Light

0.0.1

Generated by Doxygen 1.8.17

Chapter 1

Bug List

File collider.cpp

No known bugs.

File physicsworld.cpp

No known bugs.

File rigidbody.cpp

No known bugs.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

Class Documentation

5.1 Physicc::BVImpl::AABB Struct Reference

Axis Aligned Bounding Box.

```
#include <boundingvolume.hpp>
```

Public Member Functions

- **AABB** (const glm::vec3 &lb, const glm::vec3 &ub)

Public Attributes

- glm::vec3 **lowerBound**
- glm::vec3 **upperBound**

5.1.1 Detailed Description

Axis Aligned Bounding Box.

Helper struct to store diagonally opposite points of the AABB

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

- Physicc/include/boundingvolume.hpp

5.2 Light::Application Class Reference

Public Member Functions

- void **onEvent** (Event &e)
- void **run** ()
- void **pushLayer** (Layer ∗layer)
- void **pushOverlay** (Layer ∗overlay)
- Window & **getWindow** ()
- ImguiLayer ∗ **getImguiLayer** ()
- void **close** ()

Static Public Member Functions

- static Application & **get** ()

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

- LightFramework/include/core/application.hpp
- LightFramework/src/core/application.cpp

5.3 Physicc::BVImpl::BaseBV< Derived, BoundingObject > Class Template Reference

```
#include <boundingvolume.hpp>
```

Public Member Functions

- BaseBV (const BaseBV &bv)=default

  Copy constructor for BoundingVolume.
- BaseBV (const BoundingObject &volume)

  A constructor for BV which takes a BoundingObject (like the AABB struct) as a parameter.
- **BaseBV** (const glm::vec3 &lowerBound, const glm::vec3 &upperBound)
- bool overlapsWith (const BaseBV &bv) const

  Returns whether two BVs are overlapping or not.
- float **getVolume** () const
- Derived **enclosingBV** (const BaseBV &bv) const

5.3.1 Detailed Description

template<typename Derived, typename BoundingObject>
class Physicc::BVImpl::BaseBV< Derived, BoundingObject >

A templated class that defines the Bounding Volume (BV) of an object, but in a way that allows others to hot swap actual bounding volumes (like AABBs, OBBs, 8-DOPs, etc.).

TODO: Figure out if this is good enough as a description of the BV class.

Template Parameters

Derived	TODO: Update this Doxygen comment

5.3.2 Constructor & Destructor Documentation

5.3.2.1 BaseBV() [1/2]

```
template<typename Derived , typename BoundingObject >
Physicc::BVImpl::BaseBV< Derived, BoundingObject >::BaseBV (
            const BaseBV< Derived, BoundingObject > & bv )  [default]
```

Copy constructor for BoundingVolume.

Template Parameters

BV	The object to be copied

Parameters

bv	A BV object

5.3.2.2 BaseBV() [2/2]

```
template<typename Derived , typename BoundingObject >
Physicc::BVImpl::BaseBV< Derived, BoundingObject >::BaseBV (
            const BoundingObject & volume )  [inline]
```

A constructor for BV which takes a BoundingObject (like the AABB struct) as a parameter.

Template Parameters

BoundingObject	The bounding volume struct to be used (AABB, OBB, etc.)

Parameters

volume	

5.3.3 Member Function Documentation

5.3.3.1 overlapsWith()

```
template<typename Derived , typename BoundingObject >
bool Physicc::BVImpl::BaseBV< Derived, BoundingObject >::overlapsWith (
            const BaseBV< Derived, BoundingObject > & bv ) const  [inline]
```

Returns whether two BVs are overlapping or not.

Parameters

bv	A BaseBV object

Returns

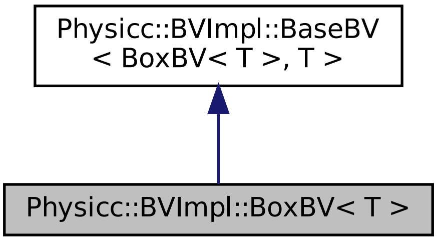
true if the BoundingVolumes are intersecting, and false otherwise TODO: Is this, as a return type description, fine?

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

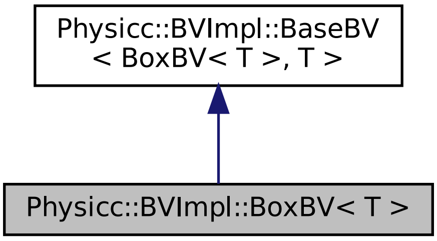
- Physicc/include/boundingvolume.hpp

5.4 Physicc::BVImpl::BoxBV< T > Class Template Reference

Inheritance diagram for Physicc::BVImpl::BoxBV< T >:



Collaboration diagram for Physicc::BVImpl::BoxBV< T >:



Public Member Functions

- **BoxBV** (const BoxBV &bv)=default
- **BoxBV** (const glm::vec3 &lowerBound, const glm::vec3 &upperBound)
- void **setVolume** (const T &volume)
- void **setVolume** (const glm::vec3 &lowerBound, const glm::vec3 &upperBound)
- float **getVolume** () const
- bool **overlapsWith** (const BoxBV &bv) const
- BoxBV **enclosingBV** (const BoxBV &bv) const

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

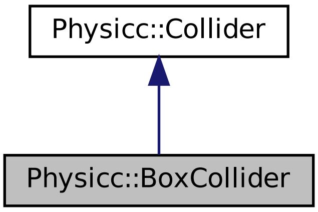
- Physicc/include/boundingvolume.hpp

5.5 Physicc::BoxCollider Class Reference

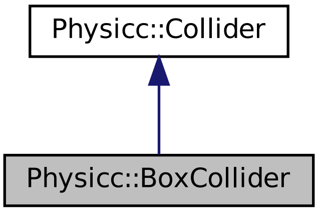
BoxCollider class.

```
#include <collider.hpp>
```

Inheritance diagram for Physicc::BoxCollider:



Collaboration diagram for Physicc::BoxCollider:



Public Member Functions

- BoxCollider (glm::vec3 position=glm::vec3(0), glm::vec3 rotation=glm::vec3(0), glm::vec3 scale=glm::vec3(1))

    Creates a BoxCollider object.
- BoundingVolume::AABB getAABB () const override

    Computes and returns Axis Aligned Bounding Box of Box shaped object.

Additional Inherited Members

5.5.1 Detailed Description

BoxCollider class.

Box shaped collider, holds the shape and transform of the body.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 BoxCollider()

```
Physicc::BoxCollider::BoxCollider (
              glm::vec3 position = glm::vec3(0),
              glm::vec3 rotation = glm::vec3(0),
              glm::vec3 scale = glm::vec3(1) )
```

Creates a BoxCollider object.

Parameters

position	Position of object in global space
rotation	Rotation about each of the axis in local space
scale	Scale of the object along each axis

5.5.3 Member Function Documentation

5.5.3.1 getAABB()

```
BoundingVolume::AABB Physicc::BoxCollider::getAABB ( ) const  [override], [virtual]
```

Computes and returns Axis Aligned Bounding Box of Box shaped object.

Computes location of vertices in global space and finds the extreme points of AABB by comparing each component of every vertex

Returns

BoundingVolume::AABB

Implements Physicc::Collider.

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

- Physicc/include/collider.hpp
- Physicc/src/collider.cpp

5.6 Light::BufferElement Struct Reference

Public Member Functions

- **BufferElement** (ShaderDataType type, std::string name, bool normalized=false)
- void **setOffset** (uint32_t offset)
- uint32_t **getSize** () const
- uint32_t **getComponentCount** () const
- uint32_t **getOffset** () const
- ShaderDataType **getType** () const
- bool **isNormalized** () const

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

- Light/include/light/rendering/buffer.hpp

5.7 Light::BufferLayout Class Reference

Public Member Functions

- **BufferLayout** (std::initializer_list< BufferElement > elements)
- uint32_t **getStride** () const
- const std::vector< BufferElement > & **getElements** () const
- std::vector< BufferElement >::iterator **begin** ()
- std::vector< BufferElement >::iterator **end** ()
- std::vector< BufferElement >::const_iterator **begin** () const
- std::vector< BufferElement >::const_iterator **end** () const

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

- Light/include/light/rendering/buffer.hpp

5.8 Physicc::BVH Class Reference

Public Member Functions

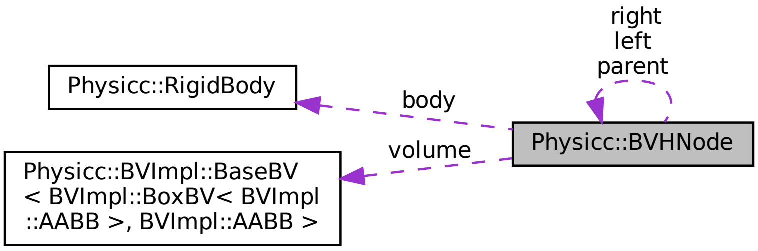
- **BVH** (std::vector< RigidBody > rigidBodyList)
- void **buildTree** ()
- std::vector< RigidBody > & **convert** ()

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

- Physicc/include/bvh.hpp
- Physicc/src/bvh.cpp

5.9 Physicc::BVHNode Struct Reference

Collaboration diagram for Physicc::BVHNode:



Public Attributes

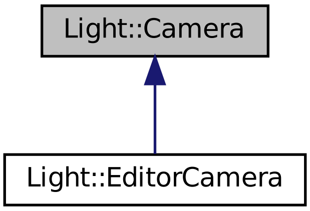
- BoundingVolume::AABB **volume**
- RigidBody ∗ **body** = nullptr
- BVHNode ∗ **parent** = nullptr
- BVHNode ∗ **left** = nullptr
- BVHNode ∗ **right** = nullptr

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

- Physicc/include/bvh.hpp

5.10 Light::Camera Class Reference

Inheritance diagram for Light::Camera:



Public Member Functions

- **Camera** (glm::mat4 projectionMatrix=glm::mat4(1.0f))
- const glm::mat4 & **getProjectionMatrix** ()
- void **setProjectionMatrix** (glm::mat4 projectionMatrix)

