

# PhysgameEngine

## .01

Generated by Doxygen 1.6.1

Sun Mar 21 22:34:08 2010



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	ActorBase Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.2	ActorDynRigid Class Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.3	ActorDynSoft Class Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.4	ActorSta Class Reference . . . . .	8
3.4.1	Detailed Description . . . . .	8
3.5	MetaCode Class Reference . . . . .	9
3.5.1	Detailed Description . . . . .	9
3.6	PhysEvent Class Reference . . . . .	10
3.6.1	Detailed Description . . . . .	10
3.7	PhysEventManager Class Reference . . . . .	11
3.7.1	Detailed Description . . . . .	11
3.8	PhysEventRenderTime Class Reference . . . . .	12
3.8.1	Detailed Description . . . . .	12
3.9	PhysEventUserInput Class Reference . . . . .	13
3.9.1	Detailed Description . . . . .	13
3.10	PhysQuaternion Class Reference . . . . .	14
3.10.1	Detailed Description . . . . .	14
3.11	PhysVector3 Class Reference . . . . .	15
3.11.1	Detailed Description . . . . .	15

3.12 PhysWorld Class Reference . . . . .	16
3.12.1 Detailed Description . . . . .	16
3.13 physworld Class Reference . . . . .	17
3.13.1 Detailed Description . . . . .	17
3.14 PhysWorldCallBackManager Class Reference . . . . .	18
3.14.1 Detailed Description . . . . .	18
3.15 Settings Class Reference . . . . .	19
3.15.1 Detailed Description . . . . .	19

# Chapter 1

## Class Index

### 1.1 Class Hierarchy

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

ActorBase . . . . .	5
ActorDynRigid . . . . .	6
ActorDynSoft . . . . .	7
ActorSta . . . . .	8
MetaCode . . . . .	9
PhysEvent . . . . .	10
PhysEventRenderTime . . . . .	12
PhysEventUserInput . . . . .	13
PhysEventManager . . . . .	11
PhysQuaternion . . . . .	14
PhysVector3 . . . . .	15
PhysWorld . . . . .	16
physworld . . . . .	17
PhysWorldCallBackManager . . . . .	18
Settings . . . . .	19



# Chapter 2

## Class Index

### 2.1 Class List

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

<a href="#">ActorBase</a>	5
<a href="#">ActorDynRigid</a>	6
<a href="#">ActorDynSoft</a>	7
<a href="#">ActorSta</a>	8
<a href="#">MetaCode</a>	9
<a href="#">PhysEvent</a>	10
<a href="#">PhysEventManager</a>	11
<a href="#">PhysEventRenderTime</a>	12
<a href="#">PhysEventUserInput</a>	13
<a href="#">PhysQuaternion</a>	14
<a href="#">PhysVector3</a>	15
<a href="#">PhysWorld</a>	16
<a href="#">physworld</a> (This is the main entry point for the entire library. The <a href="#">physworld</a> coordinates and integrates all the underlying subsystems, Currently Ogre3d is used for 3d Graphics, Bullet is used for physics, and SDL is used for user input and window management )	17
<a href="#">PhysWorldCallBackManager</a>	18
<a href="#">Settings</a>	19



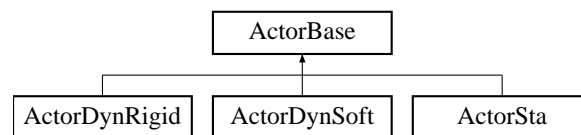


## Chapter 3

# Class Documentation

### 3.1 ActorBase Class Reference

Inheritance diagram for ActorBase::



#### Public Member Functions

- void **CreateEntity** (PhysString name, PhysString file, PhysString group)
- void **CreateSceneNode** ()
- void **SetOgreLocation** (PhysReal x, PhysReal y, PhysReal z)
- void **SetOgreOrientation** (PhysReal x, PhysReal y, PhysReal z, PhysReal w)

#### Protected Attributes

- [PhysQuaternion](#) **orientation**
- Ogre::Entity \* **entity**
- Ogre::SceneManager \* **physscenemanager**
- Ogre::SceneNode \* **node**
- btQuaternion \* **physorientation**
- btTransform \* **physlocation**

#### 3.1.1 Detailed Description

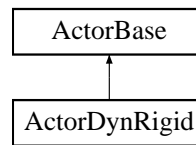
Definition at line 25 of file physactor.h.

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

- physactor.h
- physactor.cpp

## 3.2 ActorDynRigid Class Reference

Inheritance diagram for ActorDynRigid::



### Public Member Functions

- void **CreateRigidObject** ()
- void **AddObjectToWorld** ()

### Protected Attributes

- btRigidBody \* **physrigidbody**
- btMotionState \* **physmotionstate**

### 3.2.1 Detailed Description

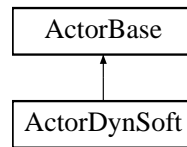
Definition at line 47 of file physactor.h.

