

# Vulkan

Paolo Bettelini

## Contents

<b>1 Vulkan</b>	<b>2</b>
<b>2 Queue</b>	<b>2</b>
<b>3 Buffers</b>	<b>2</b>
<b>4 Command buffers</b>	<b>2</b>
4.1 Primary command buffers . . . . .	2
4.2 Secondary command buffers . . . . .	2
<b>5 Compute pipelines</b>	<b>2</b>

# 1 Vulkan

Vulkan is a multi-platform low-level graphical interface (API) for GPU rendering.

The GPU is able to perform computation on a lot of data simultaneously (SIMD, Single Instruction stream, Multiple Data stream).

## 2 Queue

The GPU is able to run multiple operations in parallel. The equivalent of a CPU thread is a **queue**. Queues are grouped by **queue families**.

Whenever we want the device to perform an operation, we have to submit this operation to a specific queue under a family. Some queues support only graphical operations, some others support only compute operations, and some others support both.

## 3 Buffers

When we need the GPU to read or write data in memory, we need to use a **buffer**.

## 4 Command buffers

To execute an operation we need to create a command buffer containing a list of commands to execute.

To submit a command buffer we need to synchronize with the GPU. We can also tell the GPU to send back a signal, call **fence** when the operation is done.

### 4.1 Primary command buffers

They can contain any command. They are the only type of command buffer that can be submitted to a queue.

### 4.2 Secondary command buffers

They allow you to store functionality that you can reuse multiple times in primary command buffers.

## 5 Compute pipelines

In order to ask the GPU to perform an operation on some data, we need to write a program for it. A program that runs on the GPU is called a shader.

Shaders are written in a shading language (hlsl, glsl, wgsl, rust-gpu, ...) which is then compiled into an intermediate bytecode called **SPIR-V**.

## 6 Descriptors

When we create a compute pipeline for a shader we must bound it to a **descriptor**. A descriptor can contain a buffer to access, buffer views, images, samples images etc.

Descriptors are grouped by **descriptor sets**. The shader will declare a specific descriptor from a specific set (both indexed from 0).

## 7 Dispatch

To execute a compute pipeline we need to create a command buffer to do so. This is called **Dispatch**.