Protected Attributes

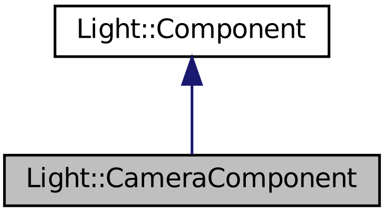
- glm::mat4 **m_projectionMatrix**

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

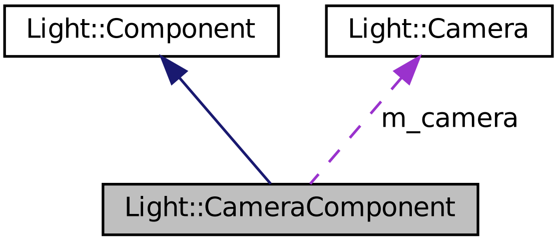
- Light/include/light/rendering/camera.hpp

5.11 Light::CameraComponent Struct Reference

Inheritance diagram for Light::CameraComponent:



Collaboration diagram for Light::CameraComponent:



Public Member Functions

- **CameraComponent** (glm::mat4 projectionMatrix)

Public Attributes

- Camera **m_camera**

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

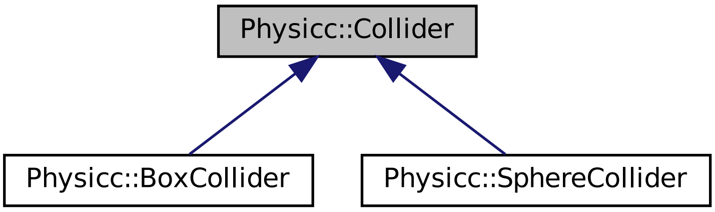
- LightFramework/include/ecs/components.hpp

5.12 Physicc::Collider Class Reference

Collider class.

`#include <collider.hpp>`

Inheritance diagram for Physicc::Collider:



Public Member Functions

- Collider (glm::vec3 position=glm::vec3(0), glm::vec3 rotation=glm::vec3(0), glm::vec3 scale=glm::vec3(1))

    Construct a new Collider:: Collider object.
- glm::vec3 getPosition ()

    Get Position of object's center.
- glm::vec3 getRotate ()

    get Angle of rotation of the object about its center
- glm::vec3 getScale ()

    get Scale of object
- glm::mat4 getTransform ()

    get Transform matrix of object
- void setPosition (glm::vec3 position)

    set Position of object's center
- void setRotate (glm::vec3 rotate)

    Set rotation of object about it's center.
- void setScale (glm::vec3 scale)

    get Position of object's center
- void updateTransform ()

    Update Transform for rendering.
- virtual BoundingVolume::AABB **getAABB** () const =0

Protected Types

- enum **Type** { **e_box** = 0, **e_sphere** = 1, **e_typecount** = 2 }

Protected Attributes

- glm::vec3 **m_position**
- glm::vec3 **m_rotate**
- glm::vec3 **m_scale**
- glm::mat4 **m_transform**
- Type **m_objectType**

5.12.1 Detailed Description

Collider class.

This is a virtual class which acts as the base for all the shape specific classes

5.12.2 Constructor & Destructor Documentation

5.12.2.1 Collider()

```
Physicc::Collider::Collider (
              glm::vec3 position = glm::vec3(0),
              glm::vec3 rotation = glm::vec3(0),
              glm::vec3 scale = glm::vec3(1) )
```

Construct a new Collider:: Collider object.

Parameters

position	Position of the object. Default = (0,0,0)
rotation	Rotations about the axes. Default = (0,0,0)
scale	Length along each of the axes. Default = (1,1,1)

5.12.3 Member Function Documentation

5.12.3.1 getPosition()

```
glm::vec3 Physicc::Collider::getPosition ( )  [inline]
```

Get Position of object's center.

Returns

glm::vec3

5.12.3.2 getRotate()

```
glm::vec3 Physicc::Collider::getRotate ( )  [inline]
```

get Angle of rotation of the object about its center

Returns

glm::vec3

5.12.3.3 getScale()

```
glm::vec3 Physicc::Collider::getScale ( )  [inline]
```

get Scale of object

Returns

glm::vec3

5.12.3.4 getTransform()

```
glm::mat4 Physicc::Collider::getTransform ( )  [inline]
```

get Transform matrix of object

Returns

glm::mat4

5.12.3.5 setPosition()

```
void Physicc::Collider::setPosition (
             glm::vec3 position )  [inline]
```

set Position of object's center

Parameters

position	Takes the (x,y,z) coordinates to place the object's center at

5.12.3.6 setRotate()

```
void Physicc::Collider::setRotate (
             glm::vec3 rotate )  [inline]
```

Set rotation of object about it's center.

Parameters

rotate	vec3 containing rotation values about x, y, z axes

5.12.3.7 setScale()

```
void Physicc::Collider::setScale (
             glm::vec3 scale )  [inline]
```

get Position of object's center

Parameters

scale	New scale of the object

5.12.3.8 updateTransform()

```
void Physicc::Collider::updateTransform ( )
```

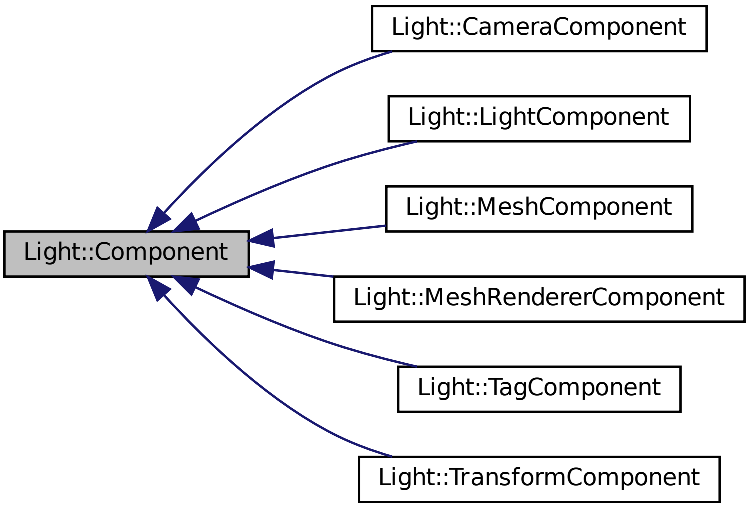
Update Transform for rendering.

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

- Physicc/include/collider.hpp
- Physicc/src/collider.cpp

5.13 Light::Component Struct Reference

Inheritance diagram for Light::Component:



Public Attributes

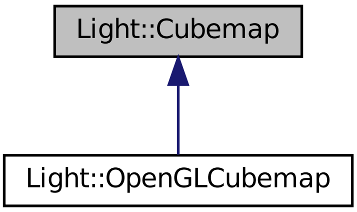
- std::string **uuid**

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

- LightFramework/include/ecs/components.hpp

5.14 Light::Cubemap Class Reference

Inheritance diagram for Light::Cubemap:



Public Member Functions

- virtual void **bind** (uint32_t slot=0) const =0

Static Public Member Functions

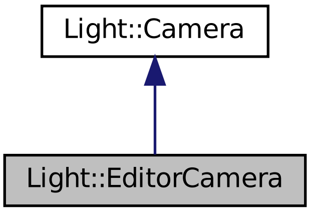
- static Cubemap ∗ **create** (const std::string &path)

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

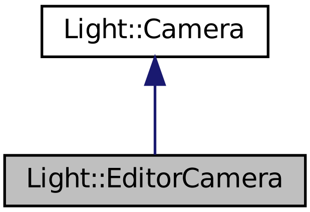
- Light/include/light/rendering/texture.hpp
- Light/src/platform/opengl/opengltexture.cpp

5.15 Light::EditorCamera Class Reference

Inheritance diagram for Light::EditorCamera:



Collaboration diagram for Light::EditorCamera:



Public Member Functions

- **EditorCamera** (float fovy, float aspectRatio, float near, float far)
- void **onUpdate** (Timestep ts)
- void **onEvent** (Event &e)
- void **setViewportSize** (int width, int height)
- const glm::mat4 & **getViewMatrix** () const
- glm::mat4 **getViewProjectionMatrix** ()
- glm::vec3 **getUpDirection** () const
- glm::vec3 **getRightDirection** () const
- glm::vec3 **getForwardDirection** () const
- glm::quat **getOrientation** () const

Additional Inherited Members

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

- LightFramework/include/rendering/editorcamera.hpp
- LightFramework/src/rendering/editorcamera.cpp

5.16 Light::Entity Class Reference

Public Member Functions

- **Entity** (entt::entity entity, Scene ∗scene)
- **Entity** (const Entity &other)=default
- template<typename T , typename... Args>
  T & **addComponent** (Args... args)
- template<typename T >
  bool **hasComponent** ()
- template<typename T >
  T & **getComponent** ()
- template<typename T >
  void **removeComponent** ()

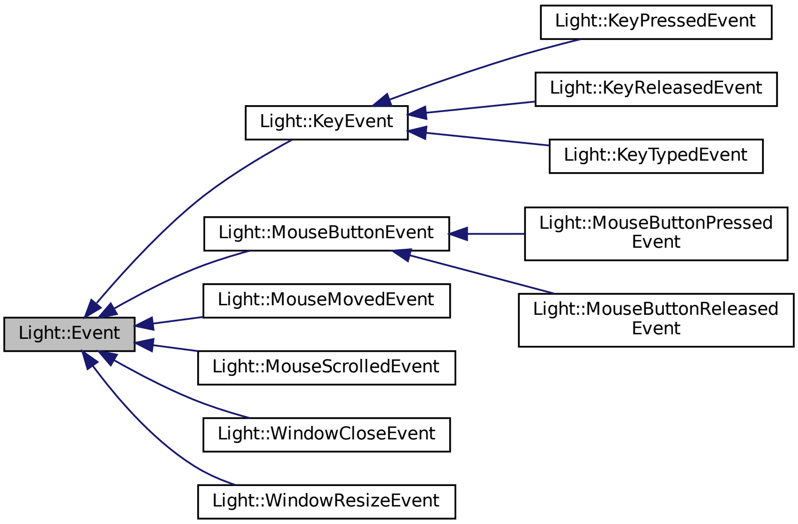
- **operator bool** ()
- **operator uint32_t** ()
- bool **operator==** (const Entity &other)
- std::string **getUUID** ()

