```
- 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
                                                                    -mWindow
                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</p>
                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_Renderer
                                              - std::unique_ptr< VRE
                                               _SwapChain > mSwapChain
                                               std::vector< VkCommandBuffer
                                               > mCommandBuffers
                                               uint32 t mCurrentImageIndex
                                              - int mCurrentFrameIndex
                                              - bool mDrawStarted
                                              + VRE_Renderer(VRE_Window
                                                &window, VRE_Device &device)
                                              + ~VRE_Renderer()
                                              + VRE_Renderer(const
                                                VRE_Renderer &)=delete
                                              VRE Renderer & operator
                                               =(const VRE_Renderer &)
                                               =delete
                                              + VkCommandBuffer BeginDraw()
                                 +mDevice
                                              + void EndDraw()
                                              + void BeginSwapChainRender
                                               Pass(VkCommandBuffer commandBuffer)
                                              + void EndSwapChainRenderPass
                                               (VkCommandBuffer commandBuffer)
                                              + bool IsDrawInProgress
                                              + VkRenderPass GetSwapChain
                                               RenderPass() const
                                              + float GetAspectRatio
                                               () const
                                              + VkCommandBuffer GetCurrent
                                               CommandBuffer() const
                                              int GetFrameIndex()
                                                const
                                               void CreateCommandBuffers()
                                              void RecreateSwapChain()
                                              void FreeCommandBuffers()
                                                                  +mRenderer
                                                     VRE::VRE_SharedContext
                                                 + std::shared_ptr< VRE
                                                   DescriptorSetLayout
                                                   > mGlobalDescSetLayout
                                                  + std::vector< std::unique
                                                   _ptr< VRE_DescriptorPool
                                                   > > mObjectDescPools
-mDevice
                                                  + std::vector< VkDescriptor
                                                   Set > mSceneDescriptorSets
                                                 + std::vector< VRE_PointLight
                                                   > mPointLights
                                                 + VRE_GameObject::GameObjects
                                                   Map * mGameObjMap
                                                 + VRE_SharedContext()
                                                 mSharedContext
                                                                  -mSharedContext
                                                                                                          -mSceneContext
                                                                                                                                                      -mRenderer
                                                                                                                                                                                                    VRE::VRE_Camera
                                                                                                                                                                                                glm::vec3 mPosition
        VRE::VRE_GameObjectManager
                                                                                                                                                                                                - glm::vec3 mRotation
      VRE_GameObject::GameObject
                                                                                                                                                                                                glm::mat4 mProjectionMatrix
       ID mLastID
                                                                                                                                                                                                glm::mat4 mViewMatrix
      VRE_GameObject::GameObjects
                                                                                                                                                                                                - glm::mat4 mInvViewMatrix
      Map mGameObjectsMap
                                                                                                                   VRE::VRE_LightRenderSystem
                                                 VRE::VRE_GameObjRenderSystem
                                                                                                                                                                                               + void SetOrthographicProjection
      std::vector< std::unique
                                                                                                                 std::unique_ptr< VRE</li>
                                                - std::unique_ptr< VRE
                                                                                                                                                                                                (float top, float bottom,
       _ptr< VRE_Buffer > > mUboBuffers
                                                                                                                  _Pipeline > mPipeline
                                                 _Pipeline > mPipeline
                                                                                                                                                                                                 float left, float right, float
     + VRE_GameObjectManager
                                                                                                                - VkPipelineLayout mPipeline
                                                                                                                                                                                                 near, float far)
                                                - std::unique_ptr< VRE
       (VRE_SharedContext *sharedContext)
                                                                                                                 Layout

    void SetPerspectiveProjection

                                                 _DescriptorSetLayout
     + ~VRE_GameObjectManager()
                                                                                                                + VRE_LightRenderSystem
                                                                                                                                                                                                (float fovY, float aspectRatio,
                                                 > mDescSetLayout
     + VRE_GameObjectManager
                                                                                                                 (VRE_SharedContext *sceneContext)
                                                                                                                                                                                                 float near, float far)

    VkPipelineLayout mPipeline

       (const VRE_GameObjectManager
                                                                                                                + ~VRE_LightRenderSystem()

    void SetViewDirection

                                                 Layout
       &)=delete
                                                                                                                                                                                                (glm::vec3 position,
                                                                                                                + VRE_LightRenderSystem
                                                + VRE_GameObjRenderSystem
                                                                                                                                                                  VRE::VRE_InputListener
     + VRE_GameObjectManager
                                                                                                                                                                                                 glm::vec3 direction,
                                                                                                                 (const VRE_LightRenderSystem
                                                 (VRE_SharedContext *sharedContext)
                                                                                                                                                                  + float mMoveSpeed
       & operator=(const VRE
                                                                                                                                                                                                 glm::vec3 up=glm::vec3
