Light

0.0.1

Generated by Doxygen 1.8.17

1 Bug List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Physicc::BVImpl::AABB Struct Reference	9
	9
•	9
5.3 Physicc::BVImpl::BaseBV< Derived, BoundingObject > Class Template Reference	0
5.3.1 Detailed Description	0
5.3.2 Constructor & Destructor Documentation	0
5.3.2.1 BaseBV() [1/2]	
5.3.2.2 BaseBV() [2/2]	
5.3.3 Member Function Documentation	
5.3.3.1 overlapsWith()	
5.4 Physicc::BVImpl::BoxBV < T > Class Template Reference	
5.5 Physicc::BoxCollider Class Reference	
5.5.1 Detailed Description	
5.5.2 Constructor & Destructor Documentation	
5.5.3 Member Function Documentation	
5.5.3.1 getAABB()	
5.6 Light::BufferElement Struct Reference	
5.7 Light::BufferLayout Class Reference	
5.8 Physicc::BVH Class Reference	6
5.9 Physicc::BVHNode Struct Reference	6
5.10 Light::Camera Class Reference	7
5.11 Light::CameraComponent Struct Reference	7
5.12 Physicc::Collider Class Reference	8
5.12.1 Detailed Description	9
5.12.2 Constructor & Destructor Documentation	9
5.12.2.1 Collider()	9
5.12.3 Member Function Documentation	0
5.12.3.1 getPosition()	0
5.12.3.2 getRotate()	0
5.12.3.3 getScale()	0
5.12.3.4 getTransform()	1

5.12.3.5 setPosition()	21
5.12.3.6 setRotate()	21
5.12.3.7 setScale()	21
5.12.3.8 updateTransform()	22
5.13 Light::Component Struct Reference	22
5.14 Light::Cubemap Class Reference	23
5.15 Light::EditorCamera Class Reference	23
5.16 Light::Entity Class Reference	24
5.17 Light::Event Class Reference	25
5.18 Light::EventDispatcher Class Reference	26
5.19 Light::Framebuffer Class Reference	26
5.20 Light::FramebufferAttachmentsSpec Struct Reference	27
5.21 Light::FramebufferSpec Struct Reference	27
5.22 Light::FramebufferTextureSpec Struct Reference	28
5.23 Light::GraphicsContext Class Reference	28
5.24 Light::ImguiLayer Class Reference	20
5.25 Light::IndexBuffer Class Reference	30
5.26 Light::Input Class Reference	30
5.27 Light::InputGlfw Class Reference	31
5.28 Light::KeyEvent Class Reference	32
5.29 Light::KeyPressedEvent Class Reference	33
5.30 Light::KeyReleasedEvent Class Reference	}4
5.31 Light::KeyTypedEvent Class Reference	35
5.32 Light::Layer Class Reference	36
5.33 Light::LayerStack Class Reference	37
5.34 Light::LightComponent Struct Reference	38
5.35 Light::MeshComponent Struct Reference	39
5.36 Light::MeshRendererComponent Struct Reference	10
5.37 Light::MouseButtonEvent Class Reference	11
5.38 Light::MouseButtonPressedEvent Class Reference	12
5.39 Light::MouseButtonReleasedEvent Class Reference	13
5.40 Light::MouseMovedEvent Class Reference	14
5.41 Light::MouseScrolledEvent Class Reference	15
5.42 Light::OpenGLContext Class Reference	16
5.43 Light::OpenGLCubemap Class Reference	17
5.44 Light::OpenGLFramebuffer Class Reference	18
5.45 Light::OpenGLIndexBuffer Class Reference	19
5.46 Light::OpenGLRendererAPI Class Reference	50
5.47 Light::OpenGLShader Class Reference	51
5.48 Light::OpenGLTexture2D Class Reference	52
5.49 Light::OpenGLVertexArray Class Reference	53
5.50 Light::OpenGLVertexBuffer Class Reference	54

5.51 Physics::PhysicsWorld Class Reference	55
5.51.1 Detailed Description	55
5.51.2 Constructor & Destructor Documentation	55
5.51.2.1 PhysicsWorld()	55
5.51.3 Member Function Documentation	56
5.51.3.1 addRigidBody()	56
5.51.3.2 stepSimulation()	56
5.52 Light::RenderCommand Class Reference	56
5.53 Light::Renderer Class Reference	57
5.54 Light::RendererAPI Class Reference	57
5.55 Physicc::RigidBody Class Reference	58
5.55.1 Detailed Description	58
5.56 Light::Scene Class Reference	58
5.57 Light::Shader Class Reference	59
5.58 Light::ShaderLibrary Class Reference	59
5.59 Physicc::SphereCollider Class Reference	60
5.59.1 Detailed Description	60
5.59.2 Constructor & Destructor Documentation	61
5.59.2.1 SphereCollider()	61
5.59.3 Member Function Documentation	61
5.59.3.1 getAABB()	61
5.60 Light::TagComponent Struct Reference	62
5.61 Light::Texture Class Reference	63
5.62 Light::Texture2D Class Reference	63
5.63 Light::Timestep Class Reference	64
5.64 Light::TransformComponent Struct Reference	65
5.65 Light::VertexArray Class Reference	66
5.66 Light::VertexBuffer Class Reference	66
5.67 Light::Window Class Reference	67
5.68 Light::WindowCloseEvent Class Reference	68
5.69 Light::WindowGlfw Class Reference	69
5.70 Light::WindowProps Struct Reference	70
5.71 Light::WindowResizeEvent Class Reference	70
6 File Documentation	73
6.1 Physicc/src/collider.cpp File Reference	73
6.1.1 Detailed Description	73
6.2 Physicc/src/physicsworld.cpp File Reference	74
6.2.1 Detailed Description	74
6.3 Physicc/src/rigidbody.cpp File Reference	74 74
6.3.1 Detailed Description	7 <del>4</del> 75
C.C. Potanou Pescription	, 5
7 Example Documentation	77

