

CG1-Tutorium

Erzeugt von Doxygen 1.8.8

Mit Dez 17 2014 12:38:44

Inhaltsverzeichnis

1	Hierarchie-Verzeichnis	1
1.1	Klassenhierarchie	1
2	Klassen-Verzeichnis	3
2.1	Auflistung der Klassen	3
3	Datei-Verzeichnis	5
3.1	Auflistung der Dateien	5
4	Klassen-Dokumentation	7
4.1	BasicObject Klassenreferenz	7
4.1.1	Ausführliche Beschreibung	8
4.1.2	Beschreibung der Konstruktoren und Destruktoren	8
4.1.2.1	BasicObject	8
4.1.3	Dokumentation der Elementfunktionen	8
4.1.3.1	allocateMemory	8
4.1.3.2	getNumberOfVertices	8
4.1.4	Dokumentation der Datenelemente	8
4.1.4.1	numVertsPerPrimitive	9
4.1.4.2	typeOfPrimitive	9
4.1.4.3	vaolD	9
4.1.4.4	vbolD	9
4.1.4.5	vertices	9
4.2	Box Klassenreferenz	9
4.2.1	Ausführliche Beschreibung	10
4.2.2	Beschreibung der Konstruktoren und Destruktoren	10
4.2.2.1	Box	10
4.3	BasicObject::vertexAttributes Strukturreferenz	10
4.3.1	Ausführliche Beschreibung	11
4.3.2	Dokumentation der Datenelemente	11
4.3.2.1	color	11
4.3.2.2	normal	11

4.3.2.3	pos	11
4.3.2.4	texCoord	11
5	Datei-Dokumentation	13
5.1	basic_object.h-Dateireferenz	13
5.2	Box.h-Dateireferenz	14
5.3	Fragment.fsh-Dateireferenz	14
5.4	LoadShader.cpp-Dateireferenz	14
5.4.1	Dokumentation der Funktionen	15
5.4.1.1	loadShaders	15
5.5	main.cpp-Dateireferenz	15
5.5.1	Ausführliche Beschreibung	16
5.5.2	Dokumentation der Funktionen	16
5.5.2.1	blueLightDirection	16
5.5.2.2	display	17
5.5.2.3	init	17
5.5.2.4	loadShaders	18
5.5.2.5	main	18
5.5.2.6	redLightDirection	19
5.5.2.7	timer	19
5.5.3	Variablen-Dokumentation	20
5.5.3.1	box	20
5.5.3.2	programID	20
5.5.3.3	rotate	20
5.6	main.h-Dateireferenz	20
5.6.1	Makro-Dokumentation	21
5.6.1.1	OK	21
5.7	Vertex.vsh-Dateireferenz	21
	Index	22

Kapitel 1

Hierarchie-Verzeichnis

1.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

BasicObject	7
Box	9
BasicObject::vertexAttributes	10

Kapitel 2

Klassen-Verzeichnis

2.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

BasicObject	7
Box	9
BasicObject::vertexAttributes	10

Kapitel 3

Datei-Verzeichnis

3.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

basic_object.h	13
Box.h	14
Fragment.fsh	14
LoadShader.cpp	14
main.cpp	15
main.h	20
Vertex.vsh	21

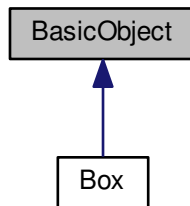
Kapitel 4

Klassen-Dokumentation

4.1 BasicObject Klassenreferenz

```
#include <basic_object.h>
```

Klassendiagramm für BasicObject:



Klassen

- struct [vertexAttributes](#)

Öffentliche Methoden

- [BasicObject](#) (GLuint [typeOfPrimitive](#))
- void [allocateMemory](#) ()
- int [getNumberOfVertices](#) ()

Öffentliche Attribute

- GLuint [vboid](#)
- GLuint [vaoid](#)
- GLuint [typeOfPrimitive](#)
- GLuint [numVertsPerPrimitive](#)
- std::vector< [vertexAttributes](#) > [vertices](#)

4.1.1 Ausführliche Beschreibung

Definiert in Zeile 7 der Datei basic_object.h.

4.1.2 Beschreibung der Konstruktoren und Destruktoren

4.1.2.1 BasicObject::BasicObject (GLuint *typeOfPrimitive*) [inline]

Definiert in Zeile 9 der Datei basic_object.h.

Benutzt numVertsPerPrimitive und typeOfPrimitive.

4.1.3 Dokumentation der Elementfunktionen

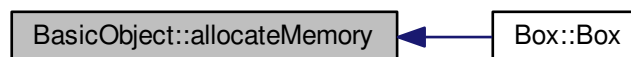
4.1.3.1 void BasicObject::allocateMemory () [inline]

Definiert in Zeile 27 der Datei basic_object.h.

