GLWrapper

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 GLWrapper Namespace Reference

Classes

- class GLWindow

  Class GLWindow handles the OpenGL window creation.
- class GLShader

  Class GLShader handles shader functionality.

Chapter 5

Class Documentation

5.1 GLWrapper::GLShader Class Reference

Class GLShader handles shader functionality.

```
#include <GLWrapper.h>
```

Public Member Functions

- GLShader (float r, float g, float b, float a)

    Construct a GLShader object with a specified color.
- GLShader ()

    Construct a GLShader object with default color(Black).
- ∼GLShader ()

    Destroy the GLShader object.
- void polygon ()

    To Draw the wireframe polygon.
- void setBackgroundColor (float r, float g, float b, float a)

    Set the background color.
- void drawTriangle ()

    Draw the triangle.

Private Member Functions

- void loadGlad ()

    Load Glad OpenGL function pointers.
- void run (float r, float g, float b, float a)

    Render with the specified colors.
- void vertex ()

    Define the vertices.

Private Attributes

- unsigned int shaderProgram
- unsigned int VAO
- unsigned int VBO

5.1.1 Detailed Description

Class GLShader handles shader functionality.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 GLShader() [1/2]

```
GLWrapper::GLShader::GLShader (
                float r,
                float g,
                float b,
                float a )
```

Construct a GLShader object with a specified color.

Parameters

r	Red component of the color.
g	Green component of the color.
b	Blue component of the color.
a	Alpha component of the color.

5.1.2.2 GLShader() [2/2]

```
GLWrapper::GLShader::GLShader ( )
```

Construct a GLShader object with default color(Black).

5.1.2.3 ∼GLShader()

```
GLWrapper::GLShader::∼GLShader ( )
```

Destroy the GLShader object.

5.1.3 Member Function Documentation

5.1.3.1 drawTriangle()

```
void GLWrapper::GLShader::drawTriangle ( )
```

Draw the triangle.

5.1.3.2 loadGlad()

```
void GLWrapper::GLShader::loadGlad ( )  [private]
```

Load Glad OpenGL function pointers.

5.1.3.3 polygon()

```
void GLWrapper::GLShader::polygon ( )
```

To Draw the wireframe polygon.

5.1.3.4 run()

```
void GLWrapper::GLShader::run (
             float r,
             float g,
             float b,
             float a )  [private]
```

Render with the specified colors.

Parameters

r	Red component of the color.
g	Green component of the color.
b	Blue component of the color.
a	Alpha component of the color.

5.1.3.5 setBackgroundColor()

```
void GLWrapper::GLShader::setBackgroundColor (
             float r,
             float g,
             float b,
             float a )
```

Set the background color.

Parameters

r	Red component of the color.
g	Green component of the color.
b	Blue component of the color.
a	Alpha component of the color.

5.1.3.6 vertex()

```
void GLWrapper::GLShader::vertex ( )  [private]
```

Define the vertices.

set up vertex data (and buffer(s)) and configure vertex attributes.

5.1.4 Member Data Documentation

5.1.4.1 shaderProgram

```
unsigned int GLWrapper::GLShader::shaderProgram  [private]
```

5.1.4.2 VAO

```
unsigned int GLWrapper::GLShader::VAO  [private]
```

5.1.4.3 VBO

```
unsigned int GLWrapper::GLShader::VBO  [private]
```

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

- GLWrapper.h
- GLWrapper.cpp

5.2 GLWrapper::GLWindow Class Reference

Class GLWindow handles the OpenGL window creation.

```
#include <GLWrapper.h>
```

Public Member Functions

- GLWindow (unsigned int width, unsigned int height, const char ∗title)

  Construct a GLWindow object.
- GLWindow ()

  Construct a GLWindow object with default values.
- ∼GLWindow ()

  Destroy GLWindow object.
- int isClosed ()

  Checks if the window is closed.
- void processInput ()

  Process all input.
- void swapBufferAndPollEvent ()

  Swap buffers and poll IO events.

Private Member Functions

- void initializeGlfw ()

  Initialize GLFW and configure its properties.
- GLFWwindow ∗ createWindow (unsigned int width, unsigned int height, const char ∗title)

  Create a new window.

Static Private Member Functions

- static void framebuffer_size_callback (GLFWwindow ∗window, int width, int height)

  callback function whenever window size changes (by OS or user resize), this callback function executes.

Private Attributes

- GLFWwindow ∗ window

5.2.1 Detailed Description

Class GLWindow handles the OpenGL window creation.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 GLWindow() [1/2]

```
GLWrapper::GLWindow::GLWindow (
            unsigned int width,
            unsigned int height,
            const char ∗ title )
```

Construct a GLWindow object.

Parameters

width	Width of the window.
height	Height of the window.
title	Title of the window.

5.2.2.2 GLWindow() [2/2]

```
GLWrapper::GLWindow::GLWindow ( )
```

Construct a GLWindow object with default values.

Default Values : width:800, height:600, title:"Hello Triangle"

5.2.2.3 ∼GLWindow()

```
GLWrapper::GLWindow::∼GLWindow ( )
```

Destroy GLWindow object.

5.2.3 Member Function Documentation

5.2.3.1 createWindow()

```
GLFWwindow ∗ GLWrapper::GLWindow::createWindow (
              unsigned int width,
              unsigned int height,
              const char ∗ title )  [private]
```

Create a new window.

Parameters

width	Width of the window.
height	Height of the window.
title	Title of the window.

Returns

GLFWwindow∗ Pointer to the created GLFW window.