7.1 Use	 	 	 	 		 		 			 		 	 77
Index														79

## **Chapter 1**

# **Bug List**

File collider.cpp

No known bugs.

File physicsworld.cpp

No known bugs.

File rigidbody.cpp

No known bugs.

2 Bug List

# **Chapter 2**

# **Hierarchical Index**

## 2.1 Class Hierarchy

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

Physicc::BVImpl::AABB
Light::Application
Physicc::BVImpl::BaseBV < Derived, BoundingObject >
$Physicc::BVImpl::BaseBV < BoxBV < T >, T > \dots \dots$
Physicc::BVImpl::BoxBV < T >
Physicc::BVImpl::BaseBV< BVImpl::BoxBV< BVImpl::AABB > , BVImpl::AABB >
Light::BufferElement
Light::BufferLayout
Physicc::BVH
Physicc::BVHNode
Light::Camera
Light::EditorCamera
Physicc::Collider
Physicc::BoxCollider
Physicc::SphereCollider
Light::Component
Light::CameraComponent
Light::LightComponent
Light::MeshComponent
Light::MeshRendererComponent
Light::TagComponent
Light::TransformComponent
Light::Cubemap
Light::OpenGLCubemap
Light::Entity
Light::Event
Light::KeyEvent
Light::KeyPressedEvent
Light::KeyReleasedEvent
Light::KeyTypedEvent
Light::MouseButtonEvent
Light::MouseButtonPressedEvent
Light::MouseButtonReleasedEvent

4 Hierarchical Index

Light::MouseMovedEvent	. 45 . 68
Light::EventDispatcher	26
Light::OpenGLFramebuffer	. 48
Light::FramebufferAttachmentsSpec	27
Light::FramebufferSpec	27
Light::FramebufferTextureSpec	
Light::GraphicsContext	28
Light::OpenGLContext	. 46
Light::IndexBuffer	30
Light::OpenGLIndexBuffer	. 49
Light::Input	30
Light::InputGlfw	. 31
Light::Layer	36
Light::ImguiLayer	
Light::LayerStack	
Physics: Physics World	
Light::RenderCommand	
Light::Renderer	
Light::RendererAPI	57
Light::OpenGLRendererAPI	. 50
Physicc::RigidBody	58
Light::Scene	58
Light::Shader	59
Light::OpenGLShader	. 51
Light::ShaderLibrary	59
Light::Texture	63
Light::Texture2D	. 63
Light::OpenGLTexture2D	. 52
Light::Timestep	64
Light::VertexArray	66
Light::OpenGLVertexArray	. 53
Light::VertexBuffer	66
Light::OpenGLVertexBuffer	. 54
Light::Window	67
Light::WindowGlfw	
Light::WindowProps	70
g	, 0

# **Chapter 3**

# **Class Index**

## 3.1 Class List

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

Physice::Bvimpi::AABB
Axis Aligned Bounding Box
Light::Application
Physicc::BVImpl::BaseBV< Derived, BoundingObject >
Physicc::BVImpl::BoxBV< T >
Physicc::BoxCollider
BoxCollider class
Light::BufferElement
Light::BufferLayout
Physicc::BVH
Physicc::BVHNode
Light::Camera
Light::CameraComponent
Physicc::Collider
Collider class
Light::Component
Light::Cubemap
Light::EditorCamera
Light::Entity
Light::Event
Light::EventDispatcher
Light::Framebuffer
Light::FramebufferAttachmentsSpec
Light::FramebufferSpec
Light::FramebufferTextureSpec
Light::GraphicsContext
Light::ImguiLayer
Light::IndexBuffer
Light::Input
Light::InputGlfw
Light::KeyEvent
Light::KeyPressedEvent
Light::KeyReleasedEvent
Light::KeyTypedEvent
Light::Layer