Benutzt vaolD, vboID und vertices.

Wird benutzt von Box::Box().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



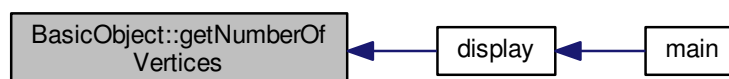
4.1.3.2 int BasicObject::getNumberOfVertices () [inline]

Definiert in Zeile 48 der Datei basic_object.h.

Benutzt vertices.

Wird benutzt von display().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



4.1.4 Dokumentation der Datenelemente

4.1.4.1 GLuint BasicObject::numVertsPerPrimitive

Definiert in Zeile 55 der Datei basic_object.h.

Wird benutzt von BasicObject() und display().

4.1.4.2 GLuint BasicObject::typeOfPrimitive

Definiert in Zeile 54 der Datei basic_object.h.

Wird benutzt von BasicObject() und display().

4.1.4.3 GLuint BasicObject::vaold

Definiert in Zeile 53 der Datei basic_object.h.

Wird benutzt von allocateMemory() und display().

4.1.4.4 GLuint BasicObject::vbold

Definiert in Zeile 50 der Datei basic_object.h.

Wird benutzt von allocateMemory().

4.1.4.5 std::vector<vertexAttributes> BasicObject::vertices

Definiert in Zeile 56 der Datei basic_object.h.

Wird benutzt von allocateMemory(), Box::Box() und getNumberOfVertices().

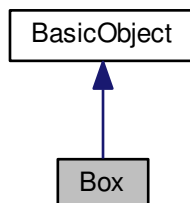
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [basic_object.h](#)

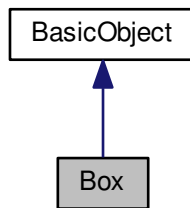
4.2 Box Klassenreferenz

```
#include <Box.h>
```

Klassendiagramm für Box:



Zusammengehörigkeiten von Box:



Öffentliche Methoden

- [Box](#) (float width=1, float height=1, float length=1)

Weitere Geerbte Elemente

4.2.1 Ausführliche Beschreibung

Definiert in Zeile 5 der Datei Box.h.

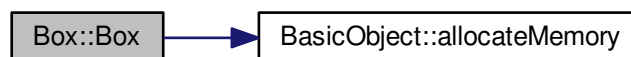
4.2.2 Beschreibung der Konstruktoren und Destruktoren

4.2.2.1 `Box::Box (float width = 1, float height = 1, float length = 1)` `[inline]`

Definiert in Zeile 8 der Datei Box.h.

Benutzt `BasicObject::allocateMemory()`, `BasicObject::vertexAttributes::normal`, `BasicObject::vertexAttributes::pos` und `BasicObject::vertices`.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Box.h](#)

4.3 BasicObject::vertexAttributes Strukturreferenz

```
#include <basic_object.h>
```

Öffentliche Attribute

- glm::vec3 [pos](#)
- glm::vec3 [normal](#)
- glm::vec4 [color](#)
- glm::vec2 [texCoord](#)

4.3.1 Ausführliche Beschreibung

Definiert in Zeile 20 der Datei basic_object.h.

4.3.2 Dokumentation der Datenelemente

4.3.2.1 glm::vec4 BasicObject::vertexAttributes::color

Definiert in Zeile 23 der Datei basic_object.h.

4.3.2.2 glm::vec3 BasicObject::vertexAttributes::normal

Definiert in Zeile 22 der Datei basic_object.h.

Wird benutzt von Box::Box().

4.3.2.3 glm::vec3 BasicObject::vertexAttributes::pos

Definiert in Zeile 21 der Datei basic_object.h.

Wird benutzt von Box::Box().

4.3.2.4 glm::vec2 BasicObject::vertexAttributes::texCoord

Definiert in Zeile 24 der Datei basic_object.h.

Die Dokumentation für diese Struktur wurde erzeugt aufgrund der Datei:

- [basic_object.h](#)

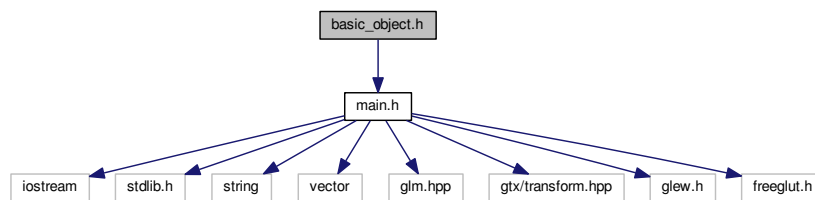
Kapitel 5

Datei-Dokumentation

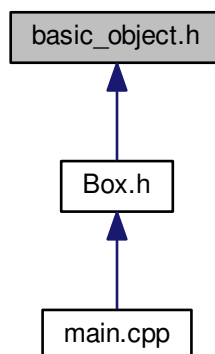
5.1 basic_object.h-Dateireferenz

```
#include "main.h"
```