5.2.3.2 framebuffer_size_callback()

```
void GLWrapper::GLWindow::framebuffer_size_callback (
            GLFWwindow * window,
            int width,
            int height )  [static], [private]
```

callback function whenever window size changes (by OS or user resize), this callback function executes.

Parameters

window	Pointer to the window.
width	New width of the viewport.
height	New height of the viewport.

5.2.3.3 initializeGlfw()

```
void GLWrapper::GLWindow::initializeGlfw ( )  [private]
```

Initialize GLFW and configure its properties.

5.2.3.4 isClosed()

```
int GLWrapper::GLWindow::isClosed ( )
```

Checks if the window is closed.

Returns

Returns 1 if the window is closed, 0 otherwise.

5.2.3.5 processInput()

```
void GLWrapper::GLWindow::processInput ( )
```

Process all input.

Query Window(GLFW) for relevant key presses/releases and reacts accordingly.

5.2.3.6 swapBufferAndPollEvent()

```
void GLWrapper::GLWindow::swapBufferAndPollEvent ( )
```

Swap buffers and poll IO events.

This function swaps the front and back buffers of the window and processes IO events such as keys pressed/released, mouse moved, etc.

5.2.4 Member Data Documentation

5.2.4.1 window

```
GLFWwindow* GLWrapper::GLWindow::window  [private]
```

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

- GLWrapper.h
- GLWrapper.cpp

Chapter 6

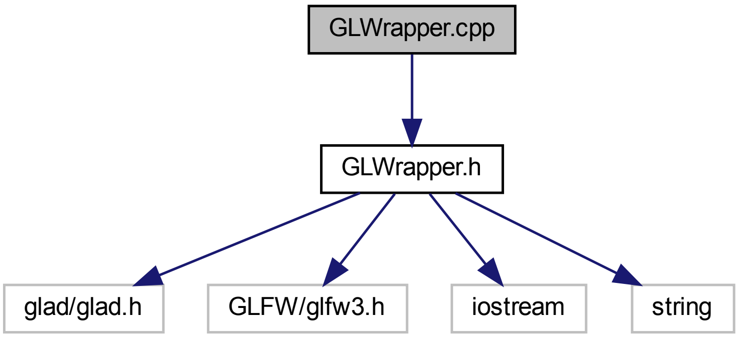
File Documentation

6.1 GLWrapper.cpp File Reference

Implementation of GLWrapper.

```
#include "GLWrapper.h"
```

Include dependency graph for GLWrapper.cpp:



Namespaces

- GLWrapper

6.1.1 Detailed Description

Implementation of GLWrapper.

Author

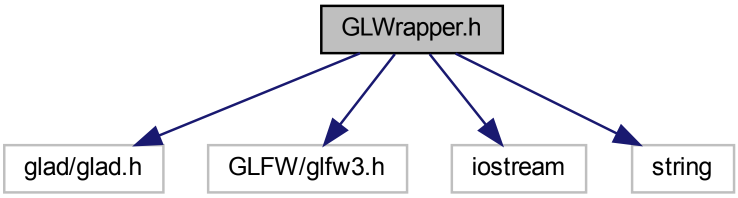
Niva

6.2 GLWrapper.h File Reference

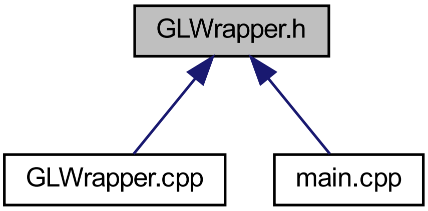
A simple wrapper for OpenGL functions in C++ to implement same functionalities as 'Hello Triangle' example.

```
#include <glad/glad.h>
#include <GLFW/glfw3.h>
#include <iostream>
#include <string>
```

Include dependency graph for GLWrapper.h:



This graph shows which files directly or indirectly include this file:



Classes

- class GLWrapper::GLWindow

  Class GLWindow handles the OpenGL window creation.
- class GLWrapper::GLShader

  Class GLShader handles shader functionality.

Namespaces

- GLWrapper

6.2.1 Detailed Description

A simple wrapper for OpenGL functions in C++ to implement same functionalities as 'Hello Triangle' example.

Author

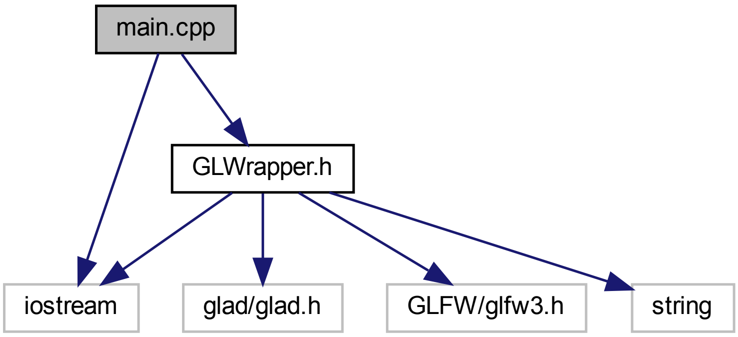
Niva

6.3 main.cpp File Reference

Application using GLWrapper - initializes the GLFW window, sets up shaders, and draws the triangle.

```
#include <iostream>
#include "GLWrapper.h"
```

Include dependency graph for main.cpp:



Functions

- int main ()

6.3.1 Detailed Description

Application using GLWrapper - initializes the GLFW window, sets up shaders, and draws the triangle.

Author

Niva

6.3.2 Function Documentation

6.3.2.1 main()

```
int main ( )
```