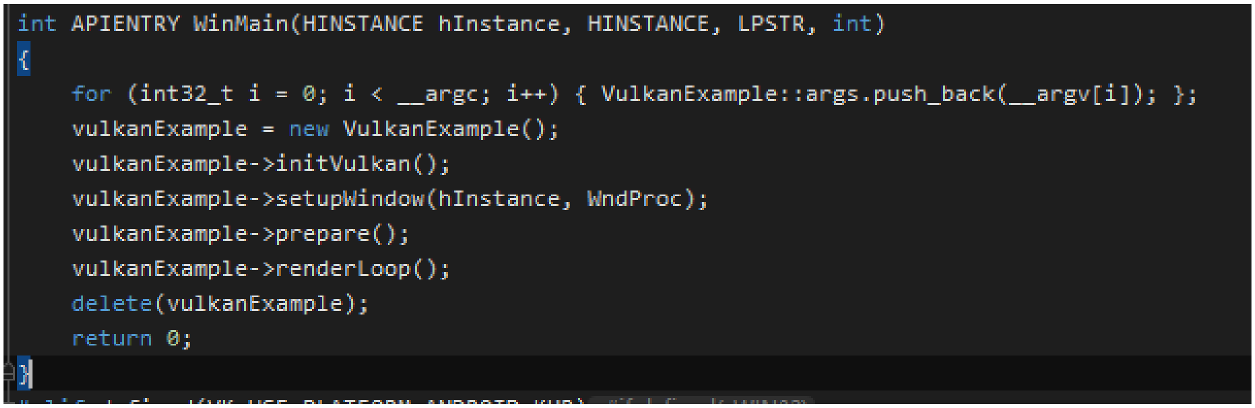
Triangle

triangle 中的VulkanExample继承于vulkanexamplebase，在base 路径下封装了许多Vulkan API，以及一些基本框架，如vulkanexamplebase，Keycodes，Camera等类。

vulkanexamplebase中为Vulkan执行的流程框架：



```
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE, LPSTR, int)
{
    for (int32_t i = 0; i < __argc; i++) { VulkanExample::args.push_back(__argv[i]); };
    vulkanExample = new VulkanExample();
    vulkanExample->initVulkan();
    vulkanExample->setupWindow(hInstance, WndProc);
    vulkanExample->prepare();
    vulkanExample->renderLoop();
    delete(vulkanExample);
    return 0;
}
#elif defined(VK_USE_PLATFORM_ANDROID_KHR)
```

后续可继承重写部分函数。

initVulkan 函数

- 创建 Vulkan 实例：createInstance
- 设置Debugging：setupDebugging
- 查询设置使用的显卡：vkEnumeratePhysicalDevices
- commandLineParser：存储显卡信息
- physicalDevice = physicalDevices[selectedDevice]：默认选择0号位显卡
- vkGetPhysicalDeviceProperties：获取显卡属性
- vkGetPhysicalDeviceFeatures：获取显卡特性
- vkGetPhysicalDeviceMemoryProperties：获取显卡内存
- getEnabledFeatures：(未实现)
- VulkanDevice：创建 VulkanDevice 实例
- vkGetDeviceQueue：Get a graphics queue from the device

- getSupportedDepthFormat：获取支持的深度 format
- swapChain.connect(instance, physicalDevice, device）
- vkCreateSemaphore (presentComplete,renderComplete)

prepare 函数

```
void prepare()
  {
    VulkanExampleBase::prepare();  //继承基类，包括SwapChain，ommandBuffers，RenderPass等的创建
    prepareSynchronizationPrimitives();
    prepareVertices(USE_STAGING);
    prepareUniformBuffers();
    setupDescriptorSetLayout();
    preparePipelines();
    setupDescriptorPool();
    setupDescriptorSet();
    buildCommandBuffers();
    prepared = true;
  }
```