Laser System for Unity

Generated by Doxygen 1.9.3

Chapter 1

Namespace Index

1.1 Packages

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

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Laser Namespace Reference

Namespaces

- namespace Cache
- namespace debug
- namespace Drawing
- namespace drawing
- namespace dto
- namespace Examples
- namespace exceptions
- namespace Logic
- namespace ResourceLoading
- namespace Scripting

5.2 Laser.Cache Namespace Reference

Classes

- class LaserDrawerCache

  Provides reuse utility for laser assets so that there is no new object created for every laser interaction. Laser assets are identified by the LaserEmitter they are connected with.
- class LaserReceiverCache

  Helper class for the LaserEmitter to avoid GetComponent invocations every frame.

5.3 Laser.debug Namespace Reference

Classes

- class DebugLaserDrawer

  Draws debug rays to show where the lasers are cast.
- class LaserReceiverDebugger

  Simple debugger for the laser receiver.

5.4 Laser.Drawing Namespace Reference

Namespaces

- namespace Coloring
- namespace Controller

5.5 Laser.drawing Namespace Reference

Namespaces

- namespace helper

Classes

- class EmitterLaserDrawer

  Draws beams and activation particle effects for a LaserEmitter.
- class ForwarderLaserDrawer

  Base class for all laser forwarders (and Laser Actor that can shoot a laser - either its own like a LaserEmitter or someone else's like a LaserRelay).
- class MultiTargetLaserDrawer

  Handles the drawing of Laser Actors that repeat multiple incoming beams, each with its own particle systems, like LaserMirror
- class NonEmittingLaserDrawer

  Handles the drawings for the BlockingLaserReceiver.
- class SingleTargetLaserDrawer

  Handles the drawings of Laser Actors that only use one activation particle system when hit, like LaserRepeater and NonBlockingLaserReceiver.

5.6 Laser.Drawing.Coloring Namespace Reference

Classes

- class IgnoreColoring

  Particle systems with this component will be ignored by the LaserParticleController when finding the particle materials to color.
- class LaserColorRegistry

  Contains info on the selected color of a given EmitterLaserDrawer and its matching LaserEmitter.
- class ParticleColorBlender

  Blends between two particle colors and returns the mixed value as a combined particle color.

5.7 Laser.Drawing.Controller Namespace Reference

Classes

- class LaserBeamController

  Encapsulates the logic needed to interact with the laser beam's shader.
- class LaserParticleController

  Draws particles for the laser system. Includes an optional light that is switched on or off (if exists) when the particle effects are active.

5.8 Laser.drawing.helper Namespace Reference

Classes

- class LaserDtoTransformer

  Utility class to transform between laser dto-s.

5.9 Laser.dto Namespace Reference

Classes

- struct LaserDirection

  Contains information on a laser hit from
- struct LaserHit

  Contains information on a laser hit from the emitter to the target it seeks.
- struct LaserLineEndpoints

  Contains information for the laser beam to draw.
- struct LaserResult

  Contains information that is returned back to the emitter upon contact with a receiver.

5.10 Laser.Examples Namespace Reference

Classes

- class EmitterActivationSwitcher

5.11 Laser.exceptions Namespace Reference

Classes

- class MissingLaserAssetException

5.12 Laser.Logic Namespace Reference

Namespaces

- namespace Query

Classes

- class BlockingLaserReceiver

  A laser target that does not forward the incoming ray (is not pass through).
- interface ILaserEmitter

  A modifiable contract that expresses a laser emitter.
- interface ILaserReceiver

  Contains methods for the emitters to call when they hit this receiver. Interface exposed for custom receiver implementations.
- interface ILaserTarget

  Denotes that this receiver is an end target for an emitter. As an ILaserReceiver, contains methods to be called by the emitter to modify this target. Exposed for custom target implementations.
- class LaserActorRoot

  Marks a complex hierarchy's root that contains a Laser Actor. Useful when we want to move e.g. the whole emitter cube (whose children contain the actual emitter script).
- class LaserEmitter

  Emits laser forward that interacts with ILaserReceivers and ILaserTargets.
- class LaserMirror

  Forwards the incoming laser by reflecting it.
- class LaserRelay

  Base class for any laser object that will forward the incoming laser.
- class LaserRepeater

  Forwards the incoming laser in the GameObject's forward direction.
- class NonBlockingLaserReceiver

  Target that forwards the incoming laser, creating an illusion that it is pass-through.

5.13 Laser.Logic.Query Namespace Reference

Classes

- interface IQueryableLaserActor

  Contains the most basic information about Laser Actors and serves as a base.
- interface IQueryableLaserEmitter

  Provides information about a LaserEmitter.
- interface IQueryableLaserForwarder

  Provides information about a Laser Actor that can shoot a laser (either its own like a LaserEmitter or someone else's like a LaserRelay)
- interface IQueryableLaserReceiver

  Provides information about a Laser Actor that can be affected by the system's lasers.
- interface IQueryableLaserRelay

  Provides a type to query for information about relays.
- interface IQueryableLaserTarget

  Differentiates between regular receivers and targets (constructs that are a final target for an emitter).

5.14 Laser.ResourceLoading Namespace Reference

Classes

- class LaserComponentLoader

  Utility class to get a component from an object by either

5.15 Laser.Scripting Namespace Reference

Classes

- class LaserActorAware

  Utility to speed up the work with Laser Actors. Contains various queries implemented about laser actors in the scene. Ideally you want to rely on the events found in interfaces and run these queries on event invocation.

- class ScriptingExample

  Example to show that by extending the LaserActorAware class all the queries and events inside it are made available. To see the full list of available queries, see LaserActorAware.

Chapter 6

Class Documentation

6.1 Laser.Logic.BlockingLaserReceiver Class Reference

A laser target that does not forward the incoming ray (is not pass through).

Inherits MonoBehaviour, and Laser.Logic.ILaserTarget.

Public Member Functions

- bool Equals (IQueryableLaserActor other)
- Transform FindLaserRoot ()

  Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.
- override string ToString ()

  Overrides ToString method to help diagnose problems.

Properties

- ISet< LaserEmitter > AttachedEmitters `[get]`

  Returns a set of LaserEmitters that are attached to this target.
- int AttachedEmitterCount `[get]`

  The number of emitters attached to the target.

Events

- ILaserTarget.LaserEmitterAttached OnNewEmitterReceived
- ILaserTarget.LaserEmitterDetached OnEmitterDetached
- ILaserReceiver.NotifyHitByLaser HitByLaser
- ILaserReceiver.NotifyAllHitsCeased AllLaserHitsCeased

6.1.1 Detailed Description

A laser target that does not forward the incoming ray (is not pass through).

6.1.2 Member Function Documentation

6.1.2.1 Equals()

```
bool Laser.Logic.BlockingLaserReceiver.Equals (
            IQueryableLaserActor other )
```

6.1.2.2 FindLaserRoot()

```
Transform Laser.Logic.BlockingLaserReceiver.FindLaserRoot ( )
```

Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.

Relatively expensive to execute so make sure to cache the result at startup.

Returns

the root of the laser actor as marked by the developer or null if no such script is added to any of the parents.

Implements Laser.Logic.Query.IQueryableLaserActor.

6.1.2.3 ToString()

```
override string Laser.Logic.BlockingLaserReceiver.ToString ( )
```

Overrides ToString method to help diagnose problems.

Returns

Implements Laser.Logic.Query.IQueryableLaserActor.

6.1.3 Property Documentation

6.1.3.1 AttachedEmitterCount

```
int Laser.Logic.BlockingLaserReceiver.AttachedEmitterCount  [get]
```

The number of emitters attached to the target.

Implements Laser.Logic.Query.IQueryableLaserReceiver.

6.1.3.2 AttachedEmitters

```
ISet<LaserEmitter> Laser.Logic.BlockingLaserReceiver.AttachedEmitters  [get]
```

Returns a set of LaserEmitters that are attached to this target.

Note that this may not be accurate frame by frame due to emitters relying on Coroutines.

Implements Laser.Logic.Query.IQueryableLaserReceiver.

