Physics Tools 1.0.1

Generated by Doxygen 1.9.5

1 Namespace Index	1
1.1 Package List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 Namespace Documentation	7
4.1 PhysicsTools Namespace Reference	7
4.2 PhysicsTools.Collisions Namespace Reference	7
4.3 PhysicsTools.Transformations Namespace Reference	7
4.4 PhysicsTools.Utility Namespace Reference	7
5 Class Documentation	9
5.1 PhysicsTools.Collisions.CollisionIgnorer Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 CollisionIgnorer()	10
5.1.3 Member Function Documentation	10
5.1.3.1 IgnoreCollider()	10
5.1.3.2 IgnoreColliderFor()	11
5.1.3.3 IgnoreColliders()	11
5.1.3.4 IgnoreRigidbody()	11
5.1.3.5 IgnoreRigidbodyFor()	12
5.1.3.6 UnignoreCollider()	12
5.1.3.7 UnignoreColliderIn()	12
5.1.3.8 UnignoreColliders()	12
5.1.3.9 UnignoreEntry()	14
5.1.3.10 UnignoreIndex()	14
5.1.3.11 UnignoreRigidbody()	14
5.1.3.12 UnignoreRigidbodyIn()	15
5.2 PhysicsTools.Collisions.CollisionTracker Class Reference	15
5.2.1 Detailed Description	15
5.2.2 Member Function Documentation	16
5.2.2.1 GetCollidingWith()	16
5.2.2.2 IsCollidingWith() [1/2]	16
5.2.2.3 IsCollidingWith() [2/2]	16
5.3 PhysicsTools.Collisions.CollisionIgnorer.Entry Class Reference	17
5.3.1 Detailed Description	17
5.3. Poetalled Description	17
5.4 Physics Tools. Transformations. TransformVelocity Tracker. Rotation Velocity TimePair Struct Reference	17
	19
5.5.1 Detailed Description	19

Index	23
5.6 PhysicsTools.Transformations.TransformVelocityTracker.VelocityTimePair Struct Reference	21
5.5.2.6 SetSampleVelocityEnabled()	. 21
5.5.2.5 SetSampleRotationVelocityEnabled()	. 20
5.5.2.4 SetRotateSampleTime()	. 20
5.5.2.3 SetMoveSampleTime()	. 20
5.5.2.2 GetAverageVelocityOverSeconds()	. 20
5.5.2.1 GetAverageRotationVelocityOverSeconds()	. 19
5.5.2 Member Function Documentation	. 19

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

PhysicsTools
PhysicsTools.Collisions
PhysicsTools.Transformations
PhysicsTools.Utility

2 Namespace Index

Hierarchical Index

2.1 Class Hierarchy

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

PhysicsTools.Collisions.CollisionIgnorer
PhysicsTools.Collisions.CollisionIgnorer.Entry
MonoBehaviour
PhysicsTools.Collisions.CollisionTracker
PhysicsTools.Transformations.TransformVelocityTracker
PhysicsTools.Transformations.TransformVelocityTracker.RotationVelocityTimePair
Physics Tools. Transform Velocity Tracker. Velocity Time Pair

4 Hierarchical Index

Class Index

3.1 Class List

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

PhysicsTools.Collisions.CollisionIgnorer	
A simple class that allows provides the neccesary methods and helpers to temporarily ignore	
collisions between a set of Colliders and a Collider.	9
PhysicsTools.Collisions.CollisionTracker	
A simple component that keeps track of collisions using OnCollisionEnter and OnCollisionExit.	15
PhysicsTools.Collisions.CollisionIgnorer.Entry	
An entry for a temporarily ignore collision rule	17
PhysicsTools.Transformations.TransformVelocityTracker.RotationVelocityTimePair	17
PhysicsTools.Transformations.TransformVelocityTracker	
A simple component that tracks velocity and angular velocity for a Transform that it is attached to.	18
PhysicsTools.Transformations.TransformVelocityTracker.VelocityTimePair	21

6 Class Index

Namespace Documentation

4.1 PhysicsTools Namespace Reference

4.2 PhysicsTools.Collisions Namespace Reference

Classes

· class CollisionIgnorer

A simple class that allows provides the neccesary methods and helpers to temporarily ignore collisions between a set of Colliders and a Collider.

