Vulkan Renderer

v0.1.0

Generated by Doxygen 1.9.8

1 Namespace Index 1

1.1 Namespace List . . . 1

2 Class Index 3

2.1 Class List . . . 3

3 File Index 5

3.1 File List . . . 5

4 Namespace Documentation 7

4.1 vkr Namespace Reference . . . 7

4.1.1 Detailed Description . . . 7

4.2 vkr::core Namespace Reference . . . 7

4.2.1 Detailed Description . . . 7

4.3 vkr::rendering Namespace Reference . . . 7

4.3.1 Detailed Description . . . 8

4.4 vkr::utils Namespace Reference . . . 8

4.4.1 Detailed Description . . . 8

5 Class Documentation 9

5.1 vke::Application Class Reference . . . 9

5.2 vke::core::Instance Class Reference . . . 9

5.3 vke::Window Class Reference . . . 9

6 File Documentation 11

6.1 Application.h . . . 11

6.2 Window.h . . . 11

6.3 Instance.h . . . 12

6.4 pch.h . . . 13

6.5 utils.h . . . 13

Index 15

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

vkr
Something 7
vkr::core
Something 7
vkr::rendering
Something 7
vkr::utils
Something 8

Chapter 2

Class Index

2.1 Class List

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

vke::Application 9
vke::core::Instance 9
vke::Window 9

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

C:/Users/Joshu/CLionProjects/VulkanRenderer/src/pch.h 13
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Application.h 11
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Window.h 11
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/core/Instance.h 12
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/utils/utils.h 13

Chapter 4

Namespace Documentation

4.1 vkr Namespace Reference

something

Namespaces

- namespace core
  something
- namespace rendering
  something
- namespace utils
  something

4.1.1 Detailed Description

something

4.2 vkr::core Namespace Reference

something

4.2.1 Detailed Description

something

4.3 vkr::rendering Namespace Reference

something

4.3.1 Detailed Description

something

4.4 vkr::utils Namespace Reference

something

4.4.1 Detailed Description

something

Chapter 5

Class Documentation

5.1 vke::Application Class Reference

Public Member Functions

- **Application** (const std::string &)
- **Application** (const Application &)=delete
- **Application** (Application &&)=delete
- Application & **operator=** (const Application &)=delete
- Application & **operator=** (Application &&)=delete
- void **run** ()

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

- C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Application.h
- C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Application.cpp

5.2 vke::core::Instance Class Reference

Public Member Functions

- **Instance** (const std::string &, const std::vector< const char * > &={}, const std::vector< const char * > &={})
- **Instance** (const Instance &)=delete
- **Instance** (Instance &&)=delete
- Instance & **operator=** (const Instance &)=delete
- Instance & **operator=** (Instance &&)=delete

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

- C:/Users/Joshu/CLionProjects/VulkanRenderer/src/core/Instance.h
- C:/Users/Joshu/CLionProjects/VulkanRenderer/src/core/Instance.cpp

5.3 vke::Window Class Reference

Public Member Functions

- void **update** ()
- bool **isClosed** () const

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

- C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Window.h
- C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Window.cpp

Chapter 6

File Documentation

6.1 Application.h

```
00001 //
00002 // Created by Joshu on 10/30/2023.
00003 //
00004
00005 #ifndef VULKANRENDERER_APPLICATION_H
00006 #define VULKANRENDERER_APPLICATION_H
00007
00008 #include "Window.h"
00009 #include "../core/Instance.h"
00010
00011
00013 namespace vkr {}
00014
00016 namespace vkr::core {}
00017
00019 namespace vkr::utils {}
00020
00022 namespace vkr::rendering {}
00023
00024
00025 namespace vke {
00026     class Application {
00027     public:
00028         Application() = default;
00029
00030         explicit Application(const std::string &);
00031
00032         ~Application() = default;
00033
00034         Application(const Application &) = delete;
00035
00036         Application(Application &&) = delete;
00037
00038         Application &operator=(const Application &) = delete;
00039
00040         Application &operator=(Application &&) = delete;
00041
00042
00043         void run();
00044
00045     private:
00046         void initVulkan();
00047
00048         Window window{};
00049         core::Instance instance{""};
00050     };
00051 } // namespace vke
00052
00053
00054 #endif //VULKANRENDERER_APPLICATION_H
```

6.2 Window.h

```
00001 //
```

```
00002 // Created by Joshu on 10/31/2023.
00003 //
00004
00005 #ifndef VULKANRENDERER_WINDOW_H
00006 #define VULKANRENDERER_WINDOW_H
00007
00008 #include "GLFW/glfw3.h"
00009
00010 namespace vke {
00011     class Window {
00012     public:
00013         Window();
00014
00015         ~Window();
00016
00017         void update();
00018
00019         [[nodiscard]] inline bool isClosed() const { return glfwWindowShouldClose(handle); };
00020
00021     private:
00022         GLFWwindow *handle = nullptr;
00023
00024         static constexpr int width{800};
00025         static constexpr int height{600};
00026     };
00027 } // namespace vke
00028
00029
00030 #endif //VULKANRENDERER_WINDOW_H
```