6.1.4 Event Documentation

6.1.4.1 AllLaserHitsCeased

```
ILaserReceiver.NotifyAllHitsCeased Laser.Logic.BlockingLaserReceiver.AllLaserHitsCeased
```

6.1.4.2 HitByLaser

```
ILaserReceiver.NotifyHitByLaser Laser.Logic.BlockingLaserReceiver.HitByLaser
```

6.1.4.3 OnEmitterDetached

```
ILaserTarget.LaserEmitterDetached Laser.Logic.BlockingLaserReceiver.OnEmitterDetached
```

6.1.4.4 OnNewEmitterReceived

```
ILaserTarget.LaserEmitterAttached Laser.Logic.BlockingLaserReceiver.OnNewEmitterReceived
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/BlockingLaserReceiver.cs

6.2 Laser.ResourceLoading.LaserComponentLoader< T >.Builder Class Reference

Builder class for the resource loader.

Public Member Functions

- Builder WithComponent (T asset)
- Builder WithPrefabPath (string prefabPath)
- Builder WithParent (Transform parent)
- LaserComponentLoader< T > Build ()

6.2.1 Detailed Description

Builder class for the resource loader.

6.2.2 Member Function Documentation

6.2.2.1 Build()

```
LaserComponentLoader< T > Laser.ResourceLoading.LaserComponentLoader< T >.Builder.Build ( )
```

6.2.2.2 WithComponent()

```
Builder Laser.ResourceLoading.LaserComponentLoader< T >.Builder.WithComponent (
            T asset )
```

6.2.2.3 WithParent()

```
Builder Laser.ResourceLoading.LaserComponentLoader< T >.Builder.WithParent (
            Transform parent )
```

6.2.2.4 WithPrefabPath()

```
Builder Laser.ResourceLoading.LaserComponentLoader< T >.Builder.WithPrefabPath (
            string prefabPath )
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/ResourceLoading/LaserComponentLoader.cs

6.3 Laser.debug.DebugLaserDrawer Class Reference

Draws debug rays to show where the lasers are cast.

Inherits MonoBehaviour.

Public Member Functions

- void DrawHittingLaser (Vector3 origin, Vector3 direction, float distance)

  Draws a laser that hits a collider.
- void DrawMissingLaser (Vector3 origin, Vector3 direction, float distance=1000f)

  Draws a laser that does not hit a collider.

6.3.1 Detailed Description

Draws debug rays to show where the lasers are cast.

6.3.2 Member Function Documentation

6.3.2.1 DrawHittingLaser()

```
void Laser.debug.DebugLaserDrawer.DrawHittingLaser (
            Vector3 origin,
            Vector3 direction,
            float distance )
```

Draws a laser that hits a collider.

Parameters

origin	the point where the laser starts
direction	the ray's direction
distance	the ray's distance

6.3.2.2 DrawMissingLaser()

```
void Laser.debug.DebugLaserDrawer.DrawMissingLaser (
            Vector3 origin,
            Vector3 direction,
            float distance = 1000f )
```

Draws a laser that does not hit a collider.

The ray is ended based on the given distance, or 1000 units, if no distance is given.

Parameters

origin	the point where the laser starts
direction	the ray's direction
distance	the ray's distance

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Debug/DebugLaserDrawer.cs

6.4 Laser.Examples.EmitterActivationSwitcher Class Reference

Inherits MonoBehaviour.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Examples/EmitterActivationSwitcher.cs

6.5 Laser.drawing.EmitterLaserDrawer Class Reference

Draws beams and activation particle effects for a LaserEmitter.

Inherits Laser.drawing.ForwarderLaserDrawer< LaserEmitter >.

Public Member Functions

- void UpdateLaserColor (Color color)

  Updates the beam color for this emitter drawer.
- void UpdateParticleColor (Color color)

  Updates the particle color for this emitter drawer.

Public Attributes

- string activationParticlesPath
- LaserParticleController activationParticlesTemplate
- Transform activationParticlesOrigin
- Color laserColor
- Color particleColor

Protected Member Functions

- override void Start ()
- override void SetUpSubscriptions ()
- override void TearDownSubscriptions ()
- override void OnLaserHitActor (IQueryableLaserForwarder sender, LaserHit laserHit, IQueryableLaserReceiver receiver)
- override void OnLaserMiss (IQueryableLaserForwarder sender, LaserDirection laserDirection)
- override void OnLaserHitNonActor (IQueryableLaserForwarder sender, LaserHit laserHit)

Additional Inherited Members

6.5.1 Detailed Description

Draws beams and activation particle effects for a LaserEmitter.

6.5.2 Member Function Documentation

6.5.2.1 OnLaserHitActor()

```
override void Laser.drawing.EmitterLaserDrawer.OnLaserHitActor (
            IQueryableLaserForwarder sender,
            LaserHit laserHit,
            IQueryableLaserReceiver receiver )  [protected]
```

6.5.2.2 OnLaserHitNonActor()

```
override void Laser.drawing.EmitterLaserDrawer.OnLaserHitNonActor (
            IQueryableLaserForwarder sender,
            LaserHit laserHit )  [protected]
```

6.5.2.3 OnLaserMiss()

```
override void Laser.drawing.EmitterLaserDrawer.OnLaserMiss (
            IQueryableLaserForwarder sender,
            LaserDirection laserDirection )  [protected]
```

6.5.2.4 SetUpSubscriptions()

```
override void Laser.drawing.EmitterLaserDrawer.SetUpSubscriptions ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< LaserEmitter >.

6.5.2.5 Start()

```
override void Laser.drawing.EmitterLaserDrawer.Start ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< LaserEmitter >.

6.5.2.6 TearDownSubscriptions()

```
override void Laser.drawing.EmitterLaserDrawer.TearDownSubscriptions ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< LaserEmitter >.

6.5.2.7 UpdateLaserColor()

```
void Laser.drawing.EmitterLaserDrawer.UpdateLaserColor (
            Color color )
```

Updates the beam color for this emitter drawer.

Parameters

color	new color for the beam

6.5.2.8 UpdateParticleColor()

```
void Laser.drawing.EmitterLaserDrawer.UpdateParticleColor (
            Color color )
```

Updates the particle color for this emitter drawer.

Parameters

color	new color for the particles

6.5.3 Member Data Documentation

6.5.3.1 activationParticlesOrigin

```
Transform Laser.drawing.EmitterLaserDrawer.activationParticlesOrigin
```

6.5.3.2 activationParticlesPath

```
string Laser.drawing.EmitterLaserDrawer.activationParticlesPath
```

6.5.3.3 activationParticlesTemplate

```
LaserParticleController Laser.drawing.EmitterLaserDrawer.activationParticlesTemplate
```

6.5.3.4 laserColor

```
Color Laser.drawing.EmitterLaserDrawer.laserColor
```

6.5.3.5 particleColor

```
Color Laser.drawing.EmitterLaserDrawer.particleColor
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/EmitterLaserDrawer.cs

6.6 Laser.drawing.ForwarderLaserDrawer< T > Class Template Reference

Base class for all laser forwarders (and Laser Actor that can shoot a laser - either its own like a LaserEmitter or someone else's like a LaserRelay).

Inherits MonoBehaviour.

Public Attributes

- string laserBeamAssetPath = "Prefabs/laser_beam"
- string endParticlesAssetPath = "Prefabs/hit_particles"
- LaserParticleController endParticlesTemplate
- LaserBeamController beamTemplate

Protected Member Functions

- virtual void Start ()
- virtual void SetUpSubscriptions ()
- virtual void TearDownSubscriptions ()
- virtual void OnLaserHitNonActor (IQueryableLaserForwarder sender, LaserHit outgoingHit)
- virtual void OnLaserHitActor (IQueryableLaserForwarder sender, LaserHit outgoingHit, IQueryableLaserReceiver receiver)
- virtual void OnLaserMiss (IQueryableLaserForwarder sender, LaserDirection outgoingDirection)
- virtual void OnLaserHitCeased (IQueryableLaserReceiver sender, LaserEmitter laserEmitter)
- virtual void OnHitByLaser (IQueryableLaserReceiver sender, LaserHit incomingHit)

Properties

- LaserDrawerCache< LaserBeamController > BeamCache [get]
- LaserColorRegistry ColorRegistry [get]
- T Forwarder [get]

6.6.1 Detailed Description

Base class for all laser forwarders (and Laser Actor that can shoot a laser - either its own like a LaserEmitter or someone else's like a LaserRelay).