6 Class Index

Light::LayerStack	7
Light::LightComponent	
Light::MeshComponent	
Light::MeshRendererComponent	
Light::MouseButtonEvent	
Light::MouseButtonPressedEvent	2
Light::MouseButtonReleasedEvent	3
Light::MouseMovedEvent	4
Light::MouseScrolledEvent	5
Light::OpenGLContext	ô
Light::OpenGLCubemap	7
Light::OpenGLFramebuffer	
Light::OpenGLIndexBuffer	Э
Light::OpenGLRendererAPI	J
Light::OpenGLShader	1
Light::OpenGLTexture2D	2
Light::OpenGLVertexArray	3
Light::OpenGLVertexBuffer	4
Physicc::PhysicsWorld	
World's Physics Class	5
Light::RenderCommand	ô
Light::Renderer	7
Light::RendererAPI	7
Physicc::RigidBody	
Rigid Body Class	8
Light::Scene 58	8
Light::Shader	Э
Light::ShaderLibrary	Э
Physicc::SphereCollider	
SphereCollider class	J
Light::TagComponent	2
Light::Texture	3
Light::Texture2D	3
Light::Timestep	4
Light::TransformComponent	5
Light::VertexArray	ô
Light::VertexBuffer	ô
Light::Window	7
Light::WindowCloseEvent	8
Light::WindowGlfw	9
Light::WindowProps	J
Light::WindowResizeEvent 70	n

# **Chapter 4**

# File Index

## 4.1 File List

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

Light/include/light/platform/opengl/openglbuffer.hpp
Light/include/light/platform/opengl/openglcontext.hpp
Light/include/light/platform/opengl/openglframebuffer.hpp
Light/include/light/platform/opengl/openglrendererapi.hpp
Light/include/light/platform/opengl/openglshader.hpp
Light/include/light/platform/opengl/opengltexture.hpp
Light/include/light/platform/opengl/openglvertexarray.hpp
Light/include/light/rendering/buffer.hpp
Light/include/light/rendering/camera.hpp
Light/include/light/rendering/framebuffer.hpp
Light/include/light/rendering/graphicscontext.hpp
Light/include/light/rendering/rendercommand.hpp
Light/include/light/rendering/renderer.hpp
Light/include/light/rendering/rendererapi.hpp
Light/include/light/rendering/shader.hpp
Light/include/light/rendering/texture.hpp
Light/include/light/rendering/vertexarray.hpp
LightFramework/include/light.hpp
LightFramework/include/core/application.hpp
LightFramework/include/core/entrypoint.hpp
LightFramework/include/core/input.hpp
LightFramework/include/core/layer.hpp
LightFramework/include/core/layerstack.hpp
LightFramework/include/core/timestep.hpp
LightFramework/include/core/ <b>uuid.hpp</b>
LightFramework/include/core/window.hpp
LightFramework/include/ecs/components.hpp
LightFramework/include/ecs/entity.hpp
LightFramework/include/ecs/scene.hpp
LightFramework/include/events/applicationevent.hpp
LightFramework/include/events/event.hpp
LightFramework/include/events/keyevent.hpp
LightFramework/include/events/mouseevent.hpp
LightFramework/include/imgui/imguilayer.hpp
LightFramework/include/input/keycodes.hpp

8 File Index

_ightFramework/include/input/ <b>mousecodes.hpp</b>	??
_ightFramework/include/platform/glfw/ <b>inputglfw.hpp</b>	??
_ightFramework/include/platform/glfw/ <b>windowglfw.hpp</b>	??
_ightFramework/include/rendering/ <b>editorcamera.hpp</b>	??
Physicc/include/boundingvolume.hpp	??
Physicc/include/bvh.hpp	??
Physicc/include/collider.hpp	??
Physicc/include/physicsworld.hpp	??
Physicc/include/rigidbody.hpp	??
Physicc/src/collider.cpp	
Contains the collider classes	73
Physicc/src/physicsworld.cpp	
The Physics World	74
Physicc/src/rigidbody.cpp	
Defines a Rigid Body	74

## **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]

Copy constructor for BoundingVolume.

## **Template Parameters**

```
BV The object to be copied
```

#### **Parameters**

```
bv A BV object
```

## 5.3.2.2 BaseBV() [2/2]

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()

Returns whether two BVs are overlapping or not.

#### **Parameters**

bv A BaseBV object

#### Returns

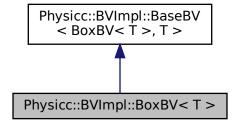
true if the BoundingVolumes are intersecting, and false otherwise TODO: Is this, as a return type description, fine?

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

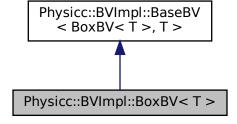
• Physicc/include/boundingvolume.hpp

## 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:

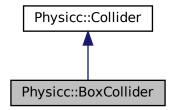
• Physicc/include/boundingvolume.hpp

## 5.5 Physicc::BoxCollider Class Reference

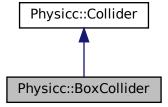
BoxCollider class.

