```
VRE::VRE_Window
  - int mWidth

    int mHeight

    bool mFrameBufferResized

    const std::string mWindowName

   GLFWwindow * mWindow
  + VRE Window(int width,
    int height, std::string name)
  + ~VRE_Window()
  + VRE_Window(const VRE
     _Window &)=delete
  + VRE Window & operator
    =(const VRE_Window &)
    =delete
  + bool ShouldClose()
  + void CreateWindowSurface
    (VkInstance instance, VkSurface
    KHR *surface)
  + VkExtent2D GetExtent()
  + bool HasWindowResized()
  + void ResetWindowResizedFlag()
  + GLFWwindow * GetGLFWwindow
    () const
  void Init()

    static void FrameBufferResized

    Callback(GLFWwindow *window,
    int width, int height)
                    -mWindow
          VRE::VRE_Device
 + const bool mEnableValidation
   Lavers

    VkPhysicalDeviceProperties

   mProperties

    VkInstance mInstance

    VkDebugUtilsMessengerEXT

   mDebugMessenger

    VkPhysicalDevice mPhysical

   Device

    VkCommandPool mCommandPool

  VkDevice mVkDevice

    VkSurfaceKHR mVkSurface

    VkQueue mGraphicsQueue

  VkQueue mPresentQueue
 const std::vector
   const char * > mValidation
   Layers
 const std::vector
   const char * > mDeviceExtensions
 + VRE Device(VRE Window
   &window)
 + ~VRE Device()
 + VRE_Device(const VRE
   _Device &)=delete
 + VRE_Device & operator
   =(const VRE_Device &)
   =delete
 + VRE_Device(VRE_Device
   &&)=delete
 + VRE_Device & operator
   =(VRE_Device &&)=delete
 + VkCommandPool GetCommand
   Pool()
 + VkDevice GetVkDevice()

    VkPhysicalDevice GetPhysical

   Device()
 + VkSurfaceKHR Surface()
   and 12 more...

    void CreateInstance()

    void SetupDebugMessenger()

    void CreateSurface()

    void PickPhysicalDevice()

    void CreateLogicalDevice()

    void CreateCommandPool()

  bool IsDeviceSuitable
   (VkPhysicalDevice device)
 std::vector< const</li>
   char * > GetRequiredExtensions()

    bool CheckValidationLayer

   Support()

    QueueFamilyIndices

    FindQueueFamilies(VkPhysical
   Device device)
   void PopulateDebugMessenger
   CreateInfo(VkDebugUtilsMessenger
   CreateInfoEXT &createInfo)
  void HasGflwRequiredInstance
   Extensions()
   bool CheckDeviceExtension
   Support(VkPhysicalDevice
   device)
  SwapChainSupportDetails
   QuerySwapChainSupport
   (VkPhysicalDevice device)
                    -mDevice
          VRE::VRE_Buffer
void * mMapped
```

minOffsetAlignment=1) + VRE\_Buffer(const VRE

VkDeviceSize

instanceCount, VkBufferUsageFlags usageFlags, VkMemoryPropertyFlags

```
=delete

    + VkResult Map(VkDeviceSize

  size=VK_WHOLE_SIZE, VkDeviceSize
```

=(const VRE\_Buffer &)

VkBuffer mBuffer

 VkBufferUsageFlags mUsageFlags

**VkMemoryPropertyFlags** mMemoryPropertyFlags + VRE Buffer(VRE Device &device, VkDeviceSize instanceSize, uint32

memoryPropertyFlags,

+ ~VRE Buffer()

+ void Unmap() + void WriteToBuffer

offset=0)

offset=0)

and 12 more...

\_Buffer &)=delete VRF Ruffer &

 VkDeviceMemory mMemory VkDeviceSize mBufferSize uint32 t mInstanceCount VkDeviceSize mInstanceSize VkDeviceSize mAlignmentSize

- (void \*data, VkDeviceSize size=VK\_WHOLE\_SIZE, VkDeviceSize offset=0) + VkResult Flush(VkDeviceSize
- + VkDescriptorBufferInfo DescriptorInfo(VkDeviceSize size=VK\_WHOLE\_SIZE, VkDeviceSize

size=VK\_WHOLE\_SIZE, VkDeviceSize

- VkResult Invalidate (VkDeviceSize size=VK WHOLE SIZE, VkDeviceSize offset=0)
  - static VkDeviceSize GetAlignment(VkDeviceSize instanceSize, VkDeviceSize minOffsetAlignment)