Template Parameters

T	The information required by the laser construct for drawing

Type Constraints

T* : *IQueryableLaserForwarder

6.6.2 Member Function Documentation

6.6.2.1 OnHitByLaser()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.OnHitByLaser (
            IQueryableLaserReceiver sender,
            LaserHit incomingHit )  [protected], [virtual]
```

6.6.2.2 OnLaserHitActor()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.OnLaserHitActor (
            IQueryableLaserForwarder sender,
            LaserHit outgoingHit,
            IQueryableLaserReceiver receiver )  [protected], [virtual]
```

6.6.2.3 OnLaserHitCeased()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.OnLaserHitCeased (
            IQueryableLaserReceiver sender,
            LaserEmitter laserEmitter )  [protected], [virtual]
```

6.6.2.4 OnLaserHitNonActor()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.OnLaserHitNonActor (
            IQueryableLaserForwarder sender,
            LaserHit outgoingHit )  [protected], [virtual]
```

6.6.2.5 OnLaserMiss()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.OnLaserMiss (
            IQueryableLaserForwarder sender,
            LaserDirection outgoingDirection )  [protected], [virtual]
```

6.6.2.6 SetUpSubscriptions()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.SetUpSubscriptions ( )  [protected],
[virtual]
```

Reimplemented in Laser.drawing.EmitterLaserDrawer, Laser.drawing.MultiTargetLaserDrawer, and Laser.drawing.SingleTargetLaserD

6.6.2.7 Start()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.Start ( )  [protected], [virtual]
```

Reimplemented in Laser.drawing.EmitterLaserDrawer, Laser.drawing.MultiTargetLaserDrawer, and Laser.drawing.SingleTargetLaserD

6.6.2.8 TearDownSubscriptions()

```
virtual void Laser.drawing.ForwarderLaserDrawer< T >.TearDownSubscriptions ( )  [protected],
[virtual]
```

Reimplemented in Laser.drawing.EmitterLaserDrawer, Laser.drawing.MultiTargetLaserDrawer, and Laser.drawing.SingleTargetLaserD

6.6.3 Member Data Documentation

6.6.3.1 beamTemplate

LaserBeamController Laser.drawing.ForwarderLaserDrawer< T >.beamTemplate

6.6.3.2 endParticlesAssetPath

string Laser.drawing.ForwarderLaserDrawer< T >.endParticlesAssetPath = "Prefabs/hit_particles"

6.6.3.3 endParticlesTemplate

LaserParticleController Laser.drawing.ForwarderLaserDrawer< T >.endParticlesTemplate

6.6.3.4 laserBeamAssetPath

string Laser.drawing.ForwarderLaserDrawer< T >.laserBeamAssetPath = "Prefabs/laser_beam"

6.6.4 Property Documentation

6.6.4.1 BeamCache

LaserDrawerCache<LaserBeamController> Laser.drawing.ForwarderLaserDrawer< T >.BeamCache [get], [protected]

6.6.4.2 ColorRegistry

LaserColorRegistry Laser.drawing.ForwarderLaserDrawer< T >.ColorRegistry [get], [protected]

6.6.4.3 Forwarder

```
T Laser.drawing.ForwarderLaserDrawer< T >.Forwarder  [get], [protected]
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/ForwarderLaserDrawer.cs

6.7 Laser.Drawing.Coloring.IgnoreColoring Class Reference

Particle systems with this component will be ignored by the LaserParticleController when finding the particle materials to color.

Inherits MonoBehaviour.

6.7.1 Detailed Description

Particle systems with this component will be ignored by the LaserParticleController when finding the particle materials to color.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Coloring/IgnoreColoring.cs

6.8 Laser.Logic.ILaserEmitter Interface Reference

A modifiable contract that expresses a laser emitter.

Inherits Laser.Logic.Query.IQueryableLaserEmitter.

Inherited by Laser.Logic.LaserEmitter.

Public Member Functions

- void Activate ()

  Used to activate an inactive LaserEmitter. Starts the emission coroutine.
- void Deactivate ()

  Used to deactivate an active LaserEmitter. Stops the emission coroutine.

Additional Inherited Members

6.8.1 Detailed Description

A modifiable contract that expresses a laser emitter.

6.8.2 Member Function Documentation

6.8.2.1 Activate()

```
void Laser.Logic.ILaserEmitter.Activate ( )
```

Used to activate an inactive LaserEmitter. Starts the emission coroutine.

Implemented in Laser.Logic.LaserEmitter.

6.8.2.2 Deactivate()

```
void Laser.Logic.ILaserEmitter.Deactivate ( )
```

Used to deactivate an active LaserEmitter. Stops the emission coroutine.

Implemented in Laser.Logic.LaserEmitter.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/ILaserEmitter.cs

6.9 Laser.Logic.ILaserReceiver Interface Reference

Contains methods for the emitters to call when they hit this receiver. Interface exposed for custom receiver implementations.

Inherits Laser.Logic.Query.IQueryableLaserReceiver.

Inherited by Laser.Logic.ILaserTarget, and Laser.Logic.LaserRelay.

Public Member Functions

- LaserResult Hit (LaserHit laserHit)

  Called by a LaserEmitter or LaserRelay whenever this instance is hit by their laser.
- void CeaseHit (LaserEmitter sender)

  Called by the LaserEmitter when it stops affecting the given target.

Additional Inherited Members

6.9.1 Detailed Description

Contains methods for the emitters to call when they hit this receiver. Interface exposed for custom receiver implementations.

If you only want to query for information about this receiver, use IQueryableLaserReceiver instead.

6.9.2 Member Function Documentation

6.9.2.1 CeaseHit()

```
void Laser.Logic.ILaserReceiver.CeaseHit (
            LaserEmitter sender )
```

Called by the LaserEmitter when it stops affecting the given target.

Interface exposed to make custom implementations available. This method is to be called by a LaserEmitter only.

Parameters

sender	the laser emitter that no longer hits this target.

6.9.2.2 Hit()

```
LaserResult Laser.Logic.ILaserReceiver.Hit (
            LaserHit laserHit )
```

Called by a LaserEmitter or LaserRelay whenever this instance is hit by their laser.

Interface exposed to make custom implementations available. This method is to be called by a LaserEmitter only.

Parameters

laserHit	Information on the laser hit.

Returns

The result of the laser hit.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/ILaserReceiver.cs

6.10 Laser.Logic.ILaserTarget Interface Reference

Denotes that this receiver is an end target for an emitter. As an ILaserReceiver, contains methods to be called by the emitter to modify this target. Exposed for custom target implementations.

Inherits Laser.Logic.ILaserReceiver, and Laser.Logic.Query.IQueryableLaserTarget.

Inherited by Laser.Logic.BlockingLaserReceiver, and Laser.Logic.NonBlockingLaserReceiver.

Additional Inherited Members

6.10.1 Detailed Description

Denotes that this receiver is an end target for an emitter. As an ILaserReceiver, contains methods to be called by the emitter to modify this target. Exposed for custom target implementations.

If you only want to query information about a target, use IQueryableLaserTarget instead.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/ILaserTarget.cs

6.11 Laser.Logic.Query.IQueryableLaserActor Interface Reference

Contains the most basic information about Laser Actors and serves as a base.

Inherits IEquatable< IQueryableLaserActor >.

Inherited by Laser.Logic.Query.IQueryableLaserForwarder, and Laser.Logic.Query.IQueryableLaserReceiver.

Public Member Functions

- int GetInstanceID ()

  The Unity Instance ID.
- Transform FindLaserRoot ()

  Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.
- string ToString ()

  Overrides ToString method to help diagnose problems.

Properties

- string name `[get]`

  The name of the Laser Actor GameObject.
- string tag `[get]`

  The tag of the Laser Actor GameObject.

6.11.1 Detailed Description

Contains the most basic information about Laser Actors and serves as a base.

6.11.2 Member Function Documentation

6.11.2.1 FindLaserRoot()

```
Transform Laser.Logic.Query.IQueryableLaserActor.FindLaserRoot ( )
```

Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.

Relatively expensive to execute so make sure to cache the result at startup.

Returns

the root of the laser actor as marked by the developer or null if no such script is added to any of the parents.

Implemented in Laser.Logic.BlockingLaserReceiver, Laser.Logic.LaserEmitter, and Laser.Logic.LaserRelay.

6.11.2.2 GetInstanceID()

```
int Laser.Logic.Query.IQueryableLaserActor.GetInstanceID ( )
```

The Unity Instance ID.

Returns

The Unity instance ID

6.11.2.3 ToString()

```
string Laser.Logic.Query.IQueryableLaserActor.ToString ( )
```

Overrides ToString method to help diagnose problems.

Returns

Implemented in Laser.Logic.BlockingLaserReceiver, Laser.Logic.LaserEmitter, and Laser.Logic.LaserRelay.