Include-Abhängigkeitsdiagramm für basic_object.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

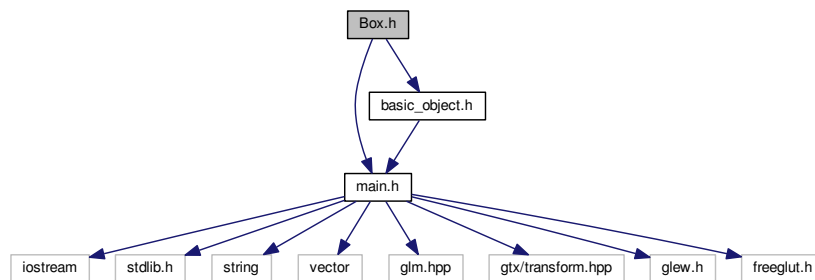
- class [BasicObject](#)
- struct [BasicObject::vertexAttributes](#)

5.2 Box.h-Dateireferenz

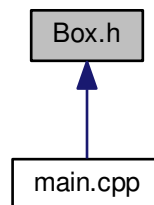
```
#include "main.h"
```

```
#include "basic_object.h"
```

Include-Abhängigkeitsdiagramm für Box.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Klassen

- class [Box](#)

5.3 Fragment.fsh-Dateireferenz

5.4 LoadShader.cpp-Dateireferenz

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string>
#include <fstream>
#include <vector>
#include <glew.h>
#include <glm.hpp>
#include <gtx/transform.hpp>
#include <gtc/matrix_transform.hpp>
Include-Abhängigkeitsdiagramm für LoadShader.cpp:
```



Funktionen

- GLuint `loadShaders` (const char *vertexFilePath, const char *fragmentFilePath)

5.4.1 Dokumentation der Funktionen

5.4.1.1 GLuint loadShaders (const char * vertexFilePath, const char * fragmentFilePath)

Definiert in Zeile 15 der Datei LoadShader.cpp.

Benutzt programID.

Wird benutzt von init().

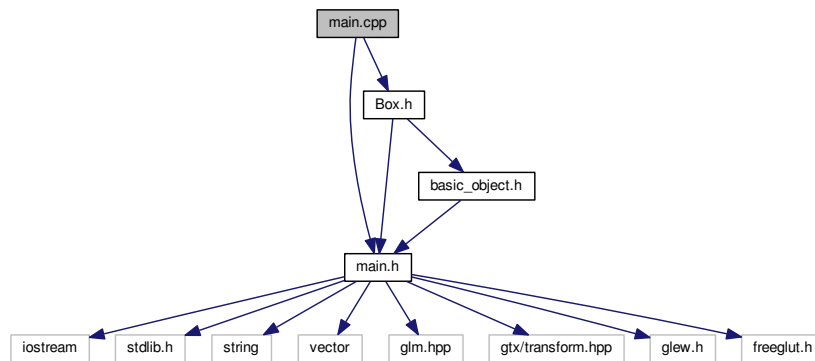
Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



5.5 main.cpp-Dateireferenz