                                                                                                                  &)=delete
                                                + ~VRE_GameObjRenderSystem()
       _GameObjectManager &)=delete
                                                                                                                                                                 + float mMouseSensitivity
                                                                                                                                                                                                (0.f, 1.f, 0.f)
                                                                                              -mSceneContext
                                                                                                                + VRE_LightRenderSystem
                                                + VRE_GameObjRenderSystem
      + VRE_GameObjectManager
                                                                                                                                                                  void Move(GLFWwindow
                                                                                                                                                                                               + void SetViewTarget
                                                                                                                  & operator=(const VRE
                                                 (const VRE_GameObjRenderSystem
       (VRE_GameObjectManager
                                                                                                                                                                   *window, float dt, VRE
                                                                                                                                                                                                (glm::vec3 position,
                                                                                                                  _LightRenderSystem &)=delete
                                                 &)=delete
       &&)=delete
                                                                                                                                                                    _Camera &camera)
                                                                                                                                                                                                 glm::vec3 target, glm
                                                                                                                + void Init()
                                                + VRE_GameObjRenderSystem
                                                                                                                                                                                                ::vec3 up=glm::vec3(0.f,
      VRE_GameObjectManager
                                                                                                                + void Update(UBO &ubo,
                                                 & operator=(const VRE
                                                                                                                                                                                                 1.f, 0.f))
       & operator=(VRE_GameObject
                                                                                                                  float dt)
                                                 _GameObjRenderSystem &)
       Manager &&)=delete
                                                                                                                                                                                                void SetViewXYZ(glm
                                                 =delete
                                                                                                                + void Render()
                                                                                                                                                                                                ::vec3 position, glm
      + VRE_GameObject & CreateGame
                                                + void Init()
                                                                                                                                                                                                ::vec3 rotation)
       Object()

    void CreatePipelineLayout

                                                                                                                 (VkDescriptorSetLayout
                                                + void Render()
                                                                                                                                                                                                const glm::mat4 GetProjection

    VkDescriptorBufferInfo

                                                                                                                  descSetLayout)
                                                                                                                                                                                                () const
       GetBufferInfoForGameObject

    void CreatePipelineLayouts()

                                                                                                                 void CreatePipeline
       (VRE_GameObject::GameObjectID
                                                                                                                                                                                               + const glm::mat4 GetViewMat

    void CreatePipelines()

                                                                                                                 (VkRenderPass renderPass)
       gObjectID) const
                                                                                                                                                                                                () const
      + void Init()
                                                                                                                                                                                               + const glm::mat4 GetInvView
                                                                                                                                                                                                Mat() const
     + void Update(float deltaTime)
                                                                                                                                                                                               + glm::vec3 GetRotation
     + VRE_GameObject::GameObjects
                                                                                                                                                                                                () const
       Map & GetGameObjectsMap()
                                                                                                                                                                                               + glm::vec3 GetPosition
                                                                                                                                                                                                () const
                                                                                                              -mPLRenderSystem
                                          -mGameObjectManager \ -mGameObjRenderSystem
                                                                                                                                                               -mInputListener
                                                                                                                                                                                         -mCamera
                                                                                    VRE::VRE_App
                                                                          + static const int DEF
                                                                            _WINDOW_WIDTH
                                                                           + static const int DEF
                                                                            _WINDOW_HEIGHT
                                                                           + static const std::string
                                                                            DEF_WINDOW_TITLE
                                                                            std::unique_ptr< VRE
                                                                            _DescriptorPool > mDescriptorPool
                                                                            std::vector< std::unique
                                                                            _ptr< VRE_DescriptorPool
                                                                            > > mObjectPools
                                                                           - std::vector< std::unique</pre>
                                                                            _ptr< VRE_Buffer > > mSceneUBOs
                                                                          + VRE_App()
                                                                          + ~VRE_App()
                                                                          + VRE_App(const VRE
```

VRE::VRE\_Window

int mWidthint mHeight

-mWindow

+ void Init()+ void Update(float dt, UBO &ubo)+ void Render(UBO &ubo)

=(const VRE\_App &)=delete

\_App &)=delete + VRE\_App & operator

void LoadObjects()

+ void Run()