6.11.3 Property Documentation

6.11.3.1 name

```
string Laser.Logic.Query.IQueryableLaserActor.name  [get]
```

The name of the Laser Actor GameObject.

Returns

The object's name

6.11.3.2 tag

```
string Laser.Logic.Query.IQueryableLaserActor.tag  [get]
```

The tag of the Laser Actor GameObject.

Returns

The object's tag

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/IQueryableLaserActor.cs

6.12 Laser.Logic.Query.IQueryableLaserEmitter Interface Reference

Provides information about a LaserEmitter.

Inherits Laser.Logic.Query.IQueryableLaserForwarder.

Inherited by Laser.Logic.ILaserEmitter.

Public Member Functions

- delegate void NotifyActivated (IQueryableLaserEmitter sender)

  Invoked when the LaserEmitter is activated.
- delegate void NotifyDeactivated (IQueryableLaserEmitter sender)

  Invoked when the LaserEmitter is deactivated.
- delegate void NotifyChainReturned (IQueryableLaserEmitter sender, LaserResult result)

  Invoked when the laser chain through the forwarders (if any) has returned. Contains information on the result of the hits and forwarded hits.

Properties

- ISet< IQueryableLaserReceiver > AffectedReceivers [get]

  The set of receivers (including relays) that are affected by this emitter.

Events

- NotifyActivated EmitterActivated
- NotifyDeactivated EmitterDeactivated
- NotifyChainReturned ChainReturned

6.12.1 Detailed Description

Provides information about a LaserEmitter.

6.12.2 Member Function Documentation

6.12.2.1 NotifyActivated()

```
delegate void Laser.Logic.Query.IQueryableLaserEmitter.NotifyActivated (
            IQueryableLaserEmitter sender )
```

Invoked when the LaserEmitter is activated.

6.12.2.2 NotifyChainReturned()

```
delegate void Laser.Logic.Query.IQueryableLaserEmitter.NotifyChainReturned (
            IQueryableLaserEmitter sender,
            LaserResult result )
```

Invoked when the laser chain through the forwarders (if any) has returned. Contains information on the result of the hits and forwarded hits.

6.12.2.3 NotifyDeactivated()

```
delegate void Laser.Logic.Query.IQueryableLaserEmitter.NotifyDeactivated (
            IQueryableLaserEmitter sender )
```

Invoked when the LaserEmitter is deactivated.

6.12.3 Property Documentation

6.12.3.1 AffectedReceivers

```
ISet<IQueryableLaserReceiver> Laser.Logic.Query.IQueryableLaserEmitter.AffectedReceivers
[get]
```

The set of receivers (including relays) that are affected by this emitter.

Creates a copy of the set when invoked.

Implemented in Laser.Logic.LaserEmitter.

6.12.4 Event Documentation

6.12.4.1 ChainReturned

```
NotifyChainReturned Laser.Logic.Query.IQueryableLaserEmitter.ChainReturned
```

6.12.4.2 EmitterActivated

```
NotifyActivated Laser.Logic.Query.IQueryableLaserEmitter.EmitterActivated
```

6.12.4.3 EmitterDeactivated

```
NotifyDeactivated Laser.Logic.Query.IQueryableLaserEmitter.EmitterDeactivated
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/IQueryableLaserEmitter.cs

6.13 Laser.Logic.Query.IQueryableLaserForwarder Interface Reference

Provides information about a Laser Actor that can shoot a laser (either its own like a LaserEmitter or someone else's like a LaserRelay)

Inherits Laser.Logic.Query.IQueryableLaserActor.

Inherited by Laser.Logic.Query.IQueryableLaserEmitter, and Laser.Logic.Query.IQueryableLaserRelay.

Public Member Functions

- delegate void NotifyLaserHitActor (IQueryableLaserForwarder sender, LaserHit outgoingHit, IQueryableLaserReceiver hitReceiver)

  Invoked when the laser emitted (or forwarded) by this Laser Actor hits another actor.
- delegate void NotifyLaserHitNonActor (IQueryableLaserForwarder sender, LaserHit outgoingHit)

  Invoked when the laser emitted (or forwarded) by this Laser Actor hits another collider that is not a Laser Actor.
- delegate void NotifyLaserMiss (IQueryableLaserForwarder sender, LaserDirection outgoingDirection)

  Invoked when the laser emitted (or forwarded) by this Laser Actor does not hit any collider.

Events

- NotifyLaserHitActor LaserHitActor
- NotifyLaserHitNonActor LaserHitNonActor
- NotifyLaserMiss LaserMiss

Additional Inherited Members

6.13.1 Detailed Description

Provides information about a Laser Actor that can shoot a laser (either its own like a LaserEmitter or someone else's like a LaserRelay)

6.13.2 Member Function Documentation

6.13.2.1 NotifyLaserHitActor()

```
delegate void Laser.Logic.Query.IQueryableLaserForwarder.NotifyLaserHitActor (
            IQueryableLaserForwarder sender,
            LaserHit outgoingHit,
            IQueryableLaserReceiver hitReceiver )
```

Invoked when the laser emitted (or forwarded) by this Laser Actor hits another actor.

Invoked every iteration continuously while the laser makes contact.

6.13.2.2 NotifyLaserHitNonActor()

```
delegate void Laser.Logic.Query.IQueryableLaserForwarder.NotifyLaserHitNonActor (
            IQueryableLaserForwarder sender,
            LaserHit outgoingHit )
```

Invoked when the laser emitted (or forwarded) by this Laser Actor hits another collider that is not a Laser Actor.

Invoked every iteration continuously while the laser makes contact.

6.13.2.3 NotifyLaserMiss()

```
delegate void Laser.Logic.Query.IQueryableLaserForwarder.NotifyLaserMiss (
            IQueryableLaserForwarder sender,
            LaserDirection outgoingDirection )
```

Invoked when the laser emitted (or forwarded) by this Laser Actor does not hit any collider.

Invoked every iteration continuously while the laser makes contact.

6.13.3 Event Documentation

6.13.3.1 LaserHitActor

```
NotifyLaserHitActor Laser.Logic.Query.IQueryableLaserForwarder.LaserHitActor
```

6.13.3.2 LaserHitNonActor

```
NotifyLaserHitNonActor Laser.Logic.Query.IQueryableLaserForwarder.LaserHitNonActor
```

6.13.3.3 LaserMiss

```
NotifyLaserMiss Laser.Logic.Query.IQueryableLaserForwarder.LaserMiss
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/IQueryableLaserForwarder.cs

6.14 Laser.Logic.Query.IQueryableLaserReceiver Interface Reference

Provides information about a Laser Actor that can be affected by the system's lasers.

Inherits Laser.Logic.Query.IQueryableLaserActor.

Inherited by Laser.Logic.ILaserReceiver, Laser.Logic.Query.IQueryableLaserRelay, and Laser.Logic.Query.IQueryableLaserTarget.

Public Member Functions

- delegate void LaserEmitterAttached (IQueryableLaserReceiver sender, LaserHit laserHit)

  Invoked when this receiver is hit by a new receiver it wasn't getting hit by before.
- delegate void LaserEmitterDetached (IQueryableLaserReceiver sender, LaserEmitter laserEmitter)

  Invoked when this Laser Actor used to receive a laser but does not anymore.
- delegate void NotifyHitByLaser (IQueryableLaserReceiver sender, LaserHit incomingHit)

  Invoked when this Laser Actor is hit by a LaserEmitter or its forwarded laser.
- delegate void NotifyAllHitsCeased (IQueryableLaserReceiver sender)

  Invoked when this Laser Actor is not hit by any LaserEmitter anymore.

Properties

- ISet< LaserEmitter > AttachedEmitters `[get]`

  Returns a set of LaserEmitters that are attached to this target.
- int AttachedEmitterCount `[get]`

  The number of emitters attached to the target.

Events

- LaserEmitterAttached OnNewEmitterReceived
- LaserEmitterDetached OnEmitterDetached
- NotifyHitByLaser HitByLaser
- NotifyAllHitsCeased AllLaserHitsCeased

6.14.1 Detailed Description

Provides information about a Laser Actor that can be affected by the system's lasers.

6.14.2 Member Function Documentation

6.14.2.1 LaserEmitterAttached()

```
delegate void Laser.Logic.Query.IQueryableLaserReceiver.LaserEmitterAttached (
            IQueryableLaserReceiver sender,
            LaserHit laserHit )
```

Invoked when this receiver is hit by a new receiver it wasn't getting hit by before.

Invoked once when the new emitter hits this receiver.