```
#include "main.h"
#include "Box.h"
```

Include-Abhängigkeitsdiagramm für main.cpp:



Funktionen

- GLuint `loadShaders` (const char *vertexFilePath, const char *fragmentFilePath)
- glm::vec3 `redLightDirection` (-1, 0,-1)
- glm::vec3 `blueLightDirection` (1, 0, 0)
- void `init` ()
- void `display` ()
- void `timer` (int value)
- int `main` (int argc, char **argv)

Variablen

- GLuint `programID`
- float `rotate` = 0
- `Box * box`

5.5.1 Ausführliche Beschreibung

OpenGL-Programm eines sich drehenden, von 2 Lampen angestrahlten, Würfels.

Die 1. Lampe leuchtet rot und ist stationär. Die 2. Lampe leuchtet blau und kreist um den Würfel.

g++ *.cpp -I/usr/include/glm -I/usr/include/GL -IGL -IGLEW -IGLU -Iglut

Definiert in Datei `main.cpp`.

5.5.2 Dokumentation der Funktionen

5.5.2.1 glm::vec3 blueLightDirection (1 , 0 , 0)

Wird benutzt von `display()`.

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



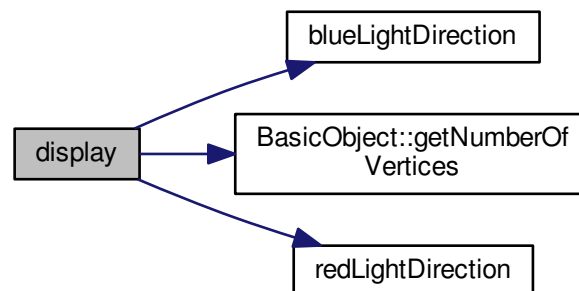
5.5.2.2 void display ()

Definiert in Zeile 31 der Datei main.cpp.

Benutzt blueLightDirection(), BasicObject::getNumberOfVertices(), BasicObject::numVertsPerPrimitive, programID, redLightDirection(), BasicObject::typeOfPrimitive und BasicObject::vaolD.

Wird benutzt von main().

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



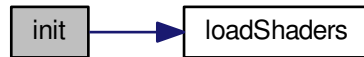
5.5.2.3 void init ()

Definiert in Zeile 22 der Datei main.cpp.

Benutzt loadShaders() und programID.

Wird benutzt von main().

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



5.5.2.4 GLuint loadShaders (const char * vertexFilePath, const char * fragmentFilePath)

Definiert in Zeile 15 der Datei LoadShader.cpp.

Benutzt programID.

Wird benutzt von init().

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

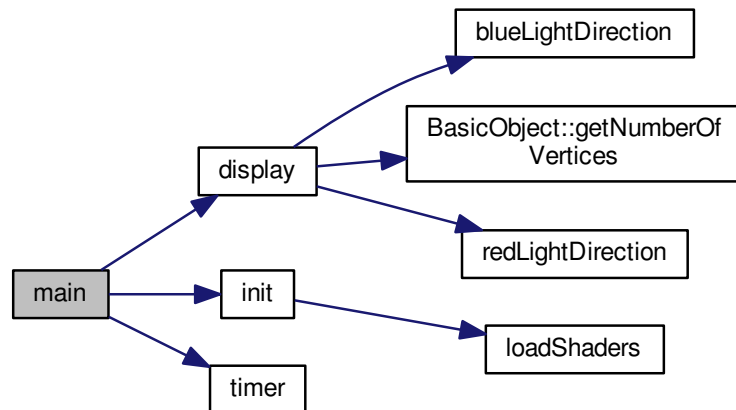


5.5.2.5 int main (int argc, char ** argv)

Definiert in Zeile 70 der Datei main.cpp.

Benutzt display(), init(), OK und timer().

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



5.5.2.6 glm::vec3 redLightDirection (- 1, 0, - 1)

Wird benutzt von `display()`.

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



5.5.2.7 void timer (int value)

Definiert in Zeile 62 der Datei `main.cpp`.

Benutzt `rotate`.

Wird benutzt von `main()`.

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



5.5.3 Variablen-Dokumentation

5.5.3.1 `Box* box`

Definiert in Zeile 17 der Datei `main.cpp`.

5.5.3.2 `GLuint programID`

Definiert in Zeile 13 der Datei `main.cpp`.

Wird benutzt von `display()`, `init()` und `loadShaders()`.

5.5.3.3 `float rotate = 0`

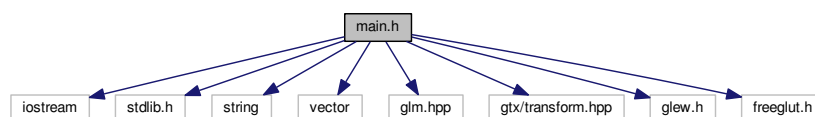
Definiert in Zeile 15 der Datei `main.cpp`.

Wird benutzt von `timer()`.

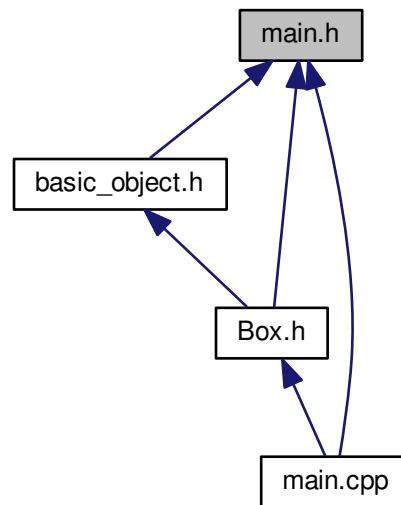
5.6 `main.h`-Dateireferenz

```
#include <iostream>
#include <stdlib.h>
#include <string>
#include <vector>
#include <glm.hpp>
#include <gtx/transform.hpp>
#include <glew.h>
#include <freeglut.h>
```

Include-Abhängigkeitsdiagramm für `main.h`:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Makrodefinitionen

- #define [OK](#) 0

5.6.1 Makro-Dokumentation

5.6.1.1 #define OK 0

Definiert in Zeile 14 der Datei main.h.

Wird benutzt von main().

5.7 Vertex.vsh-Dateireferenz

Index

Box, [9](#)
Box, [10](#)