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

- LightFramework/include/ecs/entity.hpp
- LightFramework/src/ecs/entity.cpp

5.17 Light::Event Class Reference

Inheritance diagram for Light::Event:



Public Member Functions

- virtual EventType **GetEventType** () const =0
- virtual const char ∗ **GetName** () const =0
- virtual int **GetCategoryFlags** () const =0
- virtual std::string **ToString** () const
- bool **IsInCategory** (EventCategory category)

Public Attributes

- bool **handled** = false

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

- LightFramework/include/events/event.hpp

5.18 Light::EventDispatcher Class Reference

Public Member Functions

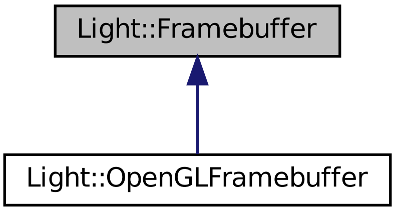
- **EventDispatcher** (Event &event)
- template<typename T , typename F >
  bool **Dispatch** (const F &func)

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

- LightFramework/include/events/event.hpp

5.19 Light::Framebuffer Class Reference

Inheritance diagram for Light::Framebuffer:



Public Member Functions

- virtual const FramebufferSpec & **getSpec** () const =0
- virtual uint32_t **getColorAttachmentRendererId** (uint32_t attachmentIndex=0) const =0
- virtual void **resize** (uint32_t width, uint32_t height)=0
- virtual int **readPixelInt** (uint32_t attachmentIndex, uint32_t x, uint32_t y)=0
- virtual glm::vec4 **readPixelVec4** (uint32_t attachmentIndex, uint32_t x, uint32_t y)=0
- virtual void **clearAttachment** (uint32_t attachmentIndex, int clearValue)=0
- virtual void **clearAttachment** (uint32_t attachmentIndex, glm::vec4 clearValue)=0
- virtual void **bind** ()=0
- virtual void **unbind** ()=0
- virtual void **bindAttachmentTexture** (uint32_t attachmentIndex, uint32_t slot)=0

Static Public Member Functions

- static std::shared_ptr< Framebuffer > **create** (const FramebufferSpec &spec)

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

- Light/include/light/rendering/framebuffer.hpp
- Light/src/platform/opengl/openglframebuffer.cpp

5.20 Light::FramebufferAttachmentsSpec Struct Reference

Public Member Functions

- **FramebufferAttachmentsSpec** (std::initializer_list< FramebufferTextureSpec > attachmentList)
- std::vector< FramebufferTextureSpec >::iterator **begin** ()
- std::vector< FramebufferTextureSpec >::iterator **end** ()
- std::vector< FramebufferTextureSpec >::const_iterator **begin** () const
- std::vector< FramebufferTextureSpec >::const_iterator **end** () const

Public Attributes

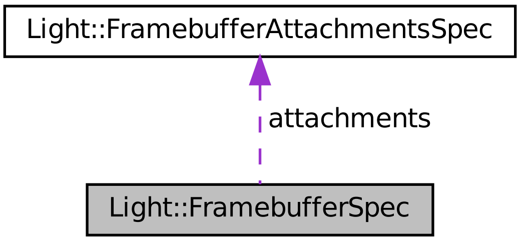
- std::vector< FramebufferTextureSpec > **attachments**

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

- Light/include/light/rendering/framebuffer.hpp

5.21 Light::FramebufferSpec Struct Reference

Collaboration diagram for Light::FramebufferSpec:



Public Attributes

- uint32_t **width**
- uint32_t **height**
- uint32_t **samples** = 1
- FramebufferAttachmentsSpec **attachments**
- bool **swapChainTarget** = false

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

- Light/include/light/rendering/framebuffer.hpp

5.22 Light::FramebufferTextureSpec Struct Reference

Public Member Functions

- **FramebufferTextureSpec** (FramebufferTextureFormat format, TextureWrap wrap)

Public Attributes

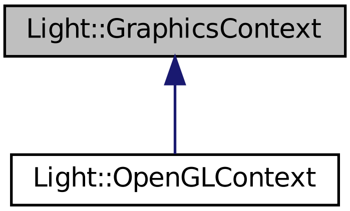
- FramebufferTextureFormat **textureFormat** = FramebufferTextureFormat::None
- TextureWrap **wrapFormat** = TextureWrap::None

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

- Light/include/light/rendering/framebuffer.hpp

5.23 Light::GraphicsContext Class Reference

Inheritance diagram for Light::GraphicsContext:



Public Member Functions

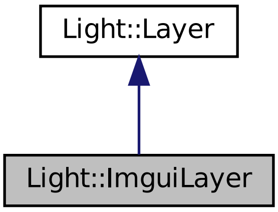
- virtual void **init** ()=0
- virtual void **swapBuffers** ()=0

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

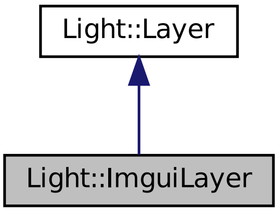
- Light/include/light/rendering/graphicscontext.hpp

5.24 Light::ImguiLayer Class Reference

Inheritance diagram for Light::ImguiLayer:



Collaboration diagram for Light::ImguiLayer:



Public Member Functions

- **ImguiLayer** (std::string name)
- void **onAttach** () override
- void **onDetach** () override
- void **onEvent** (Event &e) override
- void **onImguiRender** () override
- void **begin** ()
- void **end** ()

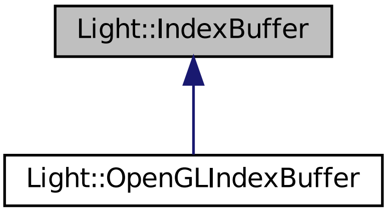
Additional Inherited Members

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

- LightFramework/include/imgui/imguilayer.hpp
- LightFramework/src/imgui/imguilayer.cpp

5.25 Light::IndexBuffer Class Reference

Inheritance diagram for Light::IndexBuffer:



Public Member Functions

- virtual void **bind** () const =0
- virtual void **unbind** () const =0
- virtual uint32_t **getCount** () const =0

Static Public Member Functions

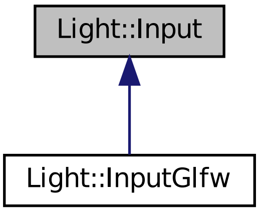
- static IndexBuffer ∗ **create** (uint32_t ∗indices, uint32_t count)

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

- Light/include/light/rendering/buffer.hpp
- Light/src/platform/opengl/openglbuffer.cpp

5.26 Light::Input Class Reference

Inheritance diagram for Light::Input:



Static Public Member Functions

- static bool **isKeyPressed** (int keycode)
- static bool **isMouseButtonPressed** (int button)
- static std::tuple< float, float > **getMousePos** ()

Protected Member Functions

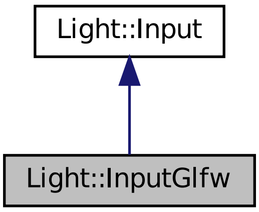
- virtual bool **isKeyPressedImpl** (int keycode)=0
- virtual bool **isMouseButtonPressedImpl** (int button)=0
- virtual std::tuple< float, float > **getMousePosImpl** ()=0

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

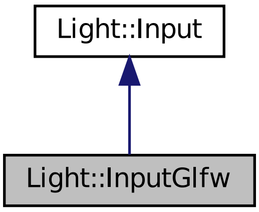
- LightFramework/include/core/input.hpp
- LightFramework/src/platform/glfw/inputglfw.cpp

5.27 Light::InputGlfw Class Reference

Inheritance diagram for Light::InputGlfw:



Collaboration diagram for Light::InputGlfw:



Protected Member Functions

- bool **isKeyPressedImpl** (int keycode) override
- bool **isMouseButtonPressedImpl** (int button) override
- std::tuple< float, float > **getMousePosImpl** () override

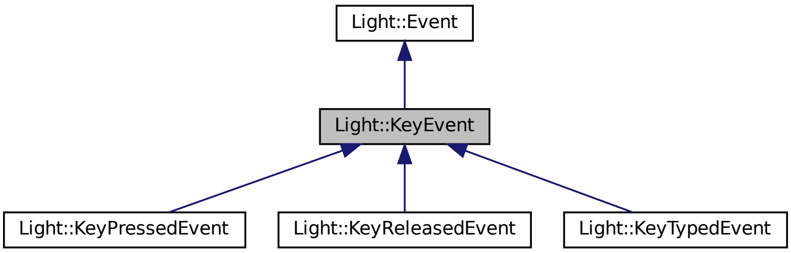
Additional Inherited Members

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

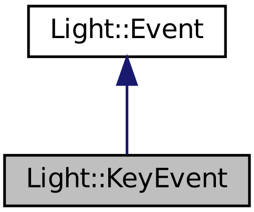
- LightFramework/include/platform/glfw/inputglfw.hpp
- LightFramework/src/platform/glfw/inputglfw.cpp

5.28 Light::KeyEvent Class Reference

Inheritance diagram for Light::KeyEvent:



Collaboration diagram for Light::KeyEvent:



Public Member Functions

- int **getKeycode** () const

Protected Member Functions

- **KeyEvent** (int keycode)

Protected Attributes

- int **m_keycode**

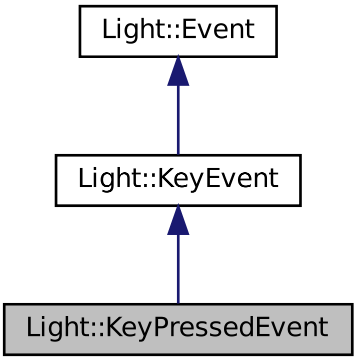
Additional Inherited Members

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

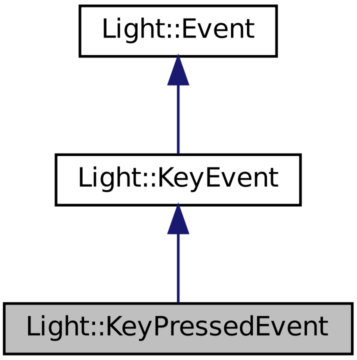
- LightFramework/include/events/keyevent.hpp