6.14.2.2 LaserEmitterDetached()

```
delegate void Laser.Logic.Query.IQueryableLaserReceiver.LaserEmitterDetached (
            IQueryableLaserReceiver sender,
            LaserEmitter laserEmitter )
```

Invoked when this Laser Actor used to receive a laser but does not anymore.

Invoked once when the laser stopped hitting this Laser Actor.

6.14.2.3 NotifyAllHitsCeased()

```
delegate void Laser.Logic.Query.IQueryableLaserReceiver.NotifyAllHitsCeased (
            IQueryableLaserReceiver sender )
```

Invoked when this Laser Actor is not hit by any LaserEmitter anymore.

Invoked once, after the hit ceased event if there is no emitter affecting this actor.

6.14.2.4 NotifyHitByLaser()

```
delegate void Laser.Logic.Query.IQueryableLaserReceiver.NotifyHitByLaser (
            IQueryableLaserReceiver sender,
            LaserHit incomingHit )
```

Invoked when this Laser Actor is hit by a LaserEmitter or its forwarded laser.

Invoked every iteration continuously while the other actor's laser made contact with this one.

6.14.3 Property Documentation

6.14.3.1 AttachedEmitterCount

```
int Laser.Logic.Query.IQueryableLaserReceiver.AttachedEmitterCount  [get]
```

The number of emitters attached to the target.

Implemented in Laser.Logic.BlockingLaserReceiver, and Laser.Logic.LaserRelay.

6.14.3.2 AttachedEmitters

```
ISet<LaserEmitter> Laser.Logic.Query.IQueryableLaserReceiver.AttachedEmitters  [get]
```

Returns a set of LaserEmitters that are attached to this target.

Note that this may not be accurate frame by frame due to emitters relying on Coroutines.

Implemented in Laser.Logic.BlockingLaserReceiver, and Laser.Logic.LaserRelay.

6.14.4 Event Documentation

6.14.4.1 AllLaserHitsCeased

`NotifyAllHitsCeased Laser.Logic.Query.IQueryableLaserReceiver.AllLaserHitsCeased`

6.14.4.2 HitByLaser

`NotifyHitByLaser Laser.Logic.Query.IQueryableLaserReceiver.HitByLaser`

6.14.4.3 OnEmitterDetached

`LaserEmitterDetached Laser.Logic.Query.IQueryableLaserReceiver.OnEmitterDetached`

6.14.4.4 OnNewEmitterReceived

`LaserEmitterAttached Laser.Logic.Query.IQueryableLaserReceiver.OnNewEmitterReceived`

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/IQueryableLaserReceiver.cs

6.15 Laser.Logic.Query.IQueryableLaserRelay Interface Reference

Provides a type to query for information about relays.

Inherits Laser.Logic.Query.IQueryableLaserForwarder, and Laser.Logic.Query.IQueryableLaserReceiver.

Inherited by Laser.Logic.LaserRelay.

Additional Inherited Members

6.15.1 Detailed Description

Provides a type to query for information about relays.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/IQueryableLaserRelay.cs

6.16 Laser.Logic.Query.IQueryableLaserTarget Interface Reference

Differentiates between regular receivers and targets (constructs that are a final target for an emitter).

Inherits Laser.Logic.Query.IQueryableLaserReceiver.

Inherited by Laser.Logic.ILaserTarget.

Additional Inherited Members

6.16.1 Detailed Description

Differentiates between regular receivers and targets (constructs that are a final target for an emitter).

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/IQueryableLaserTarget.cs

6.17 Laser.Scripting.LaserActorAware Class Reference

Utility to speed up the work with Laser Actors. Contains various queries implemented about laser actors in the scene. Ideally you want to rely on the events found in interfaces and run these queries on event invocation.

Inherits MonoBehaviour.

Inherited by Laser.Scripting.ScriptingExample.

Public Member Functions

- delegate void NotifyAllTargetsHit ()

  Invoked when all the laser targets are hit by at least one emitter.
- delegate void NotifyAnyTargetHit (IQueryableLaserTarget target)

  Invoked when a target that was unaffected before is hit by an emitter.
- delegate void NotifyAllReceiversHit ()

  Invoked when all the laser receivers are hit by at least one emitter.
- delegate void NotifyAnyReceiverHit (IQueryableLaserReceiver receiver)

  Invoked when a receiver that was unaffected before is hit by an emitter.
- T FindLaserActorByRootName< T > (string actorRootName)

  Gets the Laser Actor of given type by root name (name of the GameObject with the LaserActorRoot on it on the parent hierarchy of the LaserActor object).
- ISet< T > GetLaserActorsByType< T > ()

  Gets all Laser Actors of type T.
- int GetAffectedReceiverCount (IQueryableLaserEmitter emitter)

  Find the number of IQueryableLaserReceivers affected by the given emitter.
- int GetAffectedTargetCount (IQueryableLaserEmitter emitter)

  Find the number of IQueryableLaserTargets affected by the given receiver.
- ISet< IQueryableLaserReceiver > GetAffectedReceivers (IQueryableLaserEmitter emitter)

  Find the IQueryableLaserReceivers that are affected by the given emitter.

- ISet< IQueryableLaserTarget > GetAffectedTargets (IQueryableLaserEmitter emitter)

  Find the IQueryableLaserTargets affected by the given emitter.
- int GetNumberOfAffectingEmitters (IQueryableLaserReceiver receiver)

  Find the number of emitters affecting a given receiver - or target.
- int GetLaserActorCountOfType< T > ()

  Find the number of actors of a given type (e.g. number of laser emitters or number of laser mirrors)
- int GetTotalLaserActorCount ()

  Find the total number of laser actors in the scene.
- ISet< IQueryableLaserEmitter > GetAffectingLaserEmitters (IQueryableLaserReceiver receiver)

  Find the emitters that affect the given laser receiver (or target - targets are receivers too).
- bool IsReceiverAffectedByEmitter (IQueryableLaserReceiver receiver, IQueryableLaserEmitter emitter)

  Finds whether a given receiver (or target) is affected by a certain emitter.
- bool DoesEmitterAffectReceiver (IQueryableLaserEmitter emitter, IQueryableLaserReceiver receiver)

  Finds whether a given emitter affects a given receiver.
- bool AreAllTargetsAffectedByAnEmitter ()

  Find whether all targets are affected by an emitter.
- bool AreAllReceiversAffectedByAnEmitter ()

  Find whether all receivers are affected by an emitter.
- int GetNumberOfUnaffectedReceivers ()

  Find the number of receivers that are NOT affected by at least one emitter.
- int GetNumberOfAffectedReceivers ()

  Find the number of receivers that are affected by at least one emitter.
- int GetNumberOfAffectedTargets ()

  Find the number of targets that are affected by at least one emitter.
- int GetNumberOfUnaffectedTargets ()

  Find the number of targets (not all receivers) that are NOT affected by at least one emitter.
- ISet< IQueryableLaserEmitter > GetEmittersWithoutReceivers ()

  Find the emitters that do not affect any IQueryableLaserReceivers.
- int GetNumberOfEmittersWithoutReceivers ()

  Find the number of emitters that do not affect any IQueryableLaserReceivers.
- ISet< IQueryableLaserEmitter > GetEmittersWithoutTargets ()

  Find the emitters that do not affect any IQueryableLaserTargets.
- int GetNumberOfEmittersWithoutTargets ()

  Find the number of emitters that do not affect any IQueryableLaserTargets.

Protected Member Functions

- virtual void Awake ()
- virtual void OnDestroy ()

Events

- NotifyAllTargetsHit OnAllTargetsHit
- NotifyAnyTargetHit OnAnyTargetHit
- NotifyAllReceiversHit OnAllReceiversHit
- NotifyAnyReceiverHit OnAnyReceiverHit

6.17.1 Detailed Description

Utility to speed up the work with Laser Actors. Contains various queries implemented about laser actors in the scene. Ideally you want to rely on the events found in interfaces and run these queries on event invocation.

NOTE: only the Laser Actors that were present at the start of the scene are being taken into account. Actors instantiated during game are treated as non existent. NOTE: be careful with using GameObject.Find - many of the prefabs have a child object with a laser component on it and these child objects have the same name.

You can however name your root laser actor (object with the LaserActorRoot on it) and find a laser component by that name.

Explanations:

- affected by an emitter means that the laser emitter's laser hits the given object.

