

vengine

0.1.0

Generated by Doxygen 1.9.1



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 ven::Camera Class Reference	5
3.2 myLib::Clock Class Reference	5
3.3 ven::Device Class Reference	6
3.4 ven::Engine Class Reference	6
3.5 ven::KeyboardController Class Reference	7
3.6 ven::KeyboardController::KeyMappings Struct Reference	7
3.7 ven::Model Class Reference	7
3.8 ven::Object Class Reference	8
3.9 ven::PipelineConfigInfo Struct Reference	8
3.10 gui::PluginLoader Class Reference	9
3.11 gui::PluginLoader::PluginLoaderException Class Reference	10
3.12 ven::QueueFamilyIndices Struct Reference	10
3.13 myLib::Random Class Reference	10
3.14 ven::Renderer Class Reference	11
3.15 ven::RenderSystem Class Reference	11
3.16 ven::Shaders Class Reference	11
3.17 ven::SimplePushConstantData Struct Reference	12
3.18 ven::SwapChain Class Reference	12
3.19 ven::SwapChainSupportDetails Struct Reference	13
3.20 myLib::Time Class Reference	13
3.21 ven::Transform3DComponent Struct Reference	13
3.22 ven::Model::Vertex Struct Reference	14
3.23 ven::Window Class Reference	14
<b>Index</b>	<b>15</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ven::Camera . . . . .	5
myLib::Clock . . . . .	5
ven::Device . . . . .	6
ven::Engine . . . . .	6
std::exception	
gui::PluginLoader::PluginLoaderException . . . . .	10
ven::KeyboardController . . . . .	7
ven::KeyboardController::KeyMappings . . . . .	7
ven::Model . . . . .	7
ven::Object . . . . .	8
ven::PipelineConfigInfo . . . . .	8
gui::PluginLoader . . . . .	9
ven::QueueFamilyIndices . . . . .	10
myLib::Random . . . . .	10
ven::Renderer . . . . .	11
ven::RenderSystem . . . . .	11
ven::Shaders . . . . .	11
ven::SimplePushConstantData . . . . .	12
ven::SwapChain . . . . .	12
ven::SwapChainSupportDetails . . . . .	13
myLib::Time . . . . .	13
ven::Transform3DComponent . . . . .	13
ven::Model::Vertex . . . . .	14
ven::Window . . . . .	14



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">ven::Camera</a>	5
<a href="#">myLib::Clock</a>	5
<a href="#">ven::Device</a>	6
<a href="#">ven::Engine</a>	6
<a href="#">ven::KeyboardController</a>	7
<a href="#">ven::KeyboardController::KeyMappings</a>	7
<a href="#">ven::Model</a>	7
<a href="#">ven::Object</a>	8
<a href="#">ven::PipelineConfigInfo</a>	8
<a href="#">gui::PluginLoader</a>	9
<a href="#">gui::PluginLoader::PluginLoaderException</a>	10
<a href="#">ven::QueueFamilyIndices</a>	10
<a href="#">myLib::Random</a>	10
<a href="#">ven::Renderer</a>	11
<a href="#">ven::RenderSystem</a>	11
<a href="#">ven::Shaders</a>	11
<a href="#">ven::SimplePushConstantData</a>	12
<a href="#">ven::SwapChain</a>	12
<a href="#">ven::SwapChainSupportDetails</a>	13
<a href="#">myLib::Time</a>	13
<a href="#">ven::Transform3DComponent</a>	13
<a href="#">ven::Model::Vertex</a>	14
<a href="#">ven::Window</a>	14