#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**

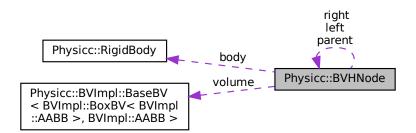
- BVH (std::vector< RigidBody > rigidBodyList)
- void buildTree ()
- std::vector< RigidBody > & convert ()

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**

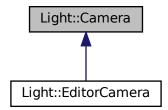
- BoundingVolume::AABB volume
- RigidBody \* body = nullptr
- BVHNode \* parent = nullptr
- BVHNode \* left = nullptr
- BVHNode \* right = nullptr

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**

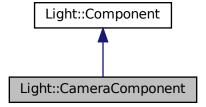
• glm::mat4 m\_projectionMatrix

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

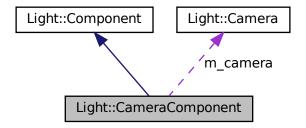
• Light/include/light/rendering/camera.hpp

## 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:

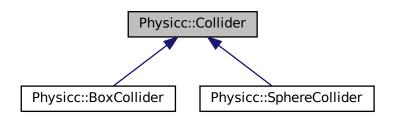
• LightFramework/include/ecs/components.hpp

## 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()

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()

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()

Set rotation of object about it's center.

**Parameters** 

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

## 5.12.3.7 setScale()

get Position of object's center

**Parameters** 

scale New scale of the object

## 5.12.3.8 updateTransform()

void Physicc::Collider::updateTransform ( )

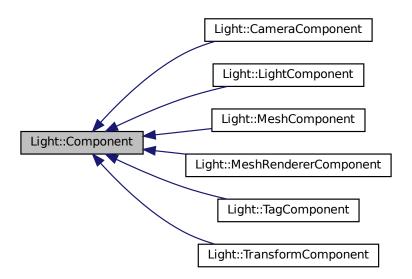
Update Transform for rendering.

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**

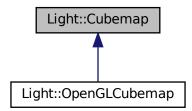
std::string uuid

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**

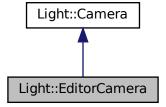
• static Cubemap \* create (const std::string &path)

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

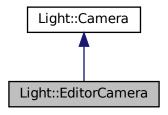
- Light/include/light/rendering/texture.hpp
- Light/src/platform/opengl/opengltexture.cpp

## 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
- $\bullet \quad template {<} typename \ T \ , \ typename ... \ Args {>}$ 
  - T & addComponent (Args... args)
- template<typename T >
- bool hasComponent ()
- template<typename T >
  - T & getComponent ()
- template<typename T >
- void removeComponent ()

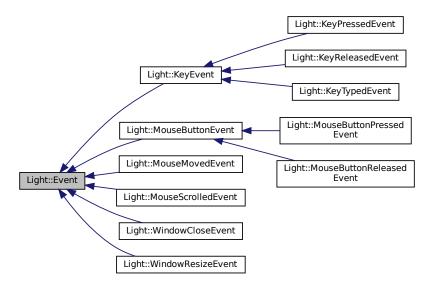
- · operator bool ()
- operator uint32\_t ()
- bool operator== (const Entity &other)
- std::string getUUID ()

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**

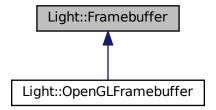
- EventDispatcher (Event &event)
- template < typename T, typename F > bool **Dispatch** (const F & func)

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**

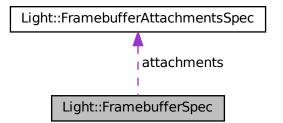
std::vector< FramebufferTextureSpec > attachments

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**

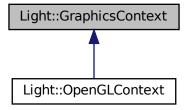
- FramebufferTextureFormat textureFormat = FramebufferTextureFormat::None
- TextureWrap wrapFormat = TextureWrap::None

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**

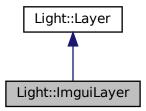
- virtual void init ()=0
- virtual void swapBuffers ()=0

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

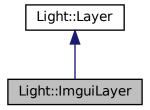
• Light/include/light/rendering/graphicscontext.hpp

## 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 onlmguiRender () override
- · void begin ()
- · void end ()

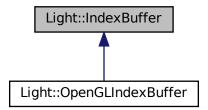
## **Additional Inherited Members**

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**

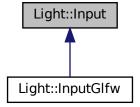
• static IndexBuffer \* create (uint32\_t \*indices, uint32\_t count)

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**

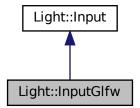
- virtual bool isKeyPressedImpI (int keycode)=0
- virtual bool isMouseButtonPressedImpI (int button)=0
- virtual std::tuple< float, float > getMousePosImpl ()=0

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

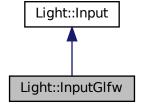
- LightFramework/include/core/input.hpp
- LightFramework/src/platform/glfw/inputglfw.cpp

# 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 isMouseButtonPressedImpI (int button) override
- std::tuple< float, float > getMousePosImpI () override

