```
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
        Layers
     + 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_glTFModel

    const std::string mFileFolder

    const std::string mFileName

std::vector< std::shared</li>
  _ptr< VRE_Texture > > mTextures
 std::vector< glTFNode
   ' > mNodes
std::vector< glTFMaterial</li>
  > mMaterials
 std::vector< int32
  _t > mTextureIndices
std::unique_ptr< VRE</li>
  _Buffer > mVertexBuffer
- std::unique ptr< VRE
  _Buffer > mIndexBuffer
 uint32_t mVertexCount
 uint32_t mIndexCount
+ VRE_glTFModel(VRE
  Device &device, const
  std::string &fileFolder,
  const std::string &fileName)
+ ~VRE_glTFModel()
```

VRE glTFModel(c

+ void LoadModel()

commandBuffer)

=delete

VRE\_glTFModel &)=delete
+ VRE\_glTFModel & operator
=(const VRE\_glTFModel &)

+ void Bind(VkCommandBuffer

+ void Render(VkCommandBuffer

 + void RenderNode(VkCommand Buffer commandBuffer, glTFNode

VRE\_DescriptorWriter &writer)

void LoadTextures(tinygltf

void CreateVertexBuffers
(const std::vector< Vertex</pre>

void CreateIndexBuffer
(const std::vector< uint32</pre>

> &vertices)

\_t > &indices)

::Model &model)
void LoadMaterials (tinygltf::Model &model)
void LoadNode(glTFNode \*parent, const tinygltf ::Node &node, const tinygltf ::Model &model, ModelData &data)

commandBuffer, VkPipelineLayout &pipelineLayout, VRE\_DescriptorWriter &writer)

\*node, VkPipelineLayout &pipelineLayout,