## Chapter 3

# Class Documentation

### 3.1 ven::Camera Class Reference

#### Public Member Functions

- void **setOrthographicProjection** (float left, float right, float top, float bottom, float near, float far)
- void **setPerspectiveProjection** (float fovy, float aspect, float near, float far)
- void **setViewDirection** (glm::vec3 position, glm::vec3 direction, glm::vec3 up=glm::vec3{0.F, -1.F, 0.F})
- void **setViewTarget** (glm::vec3 position, glm::vec3 target, glm::vec3 up=glm::vec3{0.F, -1.F, 0.F})
- void **setViewXYZ** (glm::vec3 position, glm::vec3 rotation)
- const glm::mat4 & **getProjection** () const
- const glm::mat4 & **getView** () const

The documentation for this class was generated from the following file:

- include/VEngine/Camera.hpp

### 3.2 myLib::Clock Class Reference

#### Public Member Functions

- void **restart** ()
- void **pause** ()
- void **resume** ()
- [Time](#) **getElapsedTime** () const

The documentation for this class was generated from the following file:

- lib/static/myLib/include/myLib/Clock/Clock.hpp

### 3.3 ven::Device Class Reference

#### Public Member Functions

- **Device** ([ven::Window](#) &>window)
- **Device** (const [Device](#) &)=delete
- [Device](#) & **operator=** (const [Device](#) &)=delete
- **Device** ([Device](#) &&)=delete
- [Device](#) & **operator=** ([Device](#) &&)=delete
- VkCommandPool **getCommandPool** ()
- VkDevice **device** ()
- VkSurfaceKHR **surface** ()
- VkQueue **graphicsQueue** ()
- VkQueue **presentQueue** ()
- [SwapChainSupportDetails](#) **getSwapChainSupport** ()
- uint32\_t **findMemoryType** (uint32\_t typeFilter, VkMemoryPropertyFlags properties)
- [QueueFamilyIndices](#) **findPhysicalQueueFamilies** ()
- VkFormat **findSupportedFormat** (const std::vector< VkFormat > &candidates, VkImageTiling tiling, VkFormatFeatureFlags features)
- void **createBuffer** (VkDeviceSize size, VkBufferUsageFlags usage, VkMemoryPropertyFlags properties, VkBuffer &buffer, VkDeviceMemory &bufferMemory)
- VkCommandBuffer **beginSingleTimeCommands** ()
- void **endSingleTimeCommands** (VkCommandBuffer commandBuffer)
- void **copyBuffer** (VkBuffer srcBuffer, VkBuffer dstBuffer, VkDeviceSize size)
- void **copyBufferToImage** (VkBuffer buffer, VkImage image, uint32\_t width, uint32\_t height, uint32\_t layer↔Count)
- void **createImageWithInfo** (const VkImageCreateInfo &imageInfo, VkMemoryPropertyFlags properties, VkImage &image, VkDeviceMemory &imageMemory)

#### Public Attributes

- const bool **enableValidationLayers** = true
- VkPhysicalDeviceProperties **m\_properties**

The documentation for this class was generated from the following file:

- include/VEngine/Device.hpp

### 3.4 ven::Engine Class Reference

#### Public Member Functions

- **Engine** (int width=DEFAULT\_WIDTH, int height=DEFAULT\_HEIGHT, const std::string &title="VEngine")
- **Engine** (const [Engine](#) &)=delete
- [Engine](#) **operator=** (const [Engine](#) &)=delete
- [Window](#) & **getWindow** ()
- void **mainLoop** ()

The documentation for this class was generated from the following file:

- include/VEngine/Engine.hpp

## 3.5 ven::KeyboardController Class Reference

### Classes

- struct [KeyMappings](#)

### Public Member Functions

- void **moveInPlaneXZ** (GLFWwindow \*window, float dt, [Object](#) &object)

### Public Attributes

- [KeyMappings](#) **m\_keys** {}
- float **m\_moveSpeed** {3.F}
- float **m\_lookSpeed** {1.5F}

The documentation for this class was generated from the following file:

- include/VEngine/KeyboardController.hpp

## 3.6 ven::KeyboardController::KeyMappings Struct Reference

### Public Attributes

- int **moveLeft** = GLFW\_KEY\_A
- int **moveRight** = GLFW\_KEY\_D
- int **moveForward** = GLFW\_KEY\_W
- int **moveBackward** = GLFW\_KEY\_S
- int **moveUp** = GLFW\_KEY\_SPACE
- int **moveDown** = GLFW\_KEY\_LEFT\_SHIFT
- int **lookLeft** = GLFW\_KEY\_LEFT
- int **lookRight** = GLFW\_KEY\_RIGHT
- int **lookUp** = GLFW\_KEY\_UP
- int **lookDown** = GLFW\_KEY\_DOWN

The documentation for this struct was generated from the following file:

- include/VEngine/KeyboardController.hpp

## 3.7 ven::Model Class Reference

### Classes

- struct [Vertex](#)

## Public Member Functions

- **Model** ([Device](#) &device, const std::vector< [Vertex](#) > &vertices)
- **Model** (const [Model](#) &)=delete
- void **operator=** (const [Model](#) &)=delete
- void **bind** (VkCommandBuffer commandBuffer)
- void **draw** (VkCommandBuffer commandBuffer)

The documentation for this class was generated from the following file:

- include/VEngine/Model.hpp

## 3.8 ven::Object Class Reference

### Public Member Functions

- **Object** (const [Object](#) &)=delete
- [Object](#) & **operator=** (const [Object](#) &)=delete
- **Object** ([Object](#) &&)=default
- [Object](#) & **operator=** ([Object](#) &&)=default
- id\_t **getId** () const

### Static Public Member Functions

- static [Object](#) **createObject** ()

### Public Attributes

- std::shared\_ptr< [ven::Model](#) > **model** {}
- glm::vec3 **color** {}
- [Transform3DComponent](#) **transform3D** {}

The documentation for this class was generated from the following file:

- include/VEngine/Object.hpp

## 3.9 ven::PipelineConfigInfo Struct Reference

### Public Member Functions

- **PipelineConfigInfo** (const [PipelineConfigInfo](#) &)=delete
- [PipelineConfigInfo](#) & **operator=** (const [PipelineConfigInfo](#) &)=delete

## Public Attributes

- VkPipelineInputAssemblyStateCreateInfo **inputAssemblyInfo** {}
- VkPipelineRasterizationStateCreateInfo **rasterizationInfo** {}
- VkPipelineMultisampleStateCreateInfo **multisampleInfo** {}
- VkPipelineColorBlendAttachmentState **colorBlendAttachment** {}
- VkPipelineColorBlendStateCreateInfo **colorBlendInfo** {}
- VkPipelineDepthStencilStateCreateInfo **depthStencilInfo** {}
- std::vector< VkDynamicState > **dynamicStateEnables**
- VkPipelineDynamicStateCreateInfo **dynamicStateInfo** {}
- VkPipelineLayout **pipelineLayout** = nullptr
- VkRenderPass **renderPass** = nullptr
- uint32\_t **subpass** = 0

The documentation for this struct was generated from the following file:

- include/VEngine/Shaders.hpp

## 3.10 gui::PluginLoader Class Reference

### Classes

- class [PluginLoaderException](#)

### Public Types

- using **PluginCreator** = std::unique\_ptr< IPlugin >(\*)()

### Public Member Functions

- template<typename T >  
std::unique\_ptr< T > **getPlugin** (const std::string &pluginName)
- void **closePlugins** ()

### Static Public Member Functions

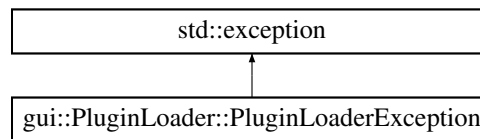
- static [PluginLoader](#) & **getInstance** ()

The documentation for this class was generated from the following file:

- include/VEngine/PluginLoader.hpp

### 3.11 gui::PluginLoader::PluginLoaderException Class Reference

Inheritance diagram for gui::PluginLoader::PluginLoaderException:



#### Public Member Functions

- **PluginLoaderException** (std::string msg)
- const char \* **what** () const noexcept override

The documentation for this class was generated from the following file:

- include/VEngine/PluginLoader.hpp

### 3.12 ven::QueueFamilyIndices Struct Reference

#### Public Member Functions

- bool **isComplete** () const

#### Public Attributes

- uint32\_t **graphicsFamily** {}
- uint32\_t **presentFamily** {}
- bool **graphicsFamilyHasValue** = false
- bool **presentFamilyHasValue** = false

The documentation for this struct was generated from the following file:

- include/VEngine/Device.hpp

### 3.13 myLib::Random Class Reference

#### Static Public Member Functions

- static int **randomInt** (int min, int max)
- static int **randomInt** ()
- static float **randomFloat** (float min, float max)
- static float **randomFloat** ()

The documentation for this class was generated from the following file:

- lib/static/myLib/include/myLib/Random.hpp

## 3.14 ven::Renderer Class Reference

### Public Member Functions

- **Renderer** ([Window](#) &>window, [Device](#) &device)
- **Renderer** (const [Renderer](#) &)=delete
- [Renderer](#) & **operator=** (const [Renderer](#) &)=delete
- VkRenderPass **getSwapChainRenderPass** () const
- float **getAspectRatio** () const
- bool **isFrameInProgress** () const
- VkCommandBuffer **getCurrentCommandBuffer** () const
- int **getFrameIndex** () const
- VkCommandBuffer **beginFrame** ()
- void **endFrame** ()
- void **beginSwapChainRenderPass** (VkCommandBuffer commandBuffer)

### Static Public Member Functions

- static void **endSwapChainRenderPass** (VkCommandBuffer commandBuffer)

The documentation for this class was generated from the following file:

- include/VEngine/Renderer.hpp

## 3.15 ven::RenderSystem Class Reference

### Public Member Functions

- **RenderSystem** ([Device](#) &device, VkRenderPass renderPass)
- **RenderSystem** (const [RenderSystem](#) &)=delete
- [RenderSystem](#) & **operator=** (const [RenderSystem](#) &)=delete
- void **renderObjects** (VkCommandBuffer commandBuffer, std::vector< [ven::Object](#) > &objects, const [Camera](#) &camera)

The documentation for this class was generated from the following file:

- include/VEngine/RenderSystem.hpp

## 3.16 ven::Shaders Class Reference

### Public Member Functions

- **Shaders** ([Device](#) &device, const std::string &vertFilepath, const std::string &fragFilepath, const [PipelineConfigInfo](#) &configInfo)
- **Shaders** (const [Shaders](#) &)=delete
- [Shaders](#) & **operator=** (const [Shaders](#) &)=delete
- void **bind** (VkCommandBuffer commandBuffer)

## Static Public Member Functions

- static void **defaultPipelineConfigInfo** ([PipelineConfigInfo](#) &configInfo)

The documentation for this class was generated from the following file:

- include/VEngine/Shaders.hpp

## 3.17 ven::SimplePushConstantData Struct Reference

### Public Attributes

- glm::mat4 **transform** {1.F}
- glm::vec3 **color**

The documentation for this struct was generated from the following file:

- include/VEngine/RenderSystem.hpp

## 3.18 ven::SwapChain Class Reference

### Public Member Functions

- **SwapChain** ([Device](#) &deviceRef, VkExtent2D windowExtent)
- **SwapChain** ([Device](#) &deviceRef, VkExtent2D windowExtent, std::shared\_ptr< [SwapChain](#) > previous)
- **SwapChain** (const [SwapChain](#) &)=delete
- [SwapChain](#) & **operator=** (const [SwapChain](#) &)=delete
- VkFramebuffer **getFrameBuffer** (unsigned long index)
- VkRenderPass **getRenderPass** ()
- VkImageView **getImageView** (int index)
- size\_t **imageCount** ()
- VkFormat **getSwapChainImageFormat** ()
- VkExtent2D **getSwapChainExtent** ()
- uint32\_t **width** () const
- uint32\_t **height** () const
- float **extentAspectRatio** () const
- VkFormat **findDepthFormat** ()
- VkResult **acquireNextImage** (uint32\_t \*imageIndex)
- VkResult **submitCommandBuffers** (const VkCommandBuffer \*buffers, const uint32\_t \*imageIndex)
- bool **compareSwapFormats** (const [SwapChain](#) &swapChainp) const