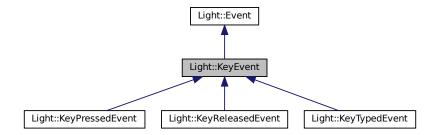
#### **Additional Inherited Members**

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

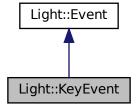
- · LightFramework/include/platform/glfw/inputglfw.hpp
- LightFramework/src/platform/glfw/inputglfw.cpp

# 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

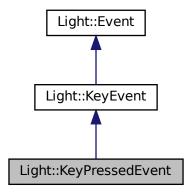
## **Additional Inherited Members**

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

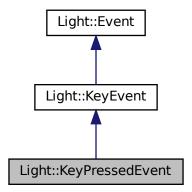
• LightFramework/include/events/keyevent.hpp

# 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

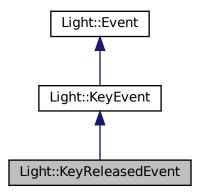
## **Additional Inherited Members**

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

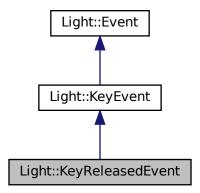
· LightFramework/include/events/keyevent.hpp

# 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

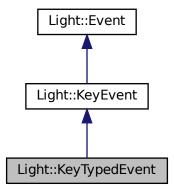
### **Additional Inherited Members**

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

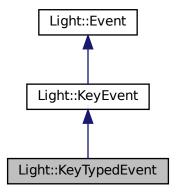
LightFramework/include/events/keyevent.hpp

# 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

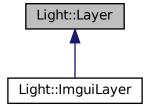
## **Additional Inherited Members**

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 onlmguiRender ()
- 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

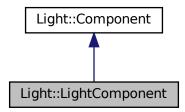
### **Public Member Functions**

- 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 ()

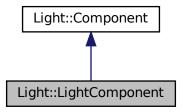
- · LightFramework/include/core/layerstack.hpp
- LightFramework/src/core/layerstack.cpp

# 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**

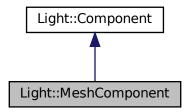
• glm::vec3 m\_lightColor

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

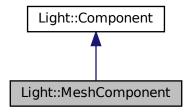
• LightFramework/include/ecs/components.hpp

# 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)

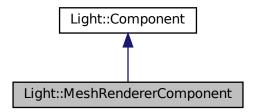
### **Public Attributes**

std::shared\_ptr< Light::VertexArray > mesh

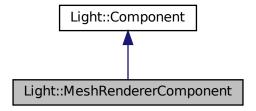
- · LightFramework/include/ecs/components.hpp
- LightFramework/src/ecs/components.cpp

# 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)

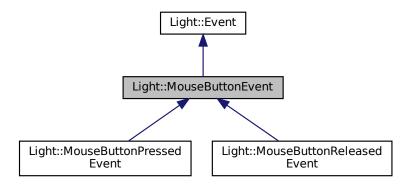
## **Public Attributes**

• std::shared\_ptr< Light::Shader > shader

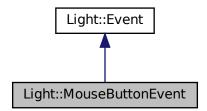
- · LightFramework/include/ecs/components.hpp
- LightFramework/src/ecs/components.cpp

# 5.37 Light::MouseButtonEvent Class Reference

Inheritance diagram for Light::MouseButtonEvent:



Collaboration diagram for Light::MouseButtonEvent:



## **Public Member Functions**

• int getButton ()

## **Protected Member Functions**

• MouseButtonEvent (int button)

### **Protected Attributes**

int button

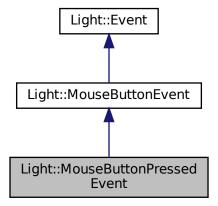
## **Additional Inherited Members**

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

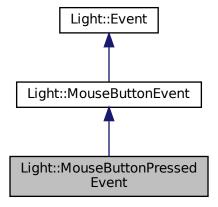
• LightFramework/include/events/mouseevent.hpp

# 5.38 Light::MouseButtonPressedEvent Class Reference

Inheritance diagram for Light::MouseButtonPressedEvent:



 $Collaboration\ diagram\ for\ Light:: Mouse Button Pressed Event:$ 



## **Public Member Functions**

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

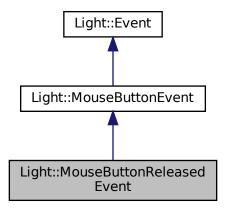
## **Additional Inherited Members**

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

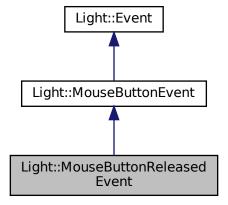
· LightFramework/include/events/mouseevent.hpp

# 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

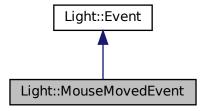
### **Additional Inherited Members**

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

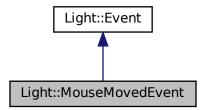
· LightFramework/include/events/mouseevent.hpp

# 5.40 Light::MouseMovedEvent Class Reference

Inheritance diagram for Light::MouseMovedEvent:



 $Collaboration\ diagram\ for\ Light:: Mouse Moved Event:$ 



## **Public Member Functions**

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

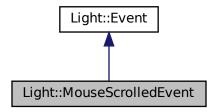
## **Additional Inherited Members**

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

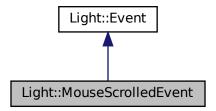
LightFramework/include/events/mouseevent.hpp

# 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

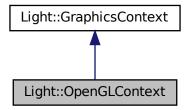
### **Additional Inherited Members**

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

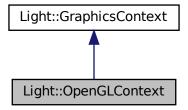
LightFramework/include/events/mouseevent.hpp

# 5.42 Light::OpenGLContext Class Reference

Inheritance diagram for Light::OpenGLContext:



Collaboration diagram for Light::OpenGLContext:



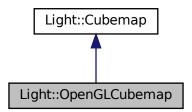
## **Public Member Functions**

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

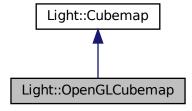
- Light/include/light/platform/opengl/openglcontext.hpp
- Light/src/platform/opengl/openglcontext.cpp

# 5.43 Light::OpenGLCubemap Class Reference

Inheritance diagram for Light::OpenGLCubemap:



Collaboration diagram for Light::OpenGLCubemap:



## **Public Member Functions**

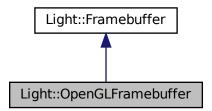
- OpenGLCubemap (const std::string &path)
- void bind (uint32\_t slot=0) const override

### **Additional Inherited Members**

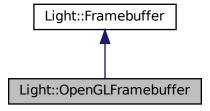
- Light/include/light/platform/opengl/opengltexture.hpp
- Light/src/platform/opengl/opengltexture.cpp

# 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

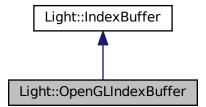
## **Additional Inherited Members**

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

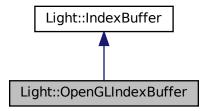
- Light/include/light/platform/opengl/openglframebuffer.hpp
- · Light/src/platform/opengl/openglframebuffer.cpp

## 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

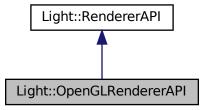
## **Additional Inherited Members**

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

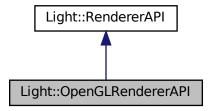
- Light/include/light/platform/opengl/openglbuffer.hpp
- Light/src/platform/opengl/openglbuffer.cpp

# 5.46 Light::OpenGLRendererAPI Class Reference

Inheritance diagram for Light::OpenGLRendererAPI:



Collaboration diagram for Light::OpenGLRendererAPI:

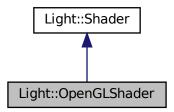


## **Additional Inherited Members**

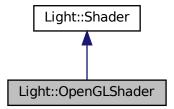
- · Light/include/light/platform/opengl/openglrendererapi.hpp
- Light/src/platform/opengl/openglrendererapi.cpp

## 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

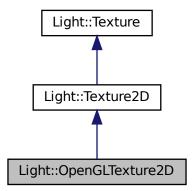
## **Additional Inherited Members**

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

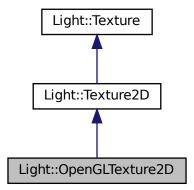
- Light/include/light/platform/opengl/openglshader.hpp
- · Light/src/platform/opengl/openglshader.cpp

# 5.48 Light::OpenGLTexture2D Class Reference

Inheritance diagram for Light::OpenGLTexture2D:



 $Collaboration\ diagram\ for\ Light:: OpenGLT exture 2D:$ 



## **Public Member Functions**

- OpenGLTexture2D (const std::string &path)
- uint32 t getRendererld () const override
- uint32\_t getWidth () const override
- uint32\_t getHeight () const override
- void bind (uint32\_t slot=0) const override

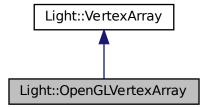
### **Additional Inherited Members**

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

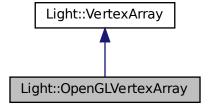
- Light/include/light/platform/opengl/opengltexture.hpp
- · Light/src/platform/opengl/opengltexture.cpp

# 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

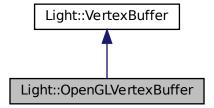
#### **Additional Inherited Members**

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

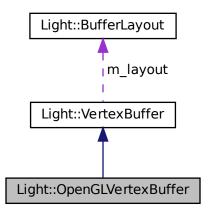
- Light/include/light/platform/opengl/openglvertexarray.hpp
- · Light/src/platform/opengl/openglvertexarray.cpp

# 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 Physics::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()

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()

Add a new RigidBody to m\_objects.