· class CollisionTracker

A simple component that keeps track of collisions using OnCollisionEnter and OnCollisionExit.

4.3 PhysicsTools.Transformations Namespace Reference

Classes

· class TransformVelocityTracker

A simple component that tracks velocity and angular velocity for a Transform that it is attached to.

4.4 PhysicsTools.Utility Namespace Reference

Classes

· class RigidbodyUtility

A public static class that provides useful Rigidbody utilities.

Class Documentation

5.1 PhysicsTools.Collisions.CollisionIgnorer Class Reference

A simple class that allows provides the neccesary methods and helpers to temporarily ignore collisions between a set of Colliders and a Collider.

Classes

· class Entry

An entry for a temporarily ignore collision rule.

Public Member Functions

CollisionIgnorer (List< Collider > pColliders)

Creates a CollisionIgnorer that provides easy methods for temporarily ignoring collisions between a collider and $p \leftarrow$ Colliders.

• void Update ()

Updates the collision ignorer allowing it to timeout any temporarily ignored collisions.

void IgnoreCollider (Collider pCollider)

Ignore collisions between all colliders associated with this component and pCollider indefinitely.

void IgnoreColliders (IEnumerable < Collider > pColliders)

Ignore collisions between all colliders associated with this component and all colliders in the pColliders enumerable indefinitely.

• void IgnoreColliderFor (Collider pCollider, float pSeconds)

Ignores collisions between all colliders associated with this component and pCollider for pSeconds.

• void UnignoreColliderIn (Collider pCollider, float pSeconds)

Unignores collisions between all colliders associated with this component and pCollider in pSeconds.

void UnignoreCollider (Collider pCollider)

Unignores collisions between all colliders associated with this component and pCollider.

void UnignoreColliders (IEnumerable < Collider > pColliders)

Unignores collisions between all colliders associated with this component and all pColliders in the enumerable.

void UnignoreEntry (Entry pEntry)

Undoes the collision ignore entry.

void UnignoreIndex (int pIndex)

Undoes the ignore collision registered in entry index plndex.

void UnignoreAll ()

Undo all temporarily ignored collisions.

void IgnoreRigidbody (Rigidbody pRigidbody)

Ignores collisions between all colliders associated with this component and the Rigidbody, pRigidbody, indefinitely.

void IgnoreRigidbodyFor (Rigidbody pRigidbody, float pSeconds)

Ignores collisions between all colliders associated with this component and the Rigidbody, pRigidbody, for pSeconds.

• void UnignoreRigidbodyIn (Rigidbody pRigidbody, float pSeconds)

Unignores collisions between all colliders associated with this component and the Rigidbody, pRigidbody, in p← Seconds.

void UnignoreRigidbody (Rigidbody pRigidbody)

Unggnores collisions between all colliders associated with this component and the Rigidbody, pRigidbody.

5.1.1 Detailed Description

A simple class that allows provides the neccesary methods and helpers to temporarily ignore collisions between a set of Colliders and a Collider.

Author: Mathew Aloisio

5.1.2 Constructor & Destructor Documentation

5.1.2.1 CollisionIgnorer()

```
\label{limit} Physics Tools. Collisions. Collision Ignorer. Collision Ignorer \ ( \\ List < Collider > pColliders \ )
```

Creates a CollisionIgnorer that provides easy methods for temporarily ignoring collisions between a collider and pColliders.

Parameters

pColliders A List of colliders whose collision with other colliders is being cnotrolled by this class instance.