6.3 Instance.h

```
00001 //
00002 // Created by Joshu on 11/1/2023.
00003 //
00004
00005 #ifndef VULKANRENDERER_INSTANCE_H
00006 #define VULKANRENDERER_INSTANCE_H
00007
00008
00009 namespace vke::core {
00010     class Instance {
00011     public:
00012         explicit Instance(const std::string &, const std::vector<const char *> & = {},
00013                           const std::vector<const char *> & = {});
00014
00015         Instance(const Instance &) = delete;
00016
00017         Instance(Instance &&) = delete;
00018
00019         ~Instance() = default;
00020
00021         Instance &operator=(const Instance &) = delete;
00022
00023         Instance &operator=(Instance &&) = delete;
00024
00025     private:
00026
00027         bool enableExtension(const char *);
00028
00029         bool enableLayer(const char *);
00030
00031         void validateExtensions(const std::vector<const char *> &);
00032
00033         void validateLayers(const std::vector<const char *> &);
00034
00035         std::vector<const char *> enabledExtensions;
00036         std::vector<vk::ExtensionProperties> availableExtensions;
00037
00038         std::vector<const char *> enabledLayers;
00039         std::vector<vk::LayerProperties> availableLayers;
00040
00041         vk::raii::Context context{};
00042         vk::raii::Instance handle{nullptr};
00043         vk::raii::DebugUtilsMessengerEXT debugMessenger{nullptr};
00044
00045     };
00046 } // namespace vke::core
00047
00048
00049
00050 #endif //VULKANRENDERER_INSTANCE_H
```

6.4 pch.h

```
00001 //
00002 // Created by Joshu on 10/30/2023.
00003 //
00004
00005 #ifndef VULKANRENDERER_PCH_H
00006 #define VULKANRENDERER_PCH_H
00007
00008 #include <iostream>
00009 #include <algorithm>
00010 #include <functional>
00011 #include <utility>
00012
00013 #include <string>
00014 #include <sstream>
00015 #include <vector>
00016 #include <unordered_map>
00017 #include <unordered_set>
00018
00019 #include "spdlog/spdlog.h"
00020
00021 #define VULKAN_HPP_DISPATCH_LOADER_DYNAMIC 1
00022 #define VULKAN_HPP_NO_CONSTRUCTORS
00023
00024 #include <vulkan/vulkan.hpp>
00025 #include <vulkan/vulkan_raii.hpp>
00026
00027 #endif //VULKANRENDERER_PCH_H
```

6.5 utils.h

```
00001 //
00002 // Created by Joshu on 11/1/2023.
00003 //
00004
00005 #ifndef VULKANRENDERER_UTILS_H
00006 #define VULKANRENDERER_UTILS_H
00007
00008 #include "../pch.h"
00009
00010 namespace vke::utils {
00011     vk::DebugUtilsMessengerCreateInfoEXT
00012     createDebugMessengerInfo(VKAPI_ATTR VkBool32 VKAPI_CALL
      (*func)(VkDebugUtilsMessageSeverityFlagBitsEXT,
00013                                                                VkDebugUtilsMessageTypeFlagsEXT,
00014
      VkDebugUtilsMessengerCallbackDataEXT const *,
00015                                                                void * /*pUserData*/)) {
00016       return vk::DebugUtilsMessengerCreateInfoEXT{
00017         .messageSeverity =
00018         vk::DebugUtilsMessageSeverityFlagBitsEXT::eWarning |
00019         vk::DebugUtilsMessageSeverityFlagBitsEXT::eError,
00020         .messageType =
00021         vk::DebugUtilsMessageTypeFlagBitsEXT::eGeneral |
00022         vk::DebugUtilsMessageTypeFlagBitsEXT::ePerformance |
00023         vk::DebugUtilsMessageTypeFlagBitsEXT::eValidation,
00024         .pfnUserCallback = func,
00025       };
00026     }
00027 }
00028
00029 #endif //VULKANRENDERER_UTILS_H
```

Index

C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Application.h, 11
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/application/Window.h, 11
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/core/Instance.h, 12
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/pch.h, 13
C:/Users/Joshu/CLionProjects/VulkanRenderer/src/utils/utils.h, 13

vke::Application, 9
vke::core::Instance, 9
vke::Window, 9
vkr, 7
vkr::core, 7
vkr::rendering, 7
vkr::utils, 8