6.17.2 Member Function Documentation

6.17.2.1 AreAllReceiversAffectedByAnEmitter()

```
bool Laser.Scripting.LaserActorAware.AreAllReceiversAffectedByAnEmitter ( )
```

Find whether all receivers are affected by an emitter.

Returns

true if all the receivers - not just, but also targets - are affected by an emitter.

6.17.2.2 AreAllTargetsAffectedByAnEmitter()

```
bool Laser.Scripting.LaserActorAware.AreAllTargetsAffectedByAnEmitter ( )
```

Find whether all targets are affected by an emitter.

Returns

true if all targets (but not all receivers that are not targets) are affected by an emitter, false otherwise.

6.17.2.3 Awake()

```
virtual void Laser.Scripting.LaserActorAware.Awake ( )  [protected], [virtual]
```

6.17.2.4 DoesEmitterAffectReceiver()

```
bool Laser.Scripting.LaserActorAware.DoesEmitterAffectReceiver (
            IQueryableLaserEmitter emitter,
            IQueryableLaserReceiver receiver )
```

Finds whether a given emitter affects a given receiver.

Parameters

emitter	the emitter in question
receiver	the receiver in question

Returns

true if the emitter affects the receiver, false otherwise.

6.17.2.5 FindLaserActorByRootName< T >()

```
T Laser.Scripting.LaserActorAware.FindLaserActorByRootName< T > (
            string actorRootName )
```

Gets the Laser Actor of given type by root name (name of the GameObject with the LaserActorRoot on it on the parent hierarchy of the LaserActor object).

Finds the first root actor by this name and returns the IQueryableLaserActor using GetComponentInChildren. Use sparingly or in scenes with few laser actors. Otherwise, store the result calculated in an initializer method.

Parameters

actorRootName	GameObject name of the actor

Template Parameters

T	type of the actor

Returns

the actor or null if no actor of given type was found by that name

Type Constraints

T* : *class

T* : *IQueryableLaserActor

6.17.2.6 GetAffectedReceiverCount()

```
int Laser.Scripting.LaserActorAware.GetAffectedReceiverCount (
            IQueryableLaserEmitter emitter )
```

Find the number of IQueryableLaserReceivers affected by the given emitter.

Parameters

emitter	the laser emitter in question

Returns

the number of receivers the emitter currently affects.

6.17.2.7 GetAffectedReceivers()

```
ISet< IQueryableLaserReceiver > Laser.Scripting.LaserActorAware.GetAffectedReceivers (
            IQueryableLaserEmitter emitter )
```

Find the IQueryableLaserReceivers that are affected by the given emitter.

Parameters

emitter	the emitter in question

Returns

the receivers that are affected by the given emitter

6.17.2.8 GetAffectedTargetCount()

```
int Laser.Scripting.LaserActorAware.GetAffectedTargetCount (
            IQueryableLaserEmitter emitter )
```

Find the number of IQueryableLaserTargets affected by the given receiver.

Parameters

emitter	the emitter in question

Returns

the number of targets affected by the given emitter

6.17.2.9 GetAffectedTargets()

```
ISet< IQueryableLaserTarget > Laser.Scripting.LaserActorAware.GetAffectedTargets (
            IQueryableLaserEmitter emitter )
```

Find the IQueryableLaserTargets affected by the given emitter.

Parameters

emitter	the emitter in question

Returns

the targets affected by the given emitter.

6.17.2.10 GetAffectingLaserEmitters()

```
ISet< IQueryableLaserEmitter > Laser.Scripting.LaserActorAware.GetAffectingLaserEmitters (
            IQueryableLaserReceiver receiver )
```

Find the emitters that affect the given laser receiver (or target - targets are receivers too).

Parameters

receiver	the receiver whose emitters are in question

Returns

the emitters that affect the given receiver.

6.17.2.11 GetEmittersWithoutReceivers()

```
ISet< IQueryableLaserEmitter > Laser.Scripting.LaserActorAware.GetEmittersWithoutReceivers ( )
```

Find the emitters that do not affect any IQueryableLaserReceivers.

Returns

the emitters that do not affect any receiver.

6.17.2.12 GetEmittersWithoutTargets()

```
ISet< IQueryableLaserEmitter > Laser.Scripting.LaserActorAware.GetEmittersWithoutTargets ( )
```

Find the emitters that do not affect any IQueryableLaserTargets.

Returns

the emitters that do not affect any targets.

6.17.2.13 GetLaserActorCountOfType< T >()

```
int Laser.Scripting.LaserActorAware.GetLaserActorCountOfType< T > ( )
```

Find the number of actors of a given type (e.g. number of laser emitters or number of laser mirrors)

Template Parameters

T	The type of the given actor. Can be either an interface or a concrete class

Returns

the number of actors of the given type.

Type Constraints

T* : *IQueryableLaserActor

6.17.2.14 GetLaserActorsByType< T >()

```
ISet< T > Laser.Scripting.LaserActorAware.GetLaserActorsByType< T > ( )
```

Gets all Laser Actors of type T.

Template Parameters

T	the type of the actors to find

Returns

all the actors of the given type.

Type Constraints

T* : *class

T* : *IQueryableLaserActor

6.17.2.15 GetNumberOfAffectedReceivers()

```
int Laser.Scripting.LaserActorAware.GetNumberOfAffectedReceivers ( )
```

Find the number of receivers that are affected by at least one emitter.

Returns

the number of receivers (including targets) that are affected by at least one receiver.

6.17.2.16 GetNumberOfAffectedTargets()

```
int Laser.Scripting.LaserActorAware.GetNumberOfAffectedTargets ( )
```

Find the number of targets that are affected by at least one emitter.

Returns

the number of targets that are affected by at least one receiver.

6.17.2.17 GetNumberOfAffectingEmitters()

```
int Laser.Scripting.LaserActorAware.GetNumberOfAffectingEmitters (
            IQueryableLaserReceiver receiver )
```

Find the number of emitters affecting a given receiver - or target.

Parameters

receiver	the receiver in question

Returns

the number of emitters that affect the receiver.

6.17.2.18 GetNumberOfEmittersWithoutReceivers()

```
int Laser.Scripting.LaserActorAware.GetNumberOfEmittersWithoutReceivers ( )
```

Find the number of emitters that do not affect any IQueryableLaserReceivers.

Returns

the number of emitters that do not affect any receiver.

6.17.2.19 GetNumberOfEmittersWithoutTargets()

```
int Laser.Scripting.LaserActorAware.GetNumberOfEmittersWithoutTargets ( )
```

Find the number of emitters that do not affect any IQueryableLaserTargets.

Returns

the number of emitters that do not affect any targets.

6.17.2.20 GetNumberOfUnaffectedReceivers()

```
int Laser.Scripting.LaserActorAware.GetNumberOfUnaffectedReceivers ( )
```

Find the number of receivers that are NOT affected by at least one emitter.

Returns

the number of receivers (including targets) that are NOT affected by at least one receiver.

6.17.2.21 GetNumberOfUnaffectedTargets()

```
int Laser.Scripting.LaserActorAware.GetNumberOfUnaffectedTargets ( )
```

Find the number of targets (not all receivers) that are NOT affected by at least one emitter.

Returns

the number of targets that are NOT affected by at least one emitter.

6.17.2.22 GetTotalLaserActorCount()

```
int Laser.Scripting.LaserActorAware.GetTotalLaserActorCount ( )
```

Find the total number of laser actors in the scene.

Returns

the total number of laser actors in the scene.

6.17.2.23 IsReceiverAffectedByEmitter()

```
bool Laser.Scripting.LaserActorAware.IsReceiverAffectedByEmitter (
            IQueryableLaserReceiver receiver,
            IQueryableLaserEmitter emitter )
```

Finds whether a given receiver (or target) is affected by a certain emitter.

Parameters

receiver	the receiver in question
emitter	the emitter in question

Returns

true if the emitter affects the receiver of false otherwise.

6.17.2.24 NotifyAllReceiversHit()

```
delegate void Laser.Scripting.LaserActorAware.NotifyAllReceiversHit ( )
```

Invoked when all the laser receivers are hit by at least one emitter.

6.17.2.25 NotifyAllTargetsHit()

```
delegate void Laser.Scripting.LaserActorAware.NotifyAllTargetsHit ( )
```

Invoked when all the laser targets are hit by at least one emitter.

6.17.2.26 NotifyAnyReceiverHit()

```
delegate void Laser.Scripting.LaserActorAware.NotifyAnyReceiverHit (
            IQueryableLaserReceiver receiver )
```

Invoked when a receiver that was unaffected before is hit by an emitter.