5.1.3 Member Function Documentation

5.1.3.1 IgnoreCollider()

```
void PhysicsTools.Collisions.CollisionIgnorer.IgnoreCollider (  {\tt Collider}\ pCollider\ )
```

Ignore collisions between all colliders associated with this component and pCollider indefinitely.

Parameters

pCollider

5.1.3.2 IgnoreColliderFor()

Ignores collisions between all colliders associated with this component and pCollider for pSeconds.

Parameters

pCollider pSeconds

5.1.3.3 IgnoreColliders()

```
void PhysicsTools.Collisions.CollisionIgnorer.IgnoreColliders ( {\tt IEnumerable} < {\tt Collider} > pColliders \; )
```

Ignore collisions between all colliders associated with this component and all colliders in the pColliders enumerable indefinitely.

Parameters

pColliders

5.1.3.4 IgnoreRigidbody()

```
void PhysicsTools.Collisions.CollisionIgnorer.IgnoreRigidbody ( {\tt Rigidbody}~pRigidbody~)
```

Ignores collisions between all colliders associated with this component and the Rigidbody, pRigidbody, indefinitely.

Parameters

pRigidbody

5.1.3.5 IgnoreRigidbodyFor()

```
void PhysicsTools.Collisions.CollisionIgnorer.IgnoreRigidbodyFor ( {\it Rigidbody,} \\ {\it float pSeconds} \ )
```

Ignores collisions between all colliders associated with this component and the Rigidbody, pRigidbody, for $p \leftarrow$ Seconds.

Parameters

pRigidbody	
pSeconds	The number of seconds to ignore collisions for.

5.1.3.6 UnignoreCollider()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreCollider ( {\tt Collider}\ pCollider\ )
```

Unignores collisions between all colliders associated with this component and pCollider.

Parameters

```
pCollider
```

5.1.3.7 UnignoreColliderIn()

Unignores collisions between all colliders associated with this component and pCollider in pSeconds.

Parameters

pCollider	
pSeconds	

5.1.3.8 UnignoreColliders()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreColliders ( {\tt IEnumerable} < {\tt Collider} > pColliders \ )
```

Unignores collisions between all colliders associated with this component and all pColliders in the enumerable.

Do					
Pа	ra	m	eı	re.	rs

pColliders

5.1.3.9 UnignoreEntry()

Undoes the collision ignore entry.

Parameters

pEntry

5.1.3.10 UnignoreIndex()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreIndex ( int \ p \textit{Index} \ )
```

Undoes the ignore collision registered in entry index plndex.

Parameters

pIndex

5.1.3.11 UnignoreRigidbody()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreRigidbody ( {\tt Rigidbody~pRigidbody~)}
```

Unggnores collisions between all colliders associated with this component and the Rigidbody, pRigidbody.

Parameters

pRigidbody

5.1.3.12 UnignoreRigidbodyIn()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreRigidbodyIn ( \label{eq:Rigidbody} Rigidbody, \\ \mbox{float $pSeconds$ )}
```

Unignores collisions between all colliders associated with this component and the Rigidbody, pRigidbody, in $p \leftarrow$ Seconds.

Parameters

pRigidbody	
pSeconds	The number of seconds to unignore collisions in.

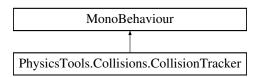
The documentation for this class was generated from the following file:

· CollisionIgnorer.cs

5.2 PhysicsTools.Collisions.CollisionTracker Class Reference

A simple component that keeps track of collisions using OnCollisionEnter and OnCollisionExit.

Inheritance diagram for PhysicsTools.Collisions.CollisionTracker:



Public Member Functions

• bool IsCollidingWith (Collider pCollider)

Returns true when the tracker is tracking a collision with the specified collider, pCollider, otherwise false.

• bool IsCollidingWith (Rigidbody pRigidbody, bool pAllowNestedRigidbodies)

Returns true if the tracker is colliding with any colliders of pRigidbody, otherwise false.

Collider GetCollidingWith (int pIndex)

Returns the Collider in the given 'colliding with' index.

Properties

int CollidingWithCount [get]

Returns the number of Colliders currently colliding with the relevant Rigidbody.

5.2.1 Detailed Description

A simple component that keeps track of collisions using OnCollisionEnter and OnCollisionExit.

Author: Mathew Aloisio

5.2.2 Member Function Documentation

5.2.2.1 GetCollidingWith()

