Physics Tools

1.0.1

Generated by Doxygen 1.9.5

Chapter 1

Namespace Index

1.1 Package List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

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.

Chapter 5

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↩ 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 pIndex.

- 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()

```
PhysicsTools.Collisions.CollisionIgnorer.CollisionIgnorer (
            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 (
            Collider pCollider )
```

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

Parameters

pCollider	

5.1.3.2 IgnoreColliderFor()

```
void PhysicsTools.Collisions.CollisionIgnorer.IgnoreColliderFor (
            Collider pCollider,
            float pSeconds )
```

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 (
            IEnumerable< 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 (
            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 (
            Rigidbody pRigidbody,
            float pSeconds )
```

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

Parameters

pRigidbody	
pSeconds	The number of seconds to ignore collisions for.

5.1.3.6 UnignoreCollider()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreCollider (
            Collider pCollider )
```

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

Parameters

pCollider	

5.1.3.7 UnignoreColliderIn()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreColliderIn (
            Collider pCollider,
            float pSeconds )
```

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 (
            IEnumerable< Collider > pColliders )
```

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

Parameters

pColliders	

5.1.3.9 UnignoreEntry()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreEntry (
            Entry pEntry )
```

Undoes the collision ignore entry.

Parameters

pEntry	

5.1.3.10 UnignoreIndex()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreIndex (
            int pIndex )
```

Undoes the ignore collision registered in entry index pIndex.

Parameters

pIndex	

5.1.3.11 UnignoreRigidbody()

```
void PhysicsTools.Collisions.CollisionIgnorer.UnignoreRigidbody (
            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 (
            Rigidbody pRigidbody,
            float pSeconds )
```

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

Parameters

pRigidbody	
pSeconds	The number of seconds to unignore collisions in.

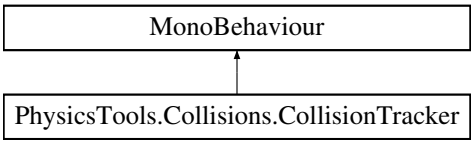
The documentation for this class was generated from the following file:

- 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 (
            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 (
            Rigidbody pRigidbody,
            bool 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.RotationVelocityTimePair Struct Reference

Public Attributes

- Quaternion **rotationVelocity**
- float **time**

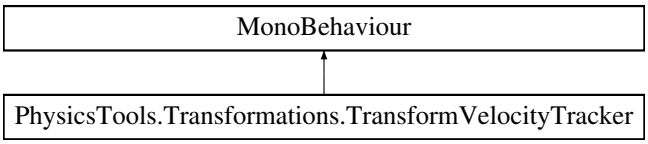
The documentation for this struct was generated from the following file:

- 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 'AverageVelocity' 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↩
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()

```
Vector3 PhysicsTools.Transformations.TransformVelocityTracker.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.

Parameters

pSeconds	

Returns

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

5.5.2.3 SetMoveSampleTime()

```
void PhysicsTools.Transformations.TransformVelocityTracker.SetMoveSampleTime (
            float pTime )
```

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

Parameters

pTime	

5.5.2.4 SetRotateSampleTime()

```
void PhysicsTools.Transformations.TransformVelocityTracker.SetRotateSampleTime (
            float pTime )
```

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

Parameters

pTime	

5.5.2.5 SetSampleRotationVelocityEnabled()

```
void PhysicsTools.Transformations.TransformVelocityTracker.SetSampleRotationVelocityEnabled (
            bool pEnabled )
```

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

Parameters

pEnabled	

5.5.2.6 SetSampleVelocityEnabled()

```
void PhysicsTools.Transformations.TransformVelocityTracker.SetSampleVelocityEnabled (
            bool pEnabled )
```

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.VelocityTimePair Struct Reference

Public Attributes

- Vector3 **velocity**
- float **time**

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

- TransformVelocityTracker.cs

Index