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

- physactor.h
- physactor.cpp

## 3.3 ActorDynSoft Class Reference

Inheritance diagram for ActorDynSoft::



### Public Member Functions

- void **CreateSoftObject** ()
- void **AddObjectToWorld** ()

### Protected Attributes

- btSoftBody \* **physsoftbody**
- btMotionState \* **physmotionstate**

#### 3.3.1 Detailed Description

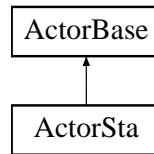
Definition at line 61 of file physactor.h.

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

- physactor.h
- physactor.cpp

## 3.4 ActorSta Class Reference

Inheritance diagram for ActorSta::



### Public Member Functions

- void **CreateRigidObject** ()
- void **AddObjectToWorld** ()

### Protected Attributes

- btRigidBody \* **physrigidbody**

#### 3.4.1 Detailed Description

Definition at line 75 of file physactor.h.

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

- physactor.h
- physactor.cpp

## 3.5 MetaCode Class Reference

### Public Member Functions

- **MetaCode** (int MetaValue\_, short unsigned int ID\_, InputCode Code\_)
- **MetaCode** (RawEvent \_RawEvent)
- InputCode **GetCode** ()
- void **SetCode** (InputCode Code\_)
- int **GetMetaValue** ()
- void **SetMetaValue** (int MetaValue\_)
- short unsigned int **GetID** ()
- void **SetID** (short unsigned int ID\_)
- bool **operator==** (const [MetaCode](#) &other) const

### 3.5.1 Detailed Description

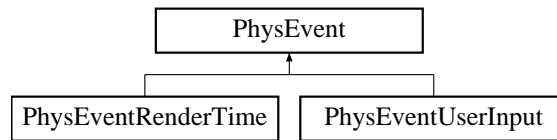
Definition at line 330 of file physeventuserinput.h.

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

- physeventuserinput.h
- physeventuserinput.cpp

## 3.6 PhysEvent Class Reference

Inheritance diagram for PhysEvent::



### Public Member Functions

- virtual EventType **getEventType** ()=0

#### 3.6.1 Detailed Description

Definition at line 21 of file physevent.h.

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

- physevent.h

## 3.7 PhysEventManager Class Reference

### Public Member Functions

- unsigned int **GetRemainingEventCount** ()
- [PhysEvent](#) \* **GetNextEvent** ()
- [PhysEventRenderTime](#) \* **GetNextRenderTimeEvent** ()
- [PhysEventUserInput](#) \* **GetNextUserInputEvent** ()
- void **AddEvent** ([PhysEvent](#) \*EventToAdd)
- bool **DoQuitMessagesExist** ()

### Static Public Member Functions

- static bool **IgnoreQuitEvents** ()
- static void **SetIgnoreQuitEvents** (bool Ignore)

#### 3.7.1 Detailed Description

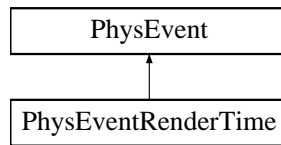
Definition at line 16 of file `physeventmanager.h`.

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

- `physeventmanager.h`
- `physeventmanager.cpp`

## 3.8 PhysEventRenderTime Class Reference

Inheritance diagram for PhysEventRenderTime::



### Public Member Functions

- **PhysEventRenderTime** (PhysWhole Milliseconds)
- virtual EventType **getEventType** ()
- PhysWhole **getMillisecondsSinceLastFrame** ()

#### 3.8.1 Detailed Description

Definition at line 13 of file physeventrendertime.h.

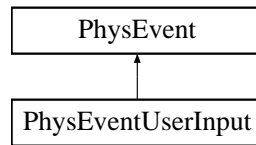
The documentation for this class was generated from the following files:

- physeventrendertime.h
- physeventrendertime.cpp



## 3.9 PhysEventUserInput Class Reference

Inheritance diagram for PhysEventUserInput::



### Public Member Functions

- **PhysEventUserInput** ([MetaCode](#) Code\_)
- **PhysEventUserInput** (vector< [MetaCode](#) > Code\_)
- [MetaCode](#) **GetCode** (unsigned int Index)
- unsigned int **GetCodeCount** ()
- void **AddCode** ([MetaCode](#) \_Code)
- void **ToggleCode** ([MetaCode](#) \_Code)
- void **ToggleCode** (unsigned int Index)
- virtual EventType **getEventType** ()

### 3.9.1 Detailed Description

Definition at line 353 of file physeventuserinput.h.