```
Collider PhysicsTools.CollisionS.CollisionTracker.GetCollidingWith ( int \ pIndex \ )
```

Returns the Collider in the given 'colliding with' index.

Parameters

pIndex

Returns

The collider in the given 'colliding with' index.

5.2.2.2 IsCollidingWith() [1/2]

```
bool PhysicsTools.Collisions.CollisionTracker.IsCollidingWith ( {\tt Collider}\ pCollider\ )
```

Returns true when the tracker is tracking a collision with the specified collider, pCollider, otherwise false.

Parameters

pCollider

Returns

5.2.2.3 IsCollidingWith() [2/2]

```
bool PhysicsTools.Collisions.CollisionTracker.IsCollidingWith (  \label{eq:Rigidbody} \textit{Rigidbody,} \\  \mbox{bool } \textit{pAllowNestedRigidbodies} \ )
```

Returns true if the tracker is colliding with any colliders of pRigidbody, otherwise false.

Parameters

pRigidbody	
pAllowNestedRigidbodies	Should nested Rigidbody colliders be included?

Returns

true if the tracker is colliding with any colliders of pRigidbody, otherwise false.

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

· CollisionTracker.cs

5.3 PhysicsTools.Collisions.CollisionIgnorer.Entry Class Reference

An entry for a temporarily ignore collision rule.

Public Attributes

· Collider collider

The Collider associated with the entry.

float expireTime

The time the entry is set to expire.

5.3.1 Detailed Description

An entry for a temporarily ignore collision rule.

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

· CollisionIgnorer.cs

5.4 PhysicsTools.Transformations.TransformVelocityTracker.Rotation VelocityTimePair Struct Reference

Public Attributes

- · Quaternion rotationVelocity
- · float time

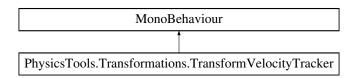
The documentation for this struct was generated from the following file:

TransformVelocityTracker.cs

5.5 PhysicsTools.Transformations.TransformVelocityTracker Class Reference

A simple component that tracks velocity and angular velocity for a Transform that it is attached to.

Inheritance diagram for PhysicsTools.Transformations.TransformVelocityTracker:



Classes

- · struct RotationVelocityTimePair
- struct VelocityTimePair

Public Member Functions

· void ClearData ()

Clears both velocity and angular velocity data.

• void ClearVelocityData ()

Clears all velocity data.

void ClearAngularVelocityData ()

Clears all angular velocity data.

Vector3 GetAverageVelocityOverSeconds (float pSeconds)

Computes the average velocity for the TransformVelocityTracker over the last pSeconds seconds. Adding more seconds past the length of data history the component tracks has no effect.

Quaternion GetAverageRotationVelocityOverSeconds (float pSeconds)

Computes the aerage rotation velocity using tracked data from this component over the last pSeconds seconds. If the component does not have that many seconds of data then increasing pSeconds will have no effect.

void SetSampleVelocityEnabled (bool pEnabled)

Sets the 'sampleVelocity' field of this component. Useful for use with Unity editor events.

void SetSampleRotationVelocityEnabled (bool pEnabled)

Sets the 'sampleRotationVelocity' field of this component. Useful for use with Unity editor events.

void SetMoveSampleTime (float pTime)

Sets the 'moveSampleTime' field of this component. Useful for use with Unity editor events.

void SetRotateSampleTime (float pTime)

Sets the 'rotateSampleTime' field of this component. Useful for use with Unity editor events.

Public Attributes

• bool sampleVelocity = true

Is velocity sampling enabled for this component?

• float moveSampleTime = 0.4f

All velocity samples up to velocitySampleTime seconds old will be used in average velocity calculations.

• bool sampleRotationVelocity = true

Is rotation velocity sampling enabled for this component?

• float rotateSampleTime = 0.1f

All angular/rotational velocity samples up to angularVelocitySampleTime seconds old will be used in average angular/rotational velocity calculations.

Properties

Vector3 PositionLastFrame [get]

The position of the Grabber's transform last frame.

Quaternion RotationLastFrame [get]

The rotation of the Grabber's transform last frame.

Vector3 AverageVelocity [get]

Returns the average velocity of the Transform in units per second over the last moveSampleTime seconds.

• int AverageVelocityEntries [get]

Returns the number of velocity entries used in the last 'Average Velocity' calculation.

• Vector3 VelocityLastFrame [get]

The velocity in units per second last frame.

Quaternion AverageRotationVelocity [get]

Returns the average rotation velocity of the Transform in units per second over the last rotateSampleTime seconds.

int AverageRotationVelocityEntries [get]

Returns the number of rotation velocity entries used in the last 'AverageRotationVelocity' calculation.

Quaternion RotationVelocityLastFrame [get]

The rotational velocity last frame represented as a Quaternion.

5.5.1 Detailed Description

A simple component that tracks velocity and angular velocity for a Transform that it is attached to.

Author: Mathew Aloisio

5.5.2 Member Function Documentation

5.5.2.1 GetAverageRotationVelocityOverSeconds()