#### **Parameters**

```
object input, const RigidBody& type
```

#### 5.51.3.2 stepSimulation()

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)

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

## 5.53 Light::Renderer Class Reference

#### **Static Public Member Functions**

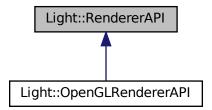
- · static void init ()
- static void **onWindowResize** (uint32\_t width, uint32\_t height)
- static void beginScene (Camera &camera, glm::mat4 camera view)
- static void endScene ()
- static void submitLight (glm::vec3 lightPos, glm::vec3 lightCol)
- static void submit (const std::shared\_ptr< Shader > &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)

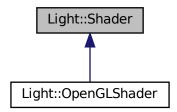
### **Friends**

- · class Entity
- · class ScenePanel

- 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)

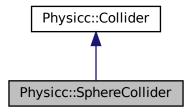
- · Light/include/light/rendering/shader.hpp
- Light/src/rendering/shader.cpp

# 5.59 Physicc::SphereCollider Class Reference

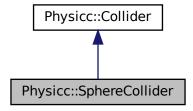
SphereCollider class.

#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()

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

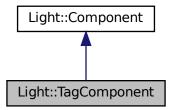
BoundingVolume::AABB

Implements Physicc::Collider.

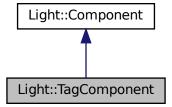
- Physicc/include/collider.hpp
- Physicc/src/collider.cpp

# 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**

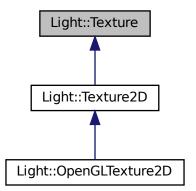
• std::string tag

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**

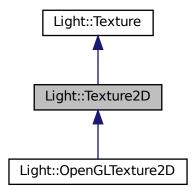
- virtual uint32\_t getWidth () const =0
- virtual uint32\_t getHeight () const =0
- virtual void **bind** (uint32\_t slot=0) const =0

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

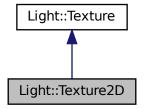
• Light/include/light/rendering/texture.hpp

# 5.62 Light::Texture2D Class Reference

Inheritance diagram for Light::Texture2D:



Collaboration diagram for Light::Texture2D:



### **Public Member Functions**

• virtual uint32\_t getRendererld () 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**

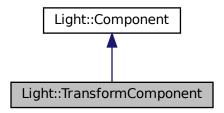
- Timestep (float time)
- float getSeconds ()
- float getMilliSeconds ()

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

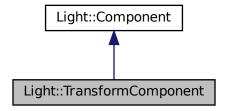
• LightFramework/include/core/timestep.hpp

# 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**

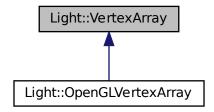
- glm::vec3 position
- glm::vec3 rotation
- glm::vec3 scale

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**

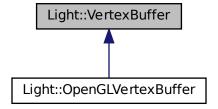
• static VertexArray \* create ()

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

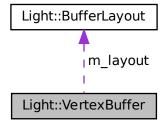
- · Light/include/light/rendering/vertexarray.hpp
- · Light/src/platform/opengl/openglvertexarray.cpp

# 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**

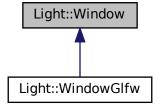
• BufferLayout m\_layout

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:



68 Class Documentation

#### **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**

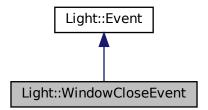
static Window \* create (const WindowProps &props=WindowProps())

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

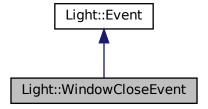
- · LightFramework/include/core/window.hpp
- LightFramework/src/platform/glfw/windowglfw.cpp

## 5.68 Light::WindowCloseEvent Class Reference

Inheritance diagram for Light::WindowCloseEvent:



Collaboration diagram for Light::WindowCloseEvent:



#### **Public Member Functions**

• std::string ToString () const override

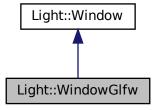
#### **Additional Inherited Members**

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

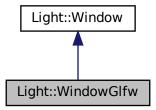
• LightFramework/include/events/applicationevent.hpp

# 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

70 Class Documentation

#### **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**

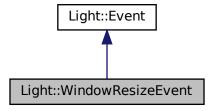
- std::string title
- · uint32\_t width
- uint32\_t height

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

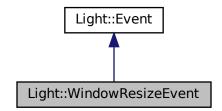
• LightFramework/include/core/window.hpp

# 5.71 Light::WindowResizeEvent Class Reference

 $Inheritance\ diagram\ for\ Light:: Window Resize Event:$ 



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

72 Class Documentation

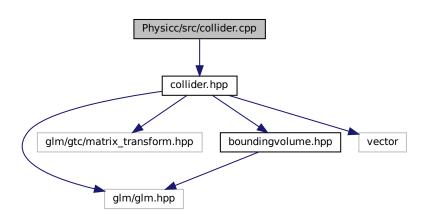
# **Chapter 6**

# **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)

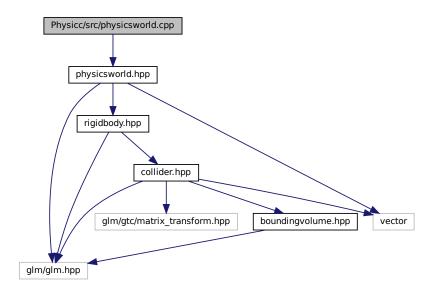
Bug No known bugs.