6.17.2.27 NotifyAnyTargetHit()

```
delegate void Laser.Scripting.LaserActorAware.NotifyAnyTargetHit (
            IQueryableLaserTarget target )
```

Invoked when a target that was unaffected before is hit by an emitter.

6.17.2.28 OnDestroy()

```
virtual void Laser.Scripting.LaserActorAware.OnDestroy ( )  [protected], [virtual]
```

Reimplemented in Laser.Scripting.ScriptingExample.

6.17.3 Event Documentation

6.17.3.1 OnAllReceiversHit

`NotifyAllReceiversHit Laser.Scripting.LaserActorAware.OnAllReceiversHit`

6.17.3.2 OnAllTargetsHit

`NotifyAllTargetsHit Laser.Scripting.LaserActorAware.OnAllTargetsHit`

6.17.3.3 OnAnyReceiverHit

`NotifyAnyReceiverHit Laser.Scripting.LaserActorAware.OnAnyReceiverHit`

6.17.3.4 OnAnyTargetHit

`NotifyAnyTargetHit Laser.Scripting.LaserActorAware.OnAnyTargetHit`

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Scripting/LaserActorAware.cs

6.18 Laser.Logic.LaserActorRoot Class Reference

Marks a complex hierarchy's root that contains a Laser Actor. Useful when we want to move e.g. the whole emitter cube (whose children contain the actual emitter script).

Inherits MonoBehaviour.

6.18.1 Detailed Description

Marks a complex hierarchy's root that contains a Laser Actor. Useful when we want to move e.g. the whole emitter cube (whose children contain the actual emitter script).

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserActorRoot.cs

6.19 Laser.Drawing.Controller.LaserBeamController Class Reference

Encapsulates the logic needed to interact with the laser beam's shader.

Inherits MonoBehaviour.

Public Member Functions

- void EraseBeam ()

  Makes the laser beam disappear.
- void DrawLaserBeam (LaserLineEndpoints endpoints, int emitterId)

  Draws the laser beam between the given endpoints and designates it to the emitter it comes from.

6.19.1 Detailed Description

Encapsulates the logic needed to interact with the laser beam's shader.

6.19.2 Member Function Documentation

6.19.2.1 DrawLaserBeam()

```
void Laser.Drawing.Controller.LaserBeamController.DrawLaserBeam (
            LaserLineEndpoints endpoints,
            int emitterId )
```

Draws the laser beam between the given endpoints and designates it to the emitter it comes from.

Parameters

endpoints	the start and end points of the laser beam
emitterId	the emitter the beam belongs to

6.19.2.2 EraseBeam()

```
void Laser.Drawing.Controller.LaserBeamController.EraseBeam ( )
```

Makes the laser beam disappear.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Controller/LaserBeamController.cs

6.20 Laser.Drawing.Coloring.LaserColorRegistry Class Reference

Contains info on the selected color of a given EmitterLaserDrawer and its matching LaserEmitter.

Public Member Functions

- void RegisterBeamColor (LaserEmitter emitter, Color color)

  Register the beam color of the emitter.
- void RegisterParticleColor (LaserEmitter emitter, Color color)

  Register the particle color of the emitter.
- Color GetBeamColor (int emitterId)

  Returns the beam color of an emitter by its Instance ID.
- Color GetParticleColor (int emittedId)

  Returns the particle system color of the given emitter by its Instance ID.

Properties

- static LaserColorRegistry Instance `[get]`

  Returns the Singleton instance

6.20.1 Detailed Description

Contains info on the selected color of a given EmitterLaserDrawer and its matching LaserEmitter.

6.20.2 Member Function Documentation

6.20.2.1 GetBeamColor()

```
Color Laser.Drawing.Coloring.LaserColorRegistry.GetBeamColor (
            int emitterId )
```

Returns the beam color of an emitter by its Instance ID.

Parameters

emitterId	Instance ID of the emitter

Returns

the beam color of the emitter

6.20.2.2 GetParticleColor()

```
Color Laser.Drawing.Coloring.LaserColorRegistry.GetParticleColor (
            int emittedId )
```

Returns the particle system color of the given emitter by its Instance ID.

Parameters

emittedId	the Instance ID of the emitter

Returns

the particle color of the emitter

6.20.2.3 RegisterBeamColor()

```
void Laser.Drawing.Coloring.LaserColorRegistry.RegisterBeamColor (
            LaserEmitter emitter,
            Color color )
```

Register the beam color of the emitter.

Parameters

emitter	the emitter whose beam color will be registered
color	the color of the beam

6.20.2.4 RegisterParticleColor()

```
void Laser.Drawing.Coloring.LaserColorRegistry.RegisterParticleColor (
            LaserEmitter emitter,
            Color color )
```

Register the particle color of the emitter.

Parameters

emitter	the emitter whose particle color will be registered.
color	the color of the particles

6.20.3 Property Documentation

6.20.3.1 Instance

```
LaserColorRegistry Laser.Drawing.Coloring.LaserColorRegistry.Instance  [static], [get]
```

Returns the Singleton instance

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Coloring/LaserColorRegistry.cs

6.21 Laser.ResourceLoading.LaserComponentLoader< T > Class Template Reference

Utility class to get a component from an object by either

Classes

- class Builder

  Builder class for the resource loader.

Public Member Functions

- T GetComponent ()

  Tries to find a given GameObject with a component of given type on it and if found returns the component, if not, instantiates the component from resources and returns the given component from it.

6.21.1 Detailed Description

Utility class to get a component from an object by either

1. seeing if the component has been dragged in from the editor
2. trying to load the prefab from resources and get the component from there.

It also throws the appropriate exception if the component could not be located in any way.

Type Constraints

T* : *Component

6.21.2 Member Function Documentation

6.21.2.1 GetComponent()

```
T Laser.ResourceLoading.LaserComponentLoader< T >.GetComponent ( )
```

Tries to find a given GameObject with a component of given type on it and if found returns the component, if not, instantiates the component from resources and returns the given component from it.

If no component found even after the instantiation from resource loading it throws an exception. This means that the component was not set up in Editor nor was it present on a prefab that tried to be loaded. Either set up the asset at editor or check if you set the correct path for the prefab at resources.

Returns

the component from the asset - that is either found or instantiated

Exceptions

MissingLaserAssetException	when the component was not present on the object nor on the instantiated prefab.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/ResourceLoading/LaserComponentLoader.cs

6.22 Laser.dto.LaserDirection Struct Reference

Contains information on a laser hit from

Static Public Member Functions

- static LaserDirection Create (Vector3 origin, Vector3 direction, LaserEmitter emitter, float distance)

  Creates a new laser direction instance.

Properties

- Vector3 Origin `[get]`

  Origin of the raycast
- Vector3 Direction `[get]`

  Direction of the raycast
- LaserEmitter Emitter `[get]`

  The emitter the raycast belongs to
- float Distance `[get]`

  The max distance of the raycast, needed to have a limit and to draw an endpoint to the beam.

6.22.1 Detailed Description

Contains information on a laser hit from

6.22.2 Member Function Documentation

6.22.2.1 Create()

```
static LaserDirection Laser.dto.LaserDirection.Create (
            Vector3 origin,
            Vector3 direction,
            LaserEmitter emitter,
            float distance )  [static]
```

Creates a new laser direction instance.

Parameters

origin	origin
direction	direction
emitter	emitter
distance	distance

Returns

6.22.3 Property Documentation

6.22.3.1 Direction

```
Vector3 Laser.dto.LaserDirection.Direction  [get]
```

Direction of the raycast

6.22.3.2 Distance

```
float Laser.dto.LaserDirection.Distance  [get]
```

The max distance of the raycast, needed to have a limit and to draw an endpoint to the beam.

6.22.3.3 Emitter

```
LaserEmitter Laser.dto.LaserDirection.Emitter  [get]
```

The emitter the raycast belongs to

6.22.3.4 Origin

```
Vector3 Laser.dto.LaserDirection.Origin  [get]
```

Origin of the raycast

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserDirection.cs

6.23 Laser.Cache.LaserDrawerCache< T > Class Template Reference

Provides reuse utility for laser assets so that there is no new object created for every laser interaction. Laser assets are identified by the LaserEmitter they are connected with.

Public Member Functions

- LaserDrawerCache (T template, Transform parent)
- T LocateLaserAsset (LaserEmitter emitter)

  Locates the laser drawer asset that belongs to the given emitter by indexing its store.