5.29 Light::KeyPressedEvent Class Reference

Inheritance diagram for Light::KeyPressedEvent:



Collaboration diagram for Light::KeyPressedEvent:



Public Member Functions

- **KeyPressedEvent** (int keycode, int repeatcount)
- int **getRepeatCount** ()
- std::string **ToString** () const override

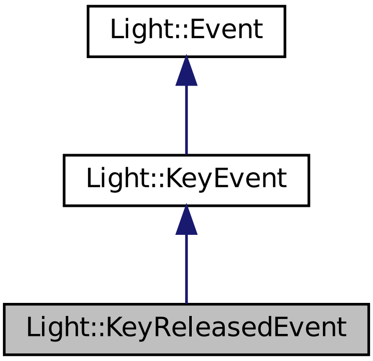
Additional Inherited Members

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

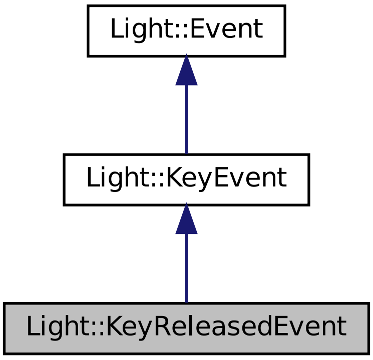
- LightFramework/include/events/keyevent.hpp

5.30 Light::KeyReleasedEvent Class Reference

Inheritance diagram for Light::KeyReleasedEvent:



Collaboration diagram for Light::KeyReleasedEvent:



Public Member Functions

- **KeyReleasedEvent** (int keycode)
- std::string **ToString** () const override

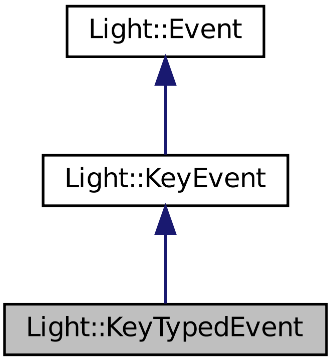
Additional Inherited Members

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

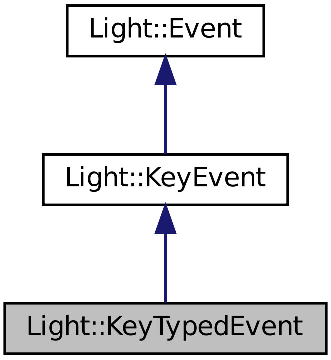
- LightFramework/include/events/keyevent.hpp

5.31 Light::KeyTypedEvent Class Reference

Inheritance diagram for Light::KeyTypedEvent:



Collaboration diagram for Light::KeyTypedEvent:



Public Member Functions

- **KeyTypedEvent** (int keycode)
- std::string **ToString** () const override

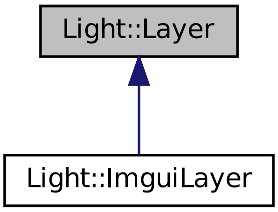
Additional Inherited Members

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

- LightFramework/include/events/keyevent.hpp

5.32 Light::Layer Class Reference

Inheritance diagram for Light::Layer:



Public Member Functions

- **Layer** (std::string name)
- virtual void **onAttach** ()
- virtual void **onDetach** ()
- virtual void **onEvent** (Event &e)
- virtual void **onUpdate** (Timestep ts)
- virtual void **onImguiRender** ()
- const std::string & **getName** ()
- void **blockHoverEvents** (bool block)
- void **blockFocusEvents** (bool block)
- bool **getHoverEventsBlocking** ()
- bool **getFocusEventsBlocking** ()

Protected Attributes

- std::string **m_name**
- bool **m_hoverEventsBlocking** = true
- bool **m_focusEventsBlocking** = true

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

- LightFramework/include/core/layer.hpp

5.33 Light::LayerStack Class Reference

Public Member Functions

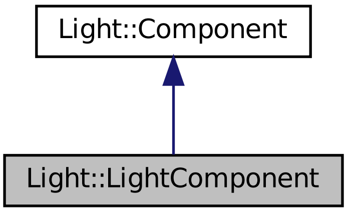
- void **pushLayer** (Layer ∗layer)
- void **popLayer** (Layer ∗layer)
- void **pushOverlay** (Layer ∗overlay)
- void **popOverlay** (Layer ∗overlay)
- std::vector< Layer ∗ >::iterator **begin** ()
- std::vector< Layer ∗ >::iterator **end** ()

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

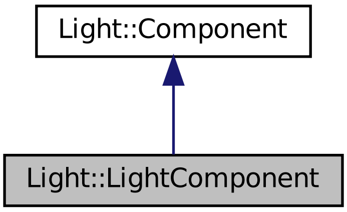
- LightFramework/include/core/layerstack.hpp
- LightFramework/src/core/layerstack.cpp

5.34 Light::LightComponent Struct Reference

Inheritance diagram for Light::LightComponent:



Collaboration diagram for Light::LightComponent:



Public Member Functions

- **LightComponent** (glm::vec3 lightColor)

Public Attributes

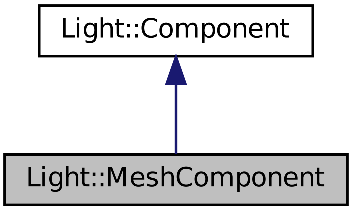
- glm::vec3 **m_lightColor**

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

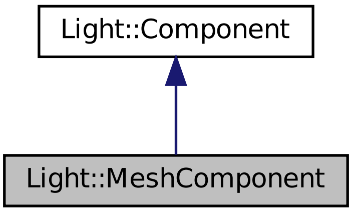
- LightFramework/include/ecs/components.hpp

5.35 Light::MeshComponent Struct Reference

Inheritance diagram for Light::MeshComponent:



Collaboration diagram for Light::MeshComponent:



Public Member Functions

- **MeshComponent** (std::shared_ptr< Light::VertexBuffer > vertexBuffer, std::shared_ptr< Light::IndexBuffer > indexBuffer)

Public Attributes

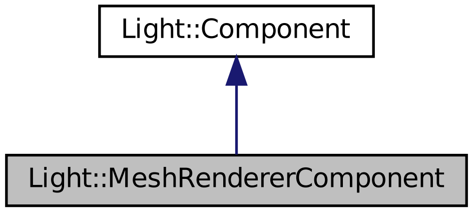
- std::shared_ptr< Light::VertexArray > **mesh**

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

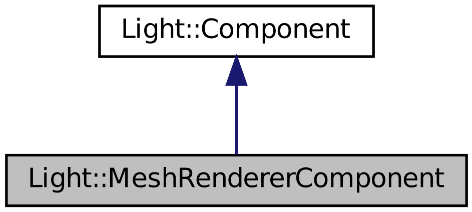
- LightFramework/include/ecs/components.hpp
- LightFramework/src/ecs/components.cpp

5.36 Light::MeshRendererComponent Struct Reference

Inheritance diagram for Light::MeshRendererComponent:



Collaboration diagram for Light::MeshRendererComponent:



Public Member Functions

- **MeshRendererComponent** (const char ∗path)
- void **bind** ()
- void **setUniformInt** (const std::string &name, int value)

Public Attributes

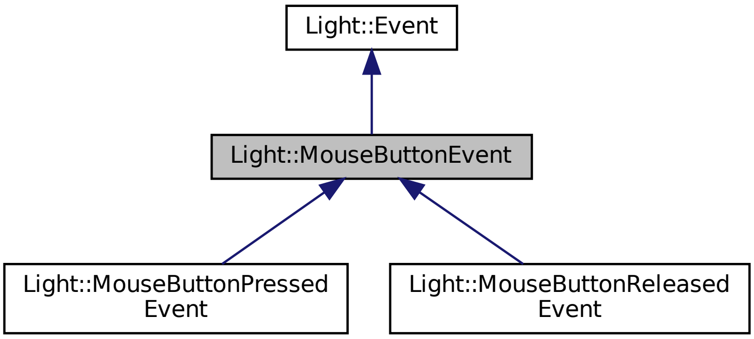
- std::shared_ptr< Light::Shader > **shader**

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

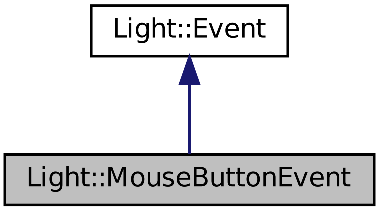
- LightFramework/include/ecs/components.hpp
- LightFramework/src/ecs/components.cpp

5.37 Light::MouseButtonEvent Class Reference

Inheritance diagram for Light::MouseButtonEvent:


Light::Event
Light::MouseButtonEvent
Light::MouseButtonPressed Event
Light::MouseButtonReleased Event

Collaboration diagram for Light::MouseButtonEvent:



Public Member Functions

- int **getButton** ()

Protected Member Functions

- **MouseButtonEvent** (int button)

Protected Attributes

- int **button**

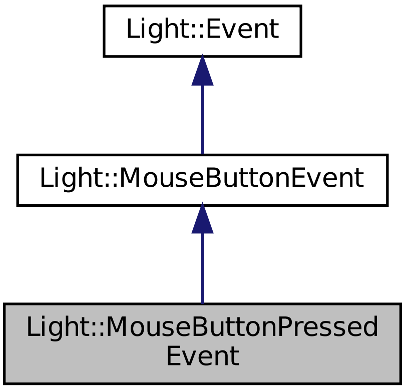
Additional Inherited Members

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

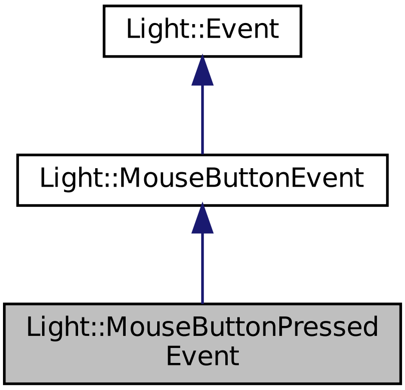
- LightFramework/include/events/mouseevent.hpp