74 File Documentation

## 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)

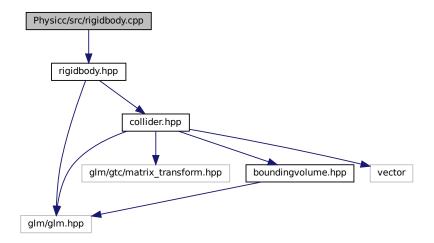
Tirthankar Mazumder (wermos)

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.

76 File Documentation

# **Chapter 7**

# **Example Documentation**

## 7.1 Use

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

# Index

addRigidBody	Light::LightComponent, 38
Physics::PhysicsWorld, 56	Light::MeshComponent, 39
	Light::MeshRendererComponent, 40
BaseBV	Light::MouseButtonEvent, 41
Physicc::BVImpl::BaseBV< Derived, BoundingOb-	Light::MouseButtonPressedEvent, 42
ject >, 10, 11	Light::MouseButtonReleasedEvent, 43
BoxCollider	Light::MouseMovedEvent, 44
Physicc::BoxCollider, 14	Light::MouseScrolledEvent, 45
	Light::OpenGLContext, 46
Collider	Light::OpenGLCubemap, 47
Physicc::Collider, 19	Light::OpenGLFramebuffer, 48
	Light::OpenGLIndexBuffer, 49
getAABB	Light::OpenGLRendererAPI, 50
Physico::BoxCollider, 14	Light::OpenGLShader, 51
Physicc::SphereCollider, 61	Light::OpenGLTexture2D, 52
getPosition	Light::OpenGLVertexArray, 53
Physicc::Collider, 20	Light::OpenGLVertexBuffer, 54
getRotate	Light::RenderCommand, 56
Physicc::Collider, 20	Light::Renderer, 57
getScale	Light::RendererAPI, 57
Physicc::Collider, 20	Light::Scene, 58
getTransform	Light::Shader, 59
Physicc::Collider, 20	Light::ShaderLibrary, 59
1.1.4.4.1.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.	Light::TagComponent, 62
Light::Application, 9	Light::Texture, 63
Light::BufferElement, 15	Light::Texture2D, 63
Light::BufferLayout, 15	Light::Timestep, 64
Light::Camera, 17	Light::TransformComponent, 65
Light::CameraComponent, 17	Light::VertexArray, 66
Light::Component, 22	Light::VertexBuffer, 66
Light::Cubemap, 23	Light::Window, 67
Light::EditorCamera, 23	Light::WindowCloseEvent, 68
Light::Entity, 24	Light::WindowGlfw, 69
Light::Event, 25	Light::WindowProps, 70
Light::EventDispatcher, 26	Light::WindowFlops, 70  Light::WindowResizeEvent, 70
Light::Framebuffer, 26	Light Window tool2020 event, 70
Light::FramebufferAttachmentsSpec, 27	overlapsWith
Light::FramebufferSpec, 27	Physicc::BVImpl::BaseBV< Derived, BoundingOb
Light::FramebufferTextureSpec, 28	ject >, 11
Light::GraphicsContext, 28	ject >, TT
Light::ImguiLayer, 29	Dhysica/aya/adliday.aya. 70
Light::IndexBuffer, 30	Physics/src/collider.cpp, 73
Light::Input, 30	Physics/src/physicsworld.cpp, 74
Light::InputGlfw, 31	Physicc/src/rigidbody.cpp, 74
Light::KeyEvent, 32	Physicc::BoxCollider, 13
Light::KeyPressedEvent, 33	BoxCollider, 14
Light::KeyReleasedEvent, 34	getAABB, 14
Light::KeyTypedEvent, 35	Physics::BVH, 16
Light::Layer, 36	Physics::BVHNode, 16
Light::LaverStack, 37	Physicc::BVImpl::AABB, 9

80 INDEX

```
Physicc::BVImpl::BaseBV< Derived, BoundingObject
          >, 10
     BaseBV, 10, 11
    overlapsWith, 11
Physicc::BVImpl::BoxBV< T >, 12
Physicc::Collider, 18
    Collider, 19
    getPosition, 20
    getRotate, 20
    getScale, 20
    getTransform, 20
    setPosition, 21
    setRotate, 21
    setScale, 21
    updateTransform, 22
Physicc::PhysicsWorld, 55
     addRigidBody, 56
     PhysicsWorld, 55
     stepSimulation, 56
Physicc::RigidBody, 58
Physicc::SphereCollider, 60
    getAABB, 61
     SphereCollider, 61
PhysicsWorld
     Physicc::PhysicsWorld, 55
setPosition
     Physicc::Collider, 21
setRotate
     Physicc::Collider, 21
setScale
     Physicc::Collider, 21
SphereCollider
     Physicc::SphereCollider, 61
stepSimulation
     Physicc::PhysicsWorld, 56
updateTransform
     Physicc::Collider, 22
```