**Virtual Buffer**

A virtual buffer is a technique for **efficient memory use**. Actually it’s better to use big buffer rather than multiple small one, you put as much data as you can in the big buffer, however this can be very tricky manage. **Virtual Buffers** are here in order to make the **management of big buffer** **easier**.

How does it work?

Buffers are **represented** in Vulkan API by ***VkBuffer* structure**. This is your buffer. So you link the VkBuffer with a ***VirtualBufferPool*** object which got the same size than VkBuffer. Then you create ***VirtualBuffer*** from the *VirtualBufferPool* using the command ***allocateBuffer****(VkBufferUsageFlags flags, uint32\_t size, std::string tag, size\_t =-1)* (Overload = 1)

**VirtualBuffer *getInfo()***