### Static Public Attributes

- static constexpr int **MAX\_FRAMES\_IN\_FLIGHT** = 2

The documentation for this class was generated from the following file:

- include/VEngine/SwapChain.hpp



## 3.19 ven::SwapChainSupportDetails Struct Reference

### Public Attributes

- VkSurfaceCapabilitiesKHR **capabilities**
- std::vector< VkSurfaceFormatKHR > **formats**
- std::vector< VkPresentModeKHR > **presentModes**

The documentation for this struct was generated from the following file:

- include/VEngine/Device.hpp

## 3.20 myLib::Time Class Reference

### Public Member Functions

- **Time** (const double seconds)
- int **asSeconds** () const
- int **asMilliseconds** () const
- int **asMicroseconds** () const

The documentation for this class was generated from the following file:

- lib/static/myLib/include/myLib/Clock/Time.hpp

## 3.21 ven::Transform3DComponent Struct Reference

### Public Member Functions

- glm::mat4 **mat4** () const

### Public Attributes

- glm::vec3 **translation** {}
- glm::vec3 **scale** {1.F, 1.F, 1.F}
- glm::vec3 **rotation** {}

The documentation for this struct was generated from the following file:

- include/VEngine/Object.hpp

## 3.22 ven::Model::Vertex Struct Reference

### Static Public Member Functions

- static std::vector< VkVertexInputBindingDescription > **getBindingDescriptions** ()
- static std::vector< VkVertexInputAttributeDescription > **getAttributeDescriptions** ()

### Public Attributes

- glm::vec3 **position**
- glm::vec3 **color**

The documentation for this struct was generated from the following file:

- include/VEngine/Model.hpp

## 3.23 ven::Window Class Reference

### Public Member Functions

- **Window** (int width, int height, const std::string &title)
- GLFWwindow \* **createWindow** (int width, int height, const std::string &title)
- void **createWindowSurface** (VkInstance instance, VkSurfaceKHR \*surface)
- GLFWwindow \* **getGLFWWindow** () const
- VkExtent2D **getExtent** () const
- bool **wasWindowResized** () const
- void **resetWindowResizedFlag** ()

The documentation for this class was generated from the following file:

- include/VEngine/Window.hpp

# Index

`gui::PluginLoader`, [9](#)  
`gui::PluginLoader::PluginLoaderException`, [10](#)

`myLib::Clock`, [5](#)  
`myLib::Random`, [10](#)  
`myLib::Time`, [13](#)

`ven::Camera`, [5](#)  
`ven::Device`, [6](#)  
`ven::Engine`, [6](#)  
`ven::KeyboardController`, [7](#)  
`ven::KeyboardController::KeyMappings`, [7](#)  
`ven::Model`, [7](#)  
`ven::Model::Vertex`, [14](#)  
`ven::Object`, [8](#)  
`ven::PipelineConfigInfo`, [8](#)  
`ven::QueueFamilyIndices`, [10](#)  
`ven::Renderer`, [11](#)  
`ven::RenderSystem`, [11](#)  
`ven::Shaders`, [11](#)  
`ven::SimplePushConstantData`, [12](#)  
`ven::SwapChain`, [12](#)  
`ven::SwapChainSupportDetails`, [13](#)  
`ven::Transform3DComponent`, [13](#)  
`ven::Window`, [14](#)