5.38 Light::MouseButtonPressedEvent Class Reference

Inheritance diagram for Light::MouseButtonPressedEvent:



Collaboration diagram for Light::MouseButtonPressedEvent:



Public Member Functions

- **MouseButtonPressedEvent** (int button)
- std::string **ToString** () const override

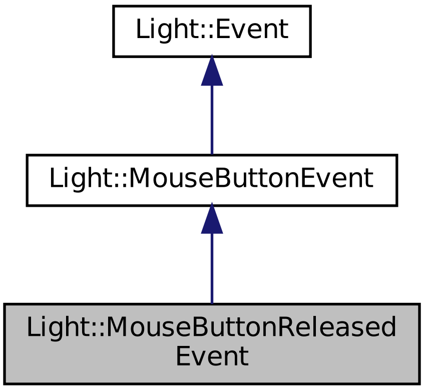
Additional Inherited Members

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

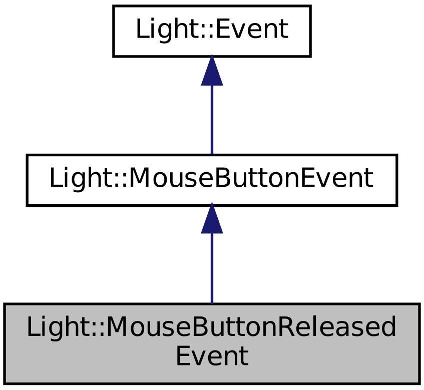
- LightFramework/include/events/mouseevent.hpp

5.39 Light::MouseButtonReleasedEvent Class Reference

Inheritance diagram for Light::MouseButtonReleasedEvent:



Collaboration diagram for Light::MouseButtonReleasedEvent:



Public Member Functions

- **MouseButtonReleasedEvent** (int button)
- std::string **ToString** () const override

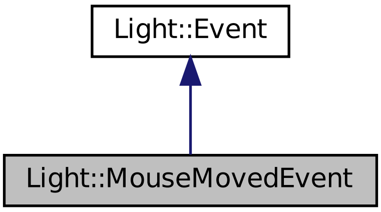
Additional Inherited Members

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

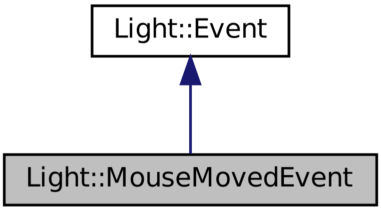
- LightFramework/include/events/mouseevent.hpp

5.40 Light::MouseMovedEvent Class Reference

Inheritance diagram for Light::MouseMovedEvent:



Collaboration diagram for Light::MouseMovedEvent:



Public Member Functions

- **MouseMovedEvent** (double x, double y)
- std::tuple< double, double > **getPos** ()
- std::string **ToString** () const override

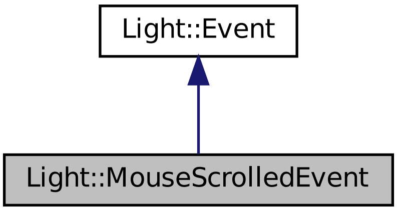
Additional Inherited Members

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

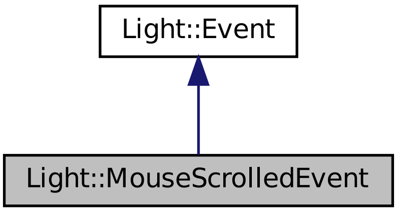
- LightFramework/include/events/mouseevent.hpp

5.41 Light::MouseScrolledEvent Class Reference

Inheritance diagram for Light::MouseScrolledEvent:



Collaboration diagram for Light::MouseScrolledEvent:



Public Member Functions

- **MouseScrolledEvent** (double x, double y)
- std::tuple< double, double > **getOffset** ()
- std::string **ToString** () const override

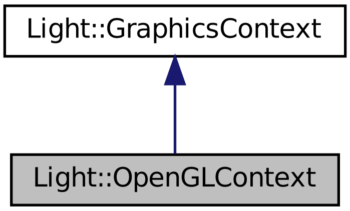
Additional Inherited Members

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

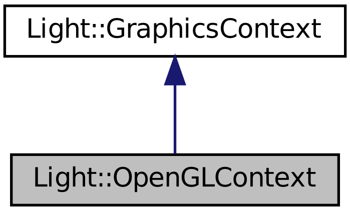
- LightFramework/include/events/mouseevent.hpp

5.42 Light::OpenGLContext Class Reference

Inheritance diagram for Light::OpenGLContext:



Collaboration diagram for Light::OpenGLContext:



Public Member Functions

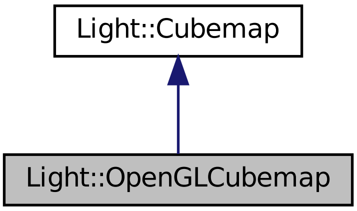
- **OpenGLContext** (GLFWwindow ∗windowHandle)
- void **init** () override
- void **swapBuffers** () override

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

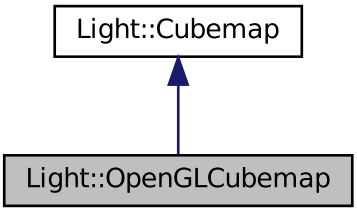
- Light/include/light/platform/opengl/openglcontext.hpp
- Light/src/platform/opengl/openglcontext.cpp

5.43 Light::OpenGLCubemap Class Reference

Inheritance diagram for Light::OpenGLCubemap:



Collaboration diagram for Light::OpenGLCubemap:



Public Member Functions

- **OpenGLCubemap** (const std::string &path)
- void **bind** (uint32_t slot=0) const override

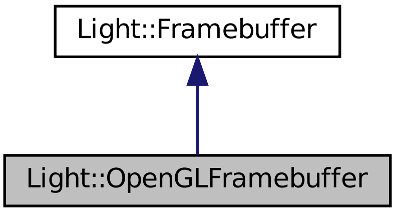
Additional Inherited Members

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

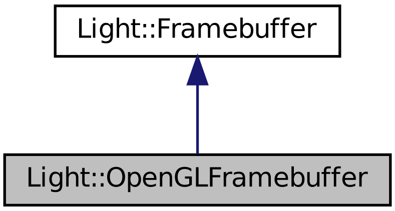
- Light/include/light/platform/opengl/opengltexture.hpp
- Light/src/platform/opengl/opengltexture.cpp

5.44 Light::OpenGLFramebuffer Class Reference

Inheritance diagram for Light::OpenGLFramebuffer:



Collaboration diagram for Light::OpenGLFramebuffer:



Public Member Functions

- **OpenGLFramebuffer** (const FramebufferSpec &spec)
- const FramebufferSpec & **getSpec** () const override
- void **bind** () override
- void **unbind** () override
- void **invalidate** ()
- void **resize** (uint32_t width, uint32_t height) override
- int **readPixelInt** (uint32_t attachmentIndex, uint32_t x, uint32_t y) override
- glm::vec4 **readPixelVec4** (uint32_t attachmentIndex, uint32_t x, uint32_t y) override
- void **clearAttachment** (uint32_t attachmentIndex, int clearValue) override
- void **clearAttachment** (uint32_t attachmentIndex, glm::vec4 clearValue) override
- uint32_t **getColorAttachmentRendererId** (uint32_t attachmentIndex=0) const override
- virtual void **bindAttachmentTexture** (uint32_t attachmentIndex, uint32_t slot) override

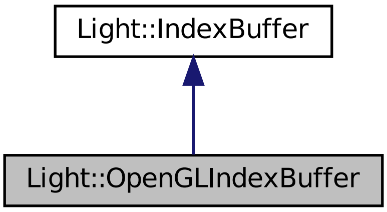
Additional Inherited Members

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

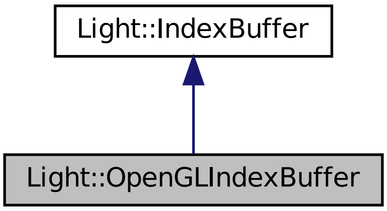
- Light/include/light/platform/opengl/openglframebuffer.hpp
- Light/src/platform/opengl/openglframebuffer.cpp

5.45 Light::OpenGLIndexBuffer Class Reference

Inheritance diagram for Light::OpenGLIndexBuffer:



Collaboration diagram for Light::OpenGLIndexBuffer:



Public Member Functions

- **OpenGLIndexBuffer** (uint32_t ∗indices, uint32_t count)
- virtual void **bind** () const override
- virtual void **unbind** () const override
- uint32_t **getCount** () const override

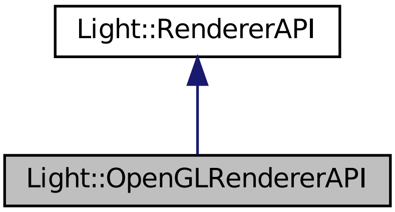
Additional Inherited Members

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

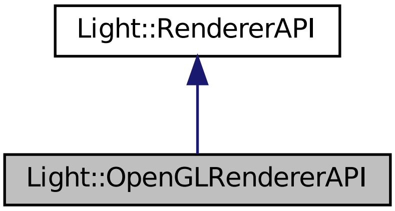
- Light/include/light/platform/opengl/openglbuffer.hpp
- Light/src/platform/opengl/openglbuffer.cpp

5.46 Light::OpenGLRendererAPI Class Reference

Inheritance diagram for Light::OpenGLRendererAPI:



Collaboration diagram for Light::OpenGLRendererAPI:



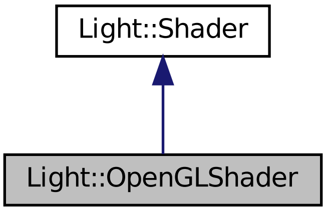
Additional Inherited Members

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

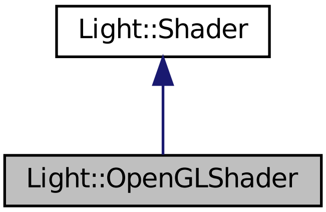
- Light/include/light/platform/opengl/openglrendererapi.hpp
- Light/src/platform/opengl/openglrendererapi.cpp

