**Virtual Buffers**

**Virtual buffer** is a technique for better **memory mng**. We have the **physical memory** where we allocate buffer. In ***Vulkan API*** the aim in memory mng is to not use too much buffer. The API **deal great** with BIG FAT buffer but not with **too much little buffer**. But it’s hard to put all your data in few big buffers. Virtual buffers are here to solve this issue. Buffers are allocated from **heap** of memory (from I only use one heap). So we create some big buffers for our data. This is where our virtual buffers are useful. The user will not see the few big buffers created. He’s **only** going to see small “virtual buffers”. In Vulkan a buffer is represented by VkBuffer structure which is a kind of id. So each buffer got a **unique** VkBuffer. Moreover in the VkBuffer definition we can see that it is in fact a **pointer**. Our virtual buffer will have a VkBuffer (No need ptr, because it’s a pointer), it will have an id