```
Quaternion PhysicsTools.Transformations.TransformVelocityTracker.GetAverageRotationVelocity \leftarrow OverSeconds ( float pSeconds )
```

Computes the aerage rotation velocity using tracked data from this component over the last pSeconds seconds. If the component does not have that many seconds of data then increasing pSeconds will have no effect.

Parameters

pSeconds

Returns

the aerage rotation velocity using tracked data from this component over the last pSeconds seconds.

5.5.2.2 GetAverageVelocityOverSeconds()

 $\label{thm:pseconds} Vector 3 \ Physics Tools. Transform Velocity Tracker. Get Average Velocity Over Seconds \ (float \textit{pSeconds} \)$

Computes the average velocity for the TransformVelocityTracker over the last pSeconds seconds. Adding more seconds past the length of data history the component tracks has no effect.

Parameters

pSeconds

Returns

the average velocity for the TransformVelocityTracker over the last pSeconds seconds.

5.5.2.3 SetMoveSampleTime()

```
void PhysicsTools.Transformations.TransformVelocityTracker.SetMoveSampleTime ( \label{eq:ptime} float \ pTime \ )
```

Sets the 'moveSampleTime' field of this component. Useful for use with Unity editor events.

Parameters

pTime

5.5.2.4 SetRotateSampleTime()

```
\label{thm:condition} \mbox{Void PhysicsTools.TransformVelocityTracker.SetRotateSampleTime (} \\ \mbox{float } p\mbox{\it Time })
```

Sets the 'rotateSampleTime' field of this component. Useful for use with Unity editor events.

Parameters

pTime

5.5.2.5 SetSampleRotationVelocityEnabled()

 $\label{thm:point} \mbox{void PhysicsTools.TransformVelocityTracker.SetSampleRotationVelocityEnabled (} \\ \mbox{bool } p\mbox{\it Enabled} \mbox{\)}$

Sets the 'sampleRotationVelocity' field of this component. Useful for use with Unity editor events.

Parameters

pEnabled

5.5.2.6 SetSampleVelocityEnabled()