5.47 Light::OpenGLShader Class Reference

Inheritance diagram for Light::OpenGLShader:



Collaboration diagram for Light::OpenGLShader:



Public Member Functions

- **OpenGLShader** (const char ∗shaderPath)
- void **bind** () override
- void **unbind** () override
- const std::string & **getName** () const override
- void **setUniformBool** (const std::string &name, bool value) const override
- void **setUniformInt** (const std::string &name, int value) const override
- void **setUniformFloat** (const std::string &name, float value) const override
- void **setUniformVec2** (const std::string &name, const glm::vec2 &value) const override
- void **setUniformVec3** (const std::string &name, const glm::vec3 &value) const override
- void **setUniformVec4** (const std::string &name, const glm::vec4 &value) const override
- void **setUniformMat2** (const std::string &name, const glm::mat2 &mat) const override
- void **setUniformMat3** (const std::string &name, const glm::mat3 &mat) const override
- void **setUniformMat4** (const std::string &name, const glm::mat4 &mat) const override

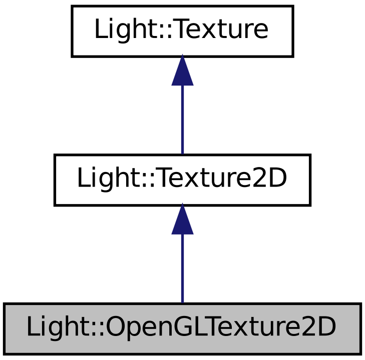
Additional Inherited Members

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

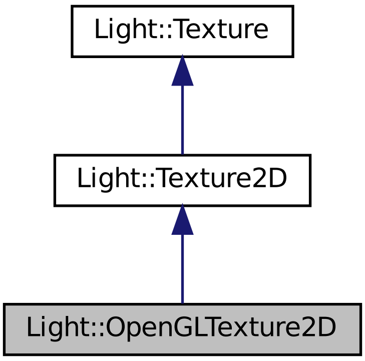
- Light/include/light/platform/opengl/openglshader.hpp
- Light/src/platform/opengl/openglshader.cpp

5.48 Light::OpenGLTexture2D Class Reference

Inheritance diagram for Light::OpenGLTexture2D:



Collaboration diagram for Light::OpenGLTexture2D:



Public Member Functions

- **OpenGLTexture2D** (const std::string &path)
- uint32_t **getRendererId** () const override
- uint32_t **getWidth** () const override
- uint32_t **getHeight** () const override
- void **bind** (uint32_t slot=0) const override

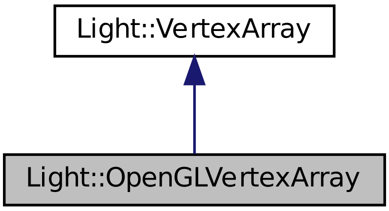
Additional Inherited Members

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

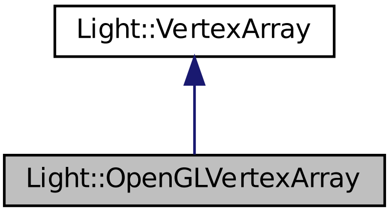
- Light/include/light/platform/opengl/opengltexture.hpp
- Light/src/platform/opengl/opengltexture.cpp

5.49 Light::OpenGLVertexArray Class Reference

Inheritance diagram for Light::OpenGLVertexArray:



Collaboration diagram for Light::OpenGLVertexArray:



Public Member Functions

- void **bind** () const override
- void **unbind** () const override
- void **addVertexBuffer** (const std::shared_ptr< VertexBuffer > &vbo) override
- void **setIndexBuffer** (const std::shared_ptr< IndexBuffer > &ibo) override
- const std::vector< std::shared_ptr< VertexBuffer > > & **getVertexBuffers** () const override
- const std::shared_ptr< IndexBuffer > & **getIndexBuffer** () const override

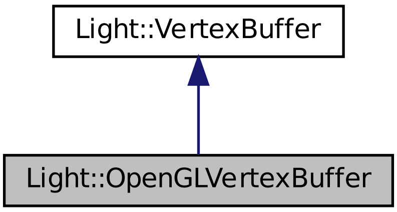
Additional Inherited Members

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

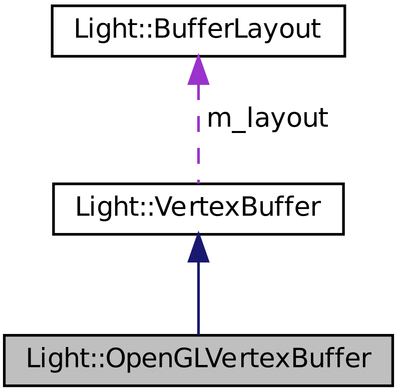
- Light/include/light/platform/opengl/openglvertexarray.hpp
- Light/src/platform/opengl/openglvertexarray.cpp

5.50 Light::OpenGLVertexBuffer Class Reference

Inheritance diagram for Light::OpenGLVertexBuffer:



Collaboration diagram for Light::OpenGLVertexBuffer:



Public Member Functions

- **OpenGLVertexBuffer** (float ∗vertices, uint32_t size)
- virtual void **bind** () const override
- virtual void **unbind** () const override

Additional Inherited Members

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

- Light/include/light/platform/opengl/openglbuffer.hpp
- Light/src/platform/opengl/openglbuffer.cpp

5.51 Physicc::PhysicsWorld Class Reference

World's Physics Class.

```
#include <physicsworld.hpp>
```

Public Member Functions

- PhysicsWorld (const glm::vec3 &gravity)

  Physics World initialisation with gravity.
- void **setGravity** (const glm::vec3 &gravity)
- glm::vec3 **getGravity** () const
- void addRigidBody (const RigidBody &object)

  Add a new RigidBody to m_objects.
- void stepSimulation (float timestep)

  steps the simulation by time timestep

5.51.1 Detailed Description

World's Physics Class.

This class describes and propagates the properties of each object using the Physics Model.

5.51.2 Constructor & Destructor Documentation

5.51.2.1 PhysicsWorld()

```
Physicc::PhysicsWorld::PhysicsWorld (
             const glm::vec3 & gravity )
```

Physics World initialisation with gravity.

This initialises the Physics World with gravity, input from the —?—.

5.51.3 Member Function Documentation

5.51.3.1 addRigidBody()

```
void Physicc::PhysicsWorld::addRigidBody (
             const RigidBody & object )
```

Add a new RigidBody to m_objects.

Parameters

object	input, const RigidBody& type

5.51.3.2 stepSimulation()

```
void Physicc::PhysicsWorld::stepSimulation (
             float time )
```

steps the simulation by time timestep

Parameters

timestep	input, float type, time interval

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

- Physicc/include/physicsworld.hpp
- Physicc/src/physicsworld.cpp

5.52 Light::RenderCommand Class Reference

Static Public Member Functions

- static void **init** ()
- static void **setViewPort** (uint32_t x, uint32_t y, uint32_t width, uint32_t height)
- static void **depthMask** (bool enable)
- static void **drawIndexed** (const std::shared_ptr< VertexArray > &vao)
- static void **clear** ()
- static void **setClearColor** (glm::vec4 color)

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

- Light/include/light/rendering/rendercommand.hpp
- Light/src/platform/opengl/openglrendererapi.cpp

5.53 Light::Renderer Class Reference

Static Public Member Functions

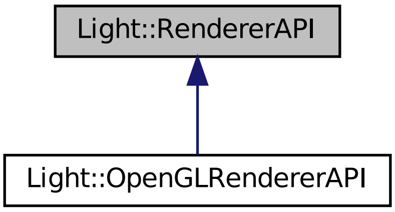
- static void **init** ()
- static void **onWindowResize** (uint32_t width, uint32_t height)
- static void **beginScene** (Camera &camera, glm::mat4 camera_view)
- static void **endScene** ()
- static void **submitLight** (glm::vec3 lightPos, glm::vec3 lightCol)
- static void **submit** (const std::shared_ptr< Shader > &shader, const std::shared_ptr< VertexArray > &vao, glm::mat4 transform=glm::mat4(1.0f))
- static void **submitID** (const std::shared_ptr< Shader > &shader, const std::shared_ptr< VertexArray > &vao, glm::mat4 transform=glm::mat4(1.0f), int id=-1)
- static void **submitSkybox** (const std::shared_ptr< Shader > &shader, const std::shared_ptr< VertexArray > &vao)

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

- Light/include/light/rendering/renderer.hpp
- Light/src/rendering/renderer.cpp

5.54 Light::RendererAPI Class Reference

Inheritance diagram for Light::RendererAPI:



Public Member Functions

- virtual void **init** ()=0
- virtual void **depthMask** (bool enable)=0
- virtual void **setViewPort** (uint32_t x, uint32_t y, uint32_t width, uint32_t height)=0
- virtual void **setClearColor** (glm::vec4 &color)=0
- virtual void **clear** ()=0
- virtual void **drawIndexed** (const std::shared_ptr< VertexArray > &vao)=0

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

- Light/include/light/rendering/rendererapi.hpp

5.55 Physicc::RigidBody Class Reference

Rigid Body Class.

```
#include <rigidbody.hpp>
```

Public Member Functions

- RigidBody (float mass, const glm::vec3 &velocity, float gravityScale)

  RigidBody initialized with a mass velocity, and a bool storing whether gravity is acting on the object or not.
- glm::vec3 **getVelocity** () const
- void **setVelocity** (const glm::vec3 &velocity)
- void **setGravityScale** (const float gravityScale)
- void **setForce** ()
- BoundingVolume::AABB **getAABB** () const

Friends

- class **PhysicsWorld**

5.55.1 Detailed Description

Rigid Body Class.

This class describes and propagates the properties of each Rigid Body.

