Vulkan Notes

Anton Khabbaz

May 10, 2022

1 GLFW Commands

glfwInit() initialize window

glfwWindowHint() set properties of the window Some are:

- `GLFW_CLIENT_API, GLFW_NO_API` do not create an OpenGL context.
- `GLFW_RESIZABLE, GLFW_FALSE` do not allow resizing of windows.

glfwCreateWindow() with argument (800, 600, ”Vulkan”, nullptr, nullptr) creates a window and return a pointer to it. First two must be width and height, next is name. The fourth parameter allows you to optionally specify a monitor to open the window on and the last parameter is only relevant to OpenGL.

2 Coding Conventions

vk functions

Vk enumerations and structs

VK_ enumeration values

3 Initialization Commands and Structures

This follows from the Triangle and the initial chapter of VulkanProgramming Guide(OpenGL). o To create an instance of a structure in Vulkan one uses

```
VkResult vKCreateInstance(
const vkInstanceCreateInfo*        pcreateInfo,
const vKAllocationCallbacks*       pAllocator,
VkInstance*    pInstance);
```

The structure inputted is a constant so that is only a source of information, and the instance returns the resulting instance data. The *instance* holds all the tracked states of the application.

The OpenGL programming guide uses the following convention:

pInstance pointer to instance

ppEnabledExtensionNames pointer to pointer to type in this case a const char.

4 Enhancing Vulkan

4.1 Layers

vkEnumerateInstanceLayerProperties (&layerCount, availableLayers.data()) will check what layers are available and return in a vector. The struct is `VkLayerProperties`. The layers need to be enabled. Layers are for debugging, parameter validation and logging and printing. Hello triangle uses one layer that starts many of them called

```
‘‘VK\_LAYER\_KHRONOS\_validation’’
```

4.2 Extensions

Adds features to vulkan by incorporating functions and software into the api. There are *instance* extensions that enhance all of Vulkan, and *Device* extensions that are only available on certain devices. These define new functions. GLFW needs specific extensions given by:

```
 glfwExtensions = glfwGetRequiredInstanceExtensions
(&glfwExtensionCount);
```