```
\label{thm:point} \mbox{void PhysicsTools.TransformVelocityTracker.SetSampleVelocityEnabled (} \\ \mbox{bool } p\mbox{\it Enabled} \mbox{ )}
```

Sets the 'sampleVelocity' field of this component. Useful for use with Unity editor events.

Parameters

pEnabled

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

· TransformVelocityTracker.cs

5.6 PhysicsTools.Transformations.TransformVelocityTracker.Velocity TimePair Struct Reference

Public Attributes

- · Vector3 velocity
- · float time

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

• TransformVelocityTracker.cs

Index

CollisionIgnorer	SetMoveSampleTime, 20
PhysicsTools.Collisions.CollisionIgnorer, 10	SetRotateSampleTime, 20
	SetSampleRotationVelocityEnabled, 20
GetAverageRotationVelocityOverSeconds	SetSampleVelocityEnabled, 21
PhysicsTools.Transformations.TransformVelocityTrac	$^{ ext{k}}$ physics Tools. Transformations. Transform Velocity Tracker. Rotation Velocity Transform Velocity Tracker.
19	17
GetAverageVelocityOverSeconds	PhysicsTools.Transformations.TransformVelocityTracker.VelocityTimePair,
PhysicsTools.Transformations.TransformVelocityTrac	ker, 21
19	PhysicsTools.Utility, 7
GetCollidingWith	•
PhysicsTools.Collisions.CollisionTracker, 16	SetMoveSampleTime
	PhysicsTools.Transformations.TransformVelocityTracker,
lgnoreCollider	20
PhysicsTools.Collisions.CollisionIgnorer, 10	SetRotateSampleTime
lgnoreColliderFor	Physics Tools. Transformations. Transform Velocity Tracker,
PhysicsTools.Collisions.CollisionIgnorer, 11	20
IgnoreColliders	SetSampleRotationVelocityEnabled
PhysicsTools.Collisions.CollisionIgnorer, 11	PhysicsTools.Transformations.TransformVelocityTracker,
lgnoreRigidbody	20
PhysicsTools.Collisions.CollisionIgnorer, 11	SetSampleVelocityEnabled
lgnoreRigidbodyFor	PhysicsTools.Transformations.TransformVelocityTracker,
PhysicsTools.Collisions.CollisionIgnorer, 11	21
sCollidingWith	
PhysicsTools.Collisions.CollisionTracker, 16	UnignoreCollider
	PhysicsTools.Collisions.CollisionIgnorer, 12
PhysicsTools, 7	UnignoreColliderIn
PhysicsTools.Collisions, 7	PhysicsTools.Collisions.CollisionIgnorer, 12
PhysicsTools.Collisions.CollisionIgnorer, 9	UnignoreColliders
CollisionIgnorer, 10	PhysicsTools.Collisions.CollisionIgnorer, 12
IgnoreCollider, 10	UnignoreEntry
IgnoreColliderFor, 11	PhysicsTools.Collisions.CollisionIgnorer, 14
IgnoreColliders, 11	UnignoreIndex
IgnoreRigidbody, 11	PhysicsTools.Collisions.CollisionIgnorer, 14
IgnoreRigidbodyFor, 11	UnignoreRigidbody
UnignoreCollider, 12	PhysicsTools.Collisions.CollisionIgnorer, 14
UnignoreColliderIn, 12	UnignoreRigidbodyIn
UnignoreColliders, 12	PhysicsTools.Collisions.CollisionIgnorer, 14
UnignoreEntry, 14	
UnignoreIndex, 14	
UnignoreRigidbody, 14	
UnignoreRigidbodyIn, 14	
PhysicsTools.Collisions.CollisionIgnorer.Entry, 17	
PhysicsTools.Collisions.CollisionTracker, 15	
GetCollidingWith, 16	
IsCollidingWith, 16	
PhysicsTools.Transformations, 7	
PhysicsTools.Transformations.TransformVelocityTracker,	
18	
GetAverageRotationVelocityOverSeconds, 19	
GetAverageVelocityOverSeconds, 19	