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

- Physicc/include/rigidbody.hpp
- Physicc/src/rigidbody.cpp

5.56 Light::Scene Class Reference

Public Member Functions

- Entity **addEntity** (const std::string &name="")
- void **update** (Light::Timestep dt)
- void **render** ()
- void **renderSelection** (Entity entity)
- void **renderOutline** (Entity entity)

Friends

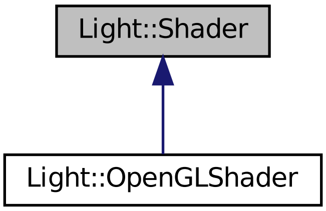
- class **Entity**
- class **ScenePanel**

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

- LightFramework/include/ecs/scene.hpp
- LightFramework/src/ecs/scene.cpp

5.57 Light::Shader Class Reference

Inheritance diagram for Light::Shader:



Public Member Functions

- virtual const std::string & **getName** () const =0
- virtual void **bind** ()=0
- virtual void **unbind** ()=0
- virtual void **setUniformBool** (const std::string &name, bool value) const =0
- virtual void **setUniformInt** (const std::string &name, int value) const =0
- virtual void **setUniformFloat** (const std::string &name, float value) const =0
- virtual void **setUniformVec2** (const std::string &name, const glm::vec2 &value) const =0
- virtual void **setUniformVec3** (const std::string &name, const glm::vec3 &value) const =0
- virtual void **setUniformVec4** (const std::string &name, const glm::vec4 &value) const =0
- virtual void **setUniformMat2** (const std::string &name, const glm::mat2 &mat) const =0
- virtual void **setUniformMat3** (const std::string &name, const glm::mat3 &mat) const =0
- virtual void **setUniformMat4** (const std::string &name, const glm::mat4 &mat) const =0

Static Public Member Functions

- static std::shared_ptr< Shader > **create** (const char ∗shaderPath)

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

- Light/include/light/rendering/shader.hpp
- Light/src/platform/opengl/openglshader.cpp

5.58 Light::ShaderLibrary Class Reference

Public Member Functions

- void **add** (const std::shared_ptr< Shader > &shader)
- void **add** (const std::string &name, const std::shared_ptr< Shader > &shader)
- std::shared_ptr< Shader > **load** (const std::string &filepath)
- std::shared_ptr< Shader > **load** (const std::string &name, const std::string &filepath)
- std::shared_ptr< Shader > **get** (const std::string &name)

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

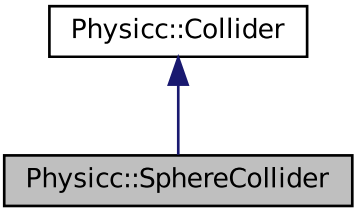
- Light/include/light/rendering/shader.hpp
- Light/src/rendering/shader.cpp

5.59 Physicc::SphereCollider Class Reference

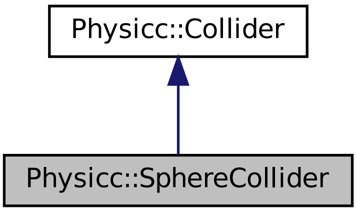
SphereCollider class.

```
#include <collider.hpp>
```

Inheritance diagram for Physicc::SphereCollider:



Collaboration diagram for Physicc::SphereCollider:



Public Member Functions

- SphereCollider (float radius=1.0f, glm::vec3 position=glm::vec3(0), glm::vec3 rotation=glm::vec3(0), glm::vec3 scale=glm::vec3(1))

  Creates a SphereCollider object.
- BoundingVolume::AABB getAABB () const override

  Computes and returns Axis Aligned Bounding Box of Sphere shaped object.

Additional Inherited Members

5.59.1 Detailed Description

SphereCollider class.

Sphere shaped collider, holds the radius and transform of the body

5.59.2 Constructor & Destructor Documentation

5.59.2.1 SphereCollider()

```
Physicc::SphereCollider::SphereCollider (
            float radius = 1.0f,
            glm::vec3 position = glm::vec3(0),
            glm::vec3 rotation = glm::vec3(0),
            glm::vec3 scale = glm::vec3(1) )
```

Creates a SphereCollider object.

Parameters

radius	Radius of the sphere
position	Position of object in global space
rotation	Rotation about each of the axis in local space
scale	Scale of the object along each axis

5.59.3 Member Function Documentation

5.59.3.1 getAABB()

```
BoundingVolume::AABB Physicc::SphereCollider::getAABB ( ) const  [override], [virtual]
```

Computes and returns Axis Aligned Bounding Box of Sphere shaped object.

Returns

BoundingVolume::AABB

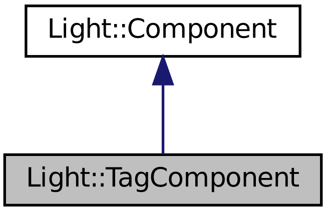
Implements Physicc::Collider.

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

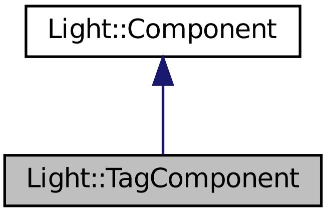
- Physicc/include/collider.hpp
- Physicc/src/collider.cpp

5.60 Light::TagComponent Struct Reference

Inheritance diagram for Light::TagComponent:



Collaboration diagram for Light::TagComponent:



Public Member Functions

- **TagComponent** (const TagComponent &)=default
- **TagComponent** (const std::string &tag)

Public Attributes

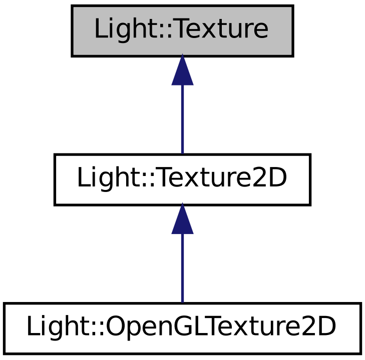
- std::string **tag**

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

- LightFramework/include/ecs/components.hpp

5.61 Light::Texture Class Reference

Inheritance diagram for Light::Texture:



Public Member Functions

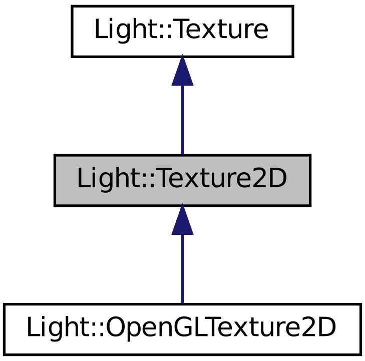
- virtual uint32_t **getWidth** () const =0
- virtual uint32_t **getHeight** () const =0
- virtual void **bind** (uint32_t slot=0) const =0

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

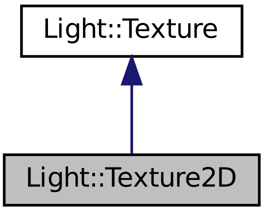
- Light/include/light/rendering/texture.hpp

5.62 Light::Texture2D Class Reference

Inheritance diagram for Light::Texture2D:



Collaboration diagram for Light::Texture2D:



Public Member Functions

- virtual uint32_t **getRendererId** () const =0

Static Public Member Functions

- static Texture2D ∗ **create** (const std::string &path)

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

- Light/include/light/rendering/texture.hpp
- Light/src/platform/opengl/opengltexture.cpp

5.63 Light::Timestep Class Reference

Public Member Functions

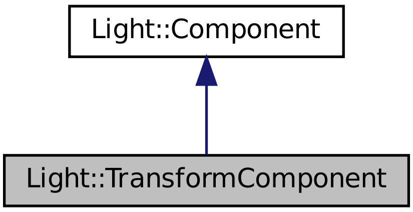
- **Timestep** (float time)
- float **getSeconds** ()
- float **getMilliSeconds** ()

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

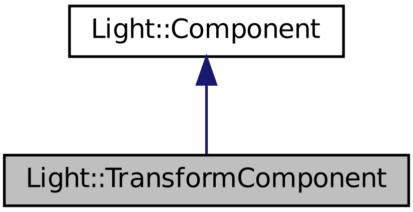
- LightFramework/include/core/timestep.hpp

5.64 Light::TransformComponent Struct Reference

Inheritance diagram for Light::TransformComponent:



Collaboration diagram for Light::TransformComponent:



Public Member Functions

- **TransformComponent** (glm::vec3 position=glm::vec3(0, 0, 0), glm::vec3 rotation=glm::vec3(0, 0, 0), glm↩ ::vec3 scale=glm::vec3(0.5))
- glm::mat4 **getTransform** () const

Public Attributes

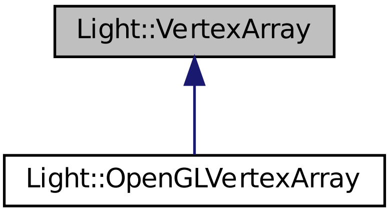
- glm::vec3 **position**
- glm::vec3 **rotation**
- glm::vec3 **scale**

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

- LightFramework/include/ecs/components.hpp

5.65 Light::VertexArray Class Reference

Inheritance diagram for Light::VertexArray:



Public Member Functions

- virtual void **bind** () const =0
- virtual void **unbind** () const =0
- virtual void **addVertexBuffer** (const std::shared_ptr< VertexBuffer > &vbo)=0
- virtual void **setIndexBuffer** (const std::shared_ptr< IndexBuffer > &ibo)=0
- virtual const std::vector< std::shared_ptr< VertexBuffer > > & **getVertexBuffers** () const =0
- virtual const std::shared_ptr< IndexBuffer > & **getIndexBuffer** () const =0

Static Public Member Functions

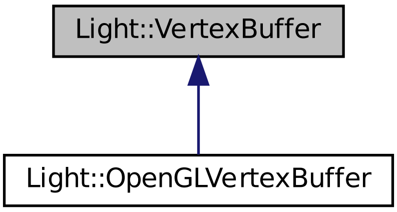
- static VertexArray ∗ **create** ()

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

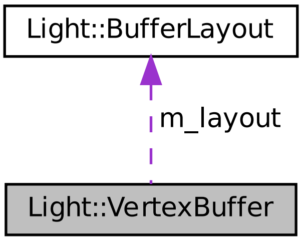
- Light/include/light/rendering/vertexarray.hpp
- Light/src/platform/opengl/openglvertexarray.cpp