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

- physeventuserinput.h
- physeventuserinput.cpp

## 3.10 PhysQuaternion Class Reference

### Public Member Functions

- **PhysQuaternion** (PhysReal X, PhysReal Y, PhysReal Z, PhysReal W)

### Public Attributes

- PhysReal **X**
- PhysReal **Y**
- PhysReal **Z**
- PhysReal **W**

#### 3.10.1 Detailed Description

Definition at line 19 of file physvector.h.

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

- physvector.h
- physvector.cpp

## 3.11 PhysVector3 Class Reference

### Public Member Functions

- **PhysVector3** (PhysReal X, PhysReal Y, PhysReal Z)

### Public Attributes

- PhysReal **X**
- PhysReal **Y**
- PhysReal **Z**

#### 3.11.1 Detailed Description

Definition at line 6 of file physvector.h.

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

- physvector.h
- physvector.cpp

## 3.12 PhysWorld Class Reference

### Public Member Functions

- **PhysWorld** ([PhysVector3](#) \*GeographyLowerBounds, [PhysVector3](#) \*GeographyUpperbounds, unsigned short int MaxPhysicsProxies=1024)
- template<class T >  
void **Log** (T Message)
- template<class T >  
void **LogAndThrow** (T Message)
- bool **ShowSystemSettingDialog** ()
- void **MoveCamera** ([PhysVector3](#) Position, [PhysVector3](#) LookAt)
- void **GameInit** ()
- void **DoMainLoopAllItems** ()
- void **DoMainLoopPhysics** ()
- void **DoMainLoopInputBuffering** ()
- void **DoMainLoopWindowManagerBuffering** ()
- void **DoMainLoopRender** ()

### Public Attributes

- [PhysWorldCallBackManager](#) \* **Callbacks**
- [PhysEventManager](#) \* **Events**

### Friends

- void **RenderPhysWorld** ([PhysWorld](#) \*TheWorld)

#### 3.12.1 Detailed Description

Definition at line 63 of file physworld.h.

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

- physworld.h
- physworld.cpp

## 3.13 physworld Class Reference

This is the main entry point for the entire library. The [physworld](#) coordinates and integrates all the underlying subsystems, Currently Ogre3d is used for 3d Graphics, Bullet is used for physics, and SDL is used for user input and window management.

```
#include <physworld.h>
```

### 3.13.1 Detailed Description

This is the main entry point for the entire library. The [physworld](#) coordinates and integrates all the underlying subsystems, Currently Ogre3d is used for 3d Graphics, Bullet is used for physics, and SDL is used for user input and window management.

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

- [physworld.h](#)

## 3.14 PhysWorldCallbackManager Class Reference

### Public Member Functions

- **PhysWorldCallbackManager** ([PhysWorld](#) \*\_Parent)
- **bool PreInput** ()
- **void ErasePreInput** ()
- **void SetPreInput** (bool(\*Callback)())
- **bool IsPreInputCallbackSet** ()
- **bool PrePhysics** ()
- **void ErasePrePhysics** ()
- **void SetPrePhysics** (bool(\*Callback)())
- **bool IsPrePhysicsCallbackSet** ()
- **bool PreRender** ()
- **void ErasePreRender** ()
- **void SetPreRender** (bool(\*Callback)())
- **bool IsPreRenderCallbackSet** ()
- **bool PostRender** ()
- **void ErasePostRender** ()
- **void SetPostRender** (bool(\*Callback)())
- **bool IsPostRenderCallbackSet** ()

### Friends

- class [PhysWorld](#)

#### 3.14.1 Detailed Description

Definition at line 13 of file physworldcallbackmanager.h.

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

- physworldcallbackmanager.h
- physworldcallbackmanager.cpp

## 3.15 Settings Class Reference

### Public Member Functions

- bool **getFullscreen** ()
- bool **setFullscreen** (bool \_Fullscreen)
- int **getRenderHeight** ()
- int **getRenderWidth** ()
- bool **setRenderHeight** (int Height)
- bool **setRenderWidth** (int Width)
- bool **getFullscreen** ()
- bool **setFullscreen** (bool \_Fullscreen)
- int **getRenderHeight** ()
- int **getRenderWidth** ()
- bool **setRenderHeight** (int Height)
- bool **setRenderWidth** (int Width)

### 3.15.1 Detailed Description

Definition at line 13 of file gamebase.h.

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

- gamebase.h
- physgamesettings.h
- gamebase.cpp
- physgamesettings.cpp

# Index

ActorBase, [5](#)  
ActorDynRigid, [6](#)  
ActorDynSoft, [7](#)  
ActorSta, [8](#)

MetaCode, [9](#)

PhysEvent, [10](#)  
PhysEventManager, [11](#)  
PhysEventRenderTime, [12](#)  
PhysEventUserInput, [13](#)  
PhysQuaternion, [14](#)  
PhysVector3, [15](#)  
PhysWorld, [16](#)  
physworld, [17](#)  
PhysWorldCallBackManager, [18](#)

Settings, [19](#)