNARGParameterValidation'
```c
VkDeviceCreateInfo vdci;
vdci.sType = VK_STRUCTURE_TYPE_DEVICE_CREATE_INFO;
vdci.pNext = nullptr;
vdci.flags = 0;
vdci.queueCreateInfoCount = 1; // # of device queues
vdci.pQueueCreateInfos = &vdqci; // array of VkDeviceQueueCreateInfo's
vdci.enabledLayerCount = sizeof(myDeviceLayers) / sizeof(char *);
vdci.enabledLayerCount = 0;
vdci.ppEnabledLayerNames = myDeviceLayers;
vdci.enabledExtensionCount = sizeof(myDeviceExtensions) / sizeof(char *);
vdci.ppEnabledExtensionNames = myDeviceExtensions;
vdci.pEnabledFeatures = &PhysicalDeviceFeatures;

result = vkCreateLogicalDevice( PhysicalDevice, &vdci, PALLOCATOR, &LogicalDevice );
```

```c
Vulkan: Creating a Logical Device
```

```c
float   queuePriorities[1] = {
    1.};
```

```c
VkDeviceQueueCreateInfo vdqci;
vqci.sType = VK_STRUCTURE_TYPE_DEVICE_QUEUE_CREATE_INFO;
vqci.pNext = nullptr;
vqci.flags = 0;
vqci.queueFamilyIndex = 0;
vqci.queueCount = 1;
vqci.pQueueProperties = queuePriorities;
```

```c
// get the queue for this logical device:
vkGetDeviceQueue( LogicalDevice, 0, 0, &Queue ); // 0, 0 = queueFamilyIndex, queueIndex
```

```c
Vulkan: Creating the Logical Device's Queue
```