5.66 Light::VertexBuffer Class Reference

Inheritance diagram for Light::VertexBuffer:



Collaboration diagram for Light::VertexBuffer:



Public Member Functions

- virtual void **bind** () const =0
- virtual void **unbind** () const =0
- void **setLayout** (BufferLayout layout)
- const BufferLayout & **getLayout** ()

Static Public Member Functions

- static VertexBuffer ∗ **create** (float ∗vertices, uint32_t size)

Protected Attributes

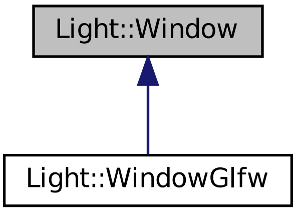
- BufferLayout **m_layout**

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

- Light/include/light/rendering/buffer.hpp
- Light/src/platform/opengl/openglbuffer.cpp

5.67 Light::Window Class Reference

Inheritance diagram for Light::Window:



Public Types

- using **EventCallbackFn** = std::function< void(Event &)>

Public Member Functions

- virtual void **onUpdate** ()=0
- virtual uint32_t **getWidth** () const =0
- virtual uint32_t **getHeight** () const =0
- virtual void **setEventCallback** (const EventCallbackFn &callback)=0
- virtual void **setVSync** (bool enabled)=0
- virtual bool **isVSync** () const =0
- virtual void ∗ **getNativeWindow** () const =0

Static Public Member Functions

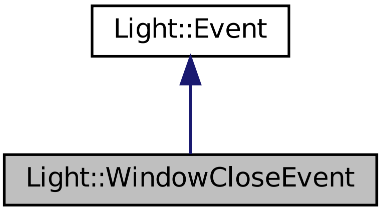
- static Window ∗ **create** (const WindowProps &props=WindowProps())

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

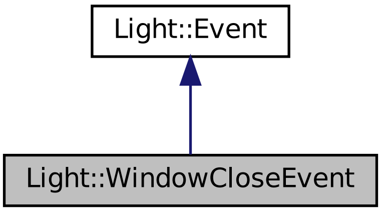
- LightFramework/include/core/window.hpp
- LightFramework/src/platform/glfw/windowglfw.cpp

5.68 Light::WindowCloseEvent Class Reference

Inheritance diagram for Light::WindowCloseEvent:



Collaboration diagram for Light::WindowCloseEvent:



Public Member Functions

- std::string **ToString** () const override

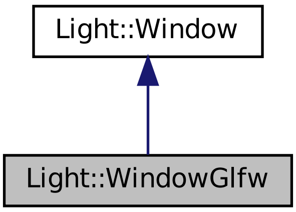
Additional Inherited Members

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

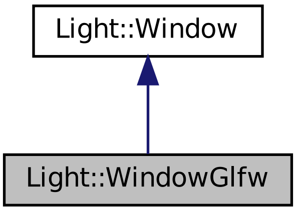
- LightFramework/include/events/applicationevent.hpp

5.69 Light::WindowGlfw Class Reference

Inheritance diagram for Light::WindowGlfw:



Collaboration diagram for Light::WindowGlfw:



Public Member Functions

- **WindowGlfw** (const WindowProps &props)
- void **onUpdate** () override
- virtual uint32_t **getWidth** () const override
- virtual uint32_t **getHeight** () const override
- virtual void **setEventCallback** (const EventCallbackFn &callback) override
- virtual void **setVSync** (bool enabled) override
- virtual bool **isVSync** () const override
- virtual void ∗ **getNativeWindow** () const override

Additional Inherited Members

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

- LightFramework/include/platform/glfw/windowglfw.hpp
- LightFramework/src/platform/glfw/windowglfw.cpp

5.70 Light::WindowProps Struct Reference

Public Member Functions

- **WindowProps** (const std::string title="Light Engine", uint32_t width=1600, uint32_t height=900)

Public Attributes

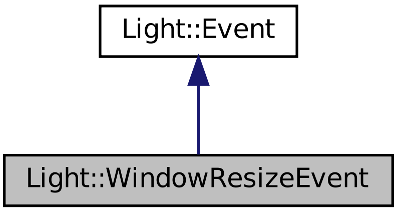
- std::string **title**
- uint32_t **width**
- uint32_t **height**

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

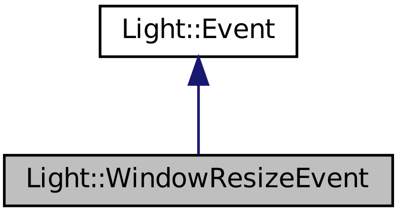
- LightFramework/include/core/window.hpp

5.71 Light::WindowResizeEvent Class Reference

Inheritance diagram for Light::WindowResizeEvent:



Collaboration diagram for Light::WindowResizeEvent:



Public Member Functions

- **WindowResizeEvent** (int x, int y)
- std::tuple< int, int > **getSize** ()
- std::string **ToString** () const override

Additional Inherited Members

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

- LightFramework/include/events/applicationevent.hpp

Chapter 6

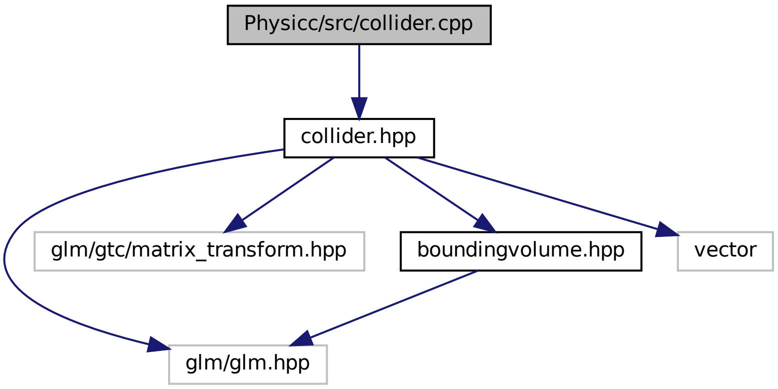
File Documentation

6.1 Physicc/src/collider.cpp File Reference

Contains the collider classes.

```
#include "collider.hpp"
```

Include dependency graph for collider.cpp:



6.1.1 Detailed Description

Contains the collider classes.

The Collider file contains the collider classes which hold the shape and transform of the objects

Author

Prakhar Mittal (prak74)

Tirthankar Mazumder (wermos)

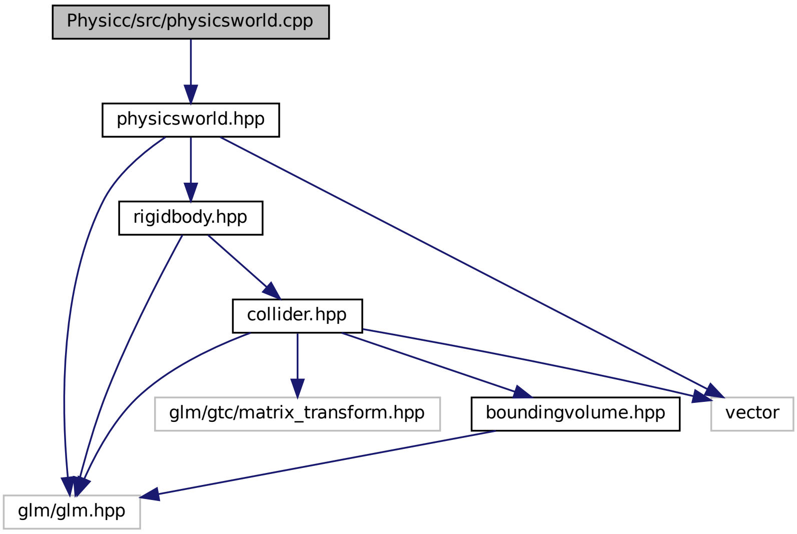
Bug No known bugs.

6.2 Physicc/src/physicsworld.cpp File Reference

The Physics World.

```
#include "physicsworld.hpp"
```

Include dependency graph for physicsworld.cpp:



6.2.1 Detailed Description

The Physics World.

The Physics World file class conatins the Physics Model for the Physics Engine. This includes the Gravity Model, Rigid Body objects.

Author

Divyansh Tiwari (divyanshtiwari237)

Neilabh Banzal (Neilabh21)

Tirthankar Mazumder (wermos)

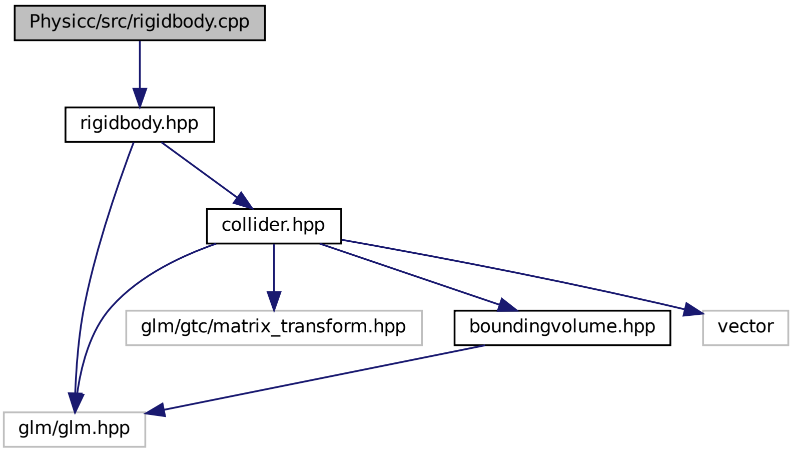
Bug No known bugs.

6.3 Physicc/src/rigidbody.cpp File Reference

Defines a Rigid Body.

```
#include "rigidbody.hpp"
```

Include dependency graph for rigidbody.cpp:



6.3.1 Detailed Description

Defines a Rigid Body.

Author

Divyansh Tiwari (divyanshtiwari237)

Neilabh Banzal (Neilabh21)

Tirthankar mazumder (wermos)

Bug No known bugs.

Chapter 7

Example Documentation

7.1 Use

First index of Depth Type attachment like if(fmt < FramebufferAttachmentFormat::DepthType) { // Code for color buffers }

Index