6.23.1 Detailed Description

Provides reuse utility for laser assets so that there is no new object created for every laser interaction. Laser assets are identified by the LaserEmitter they are connected with.

For example, when an emitter hits an object multiple times, the same beam renderer will be used, as identified by emitter id.

Type Constraints

T* : *Component

6.23.2 Constructor & Destructor Documentation

6.23.2.1 LaserDrawerCache()

```
Laser.Cache.LaserDrawerCache< T >.LaserDrawerCache (
            T template,
            Transform parent )
```

6.23.3 Member Function Documentation

6.23.3.1 LocateLaserAsset()

```
T Laser.Cache.LaserDrawerCache< T >.LocateLaserAsset (
            LaserEmitter emitter )
```

Locates the laser drawer asset that belongs to the given emitter by indexing its store.

Creates a new asset by using the added template or instantiating a new asset if needed.

Parameters

emitter	The emitter the drawer asset belongs to

Returns

the found or created drawer asset that belongs to the emitter.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Cache/LaserDrawerCache.cs

6.24 Laser.drawing.helper.LaserDtoTransformer Class Reference

Utility class to transform between laser dto-s.

Public Member Functions

- LaserLineEndpoints Transform (LaserHit laserHit)
  Create the start and endpoints of the laser beam based on the laser hit data.
- LaserLineEndpoints Transform (LaserDirection laserDirection)
  Create the start and endpoints of the laser beam based on the laser's origin and distance (if it didn't connect).

6.24.1 Detailed Description

Utility class to transform between laser dto-s.

Note - transform as not in a 3D environment but convert one dto to another.

6.24.2 Member Function Documentation

6.24.2.1 Transform() [1/2]

```
LaserLineEndpoints Laser.drawing.helper.LaserDtoTransformer.Transform (
            LaserDirection laserDirection )
```

Create the start and endpoints of the laser beam based on the laser's origin and distance (if it didn't connect).

Parameters

laserDirection	the direction and distance data of the laser

Returns

the start and endpoints of the beam

6.24.2.2 Transform() [2/2]

```
LaserLineEndpoints Laser.drawing.helper.LaserDtoTransformer.Transform (
            LaserHit laserHit )
```

Create the start and endpoints of the laser beam based on the laser hit data.

Parameters

laserHit	the hit data

Returns

the start and endpoints of the beam

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Helper/LaserDtoTransformer.cs

6.25 Laser.Logic.LaserEmitter Class Reference

Emits laser forward that interacts with ILaserReceivers and ILaserTargets.

Inherits MonoBehaviour, and Laser.Logic.ILaserEmitter.

Public Member Functions

- void Activate ()

  Used to activate an inactive LaserEmitter. Starts the emission coroutine.
- void Deactivate ()

  Used to deactivate an active LaserEmitter. Stops the emission coroutine.
- bool Equals (IQueryableLaserActor other)
- Transform FindLaserRoot ()

  Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.
- override string ToString ()

  Overrides ToString method to help diagnose problems.

Public Attributes

- string[ ] supportedLayers = new string[1]

Properties

- ISet< IQueryableLaserReceiver > AffectedReceivers `[get]`

  Returns a copy of the ILaserReceivers that this emitter affects.

Events

- IQueryableLaserForwarder.NotifyLaserHitActor LaserHitActor
- IQueryableLaserForwarder.NotifyLaserHitNonActor LaserHitNonActor
- IQueryableLaserForwarder.NotifyLaserMiss LaserMiss
- IQueryableLaserEmitter.NotifyActivated EmitterActivated
- IQueryableLaserEmitter.NotifyDeactivated EmitterDeactivated
- IQueryableLaserEmitter.NotifyChainReturned ChainReturned

6.25.1 Detailed Description

Emits laser forward that interacts with ILaserReceivers and ILaserTargets.

6.25.2 Member Function Documentation

6.25.2.1 Activate()

```
void Laser.Logic.LaserEmitter.Activate ( )
```

Used to activate an inactive LaserEmitter. Starts the emission coroutine.

Implements Laser.Logic.ILaserEmitter.

6.25.2.2 Deactivate()

```
void Laser.Logic.LaserEmitter.Deactivate ( )
```

Used to deactivate an active LaserEmitter. Stops the emission coroutine.

Implements Laser.Logic.ILaserEmitter.

6.25.2.3 Equals()

```
bool Laser.Logic.LaserEmitter.Equals (
            IQueryableLaserActor other )
```

6.25.2.4 FindLaserRoot()

```
Transform Laser.Logic.LaserEmitter.FindLaserRoot ( )
```

Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.

Relatively expensive to execute so make sure to cache the result at startup.

Returns

the root of the laser actor as marked by the developer or null if no such script is added to any of the parents.

Implements Laser.Logic.Query.IQueryableLaserActor.

6.25.2.5 ToString()

```
override string Laser.Logic.LaserEmitter.ToString ( )
```

Overrides ToString method to help diagnose problems.

Returns

Implements Laser.Logic.Query.IQueryableLaserActor.

6.25.3 Member Data Documentation

6.25.3.1 supportedLayers

```
string [] Laser.Logic.LaserEmitter.supportedLayers = new string[1]
```

6.25.4 Property Documentation

6.25.4.1 AffectedReceivers

```
ISet<IQueryableLaserReceiver> Laser.Logic.LaserEmitter.AffectedReceivers  [get]
```

Returns a copy of the ILaserReceivers that this emitter affects.

Note that depending on the scripts' execution order this may be off by one shoot iteration.

Implements Laser.Logic.Query.IQueryableLaserEmitter.

6.25.5 Event Documentation

6.25.5.1 ChainReturned

```
IQueryableLaserEmitter.NotifyChainReturned Laser.Logic.LaserEmitter.ChainReturned
```

6.25.5.2 EmitterActivated

```
IQueryableLaserEmitter.NotifyActivated Laser.Logic.LaserEmitter.EmitterActivated
```

6.25.5.3 EmitterDeactivated

```
IQueryableLaserEmitter.NotifyDeactivated Laser.Logic.LaserEmitter.EmitterDeactivated
```

6.25.5.4 LaserHitActor

```
IQueryableLaserForwarder.NotifyLaserHitActor Laser.Logic.LaserEmitter.LaserHitActor
```

6.25.5.5 LaserHitNonActor

```
IQueryableLaserForwarder.NotifyLaserHitNonActor Laser.Logic.LaserEmitter.LaserHitNonActor
```

6.25.5.6 LaserMiss

```
IQueryableLaserForwarder.NotifyLaserMiss Laser.Logic.LaserEmitter.LaserMiss
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserEmitter.cs

6.26 Laser.dto.LaserHit Struct Reference

Contains information on a laser hit from the emitter to the target it seeks.

Public Member Functions

- LaserHit WithHistory (IEnumerable< int > raycastHistory)

  Builder pattern member that replaces the raycast history of this hit.
- LaserHit WithHitRelay (int objectInstanceId)

  Builder pattern member that adds the relay to the history of this hit.

Static Public Member Functions

- static LaserHit Create (RaycastHit hit, Vector3 origin, Vector3 direction, LaserEmitter emitter, float distance, int layers)

  Used to create a new instance. Initializes the raycast history so it is not null.

Properties

- RaycastHit Hit `[get]`

  The RaycastHit Unity returned after casting the ray.
- Vector3 Origin `[get]`

  Origin of the raycast.
- Vector3 Direction `[get]`

  Direction of the raycast.
- LaserEmitter Emitter `[get]`

  The emitter the raycast belongs to.
- ISet< int > RaycastHistory `[get]`

  The laser actors (their InstanceIDs, to be precise) that this emitter has already hit as part of this chain.
- float Distance `[get]`

  The remaining distance this raycast has.
- int SupportedLayers `[get]`

  The layers this raycast takes into account.

6.26.1 Detailed Description

Contains information on a laser hit from the emitter to the target it seeks.

6.26.2 Member Function Documentation

6.26.2.1 Create()

```
static LaserHit Laser.dto.LaserHit.Create (
            RaycastHit hit,
            Vector3 origin,
            Vector3 direction,
            LaserEmitter emitter,
            float distance,
            int layers )  [static]
```

Used to create a new instance. Initializes the raycast history so it is not null.

Parameters

hit	the raycast hit
origin	the origin of the raycast
direction	direction of the raycast
emitter	the emitter that sent the ray
distance	the distance that the ray has left to travel

Returns

the new laser hit instance

6.26.2.2 WithHistory()

```
LaserHit Laser.dto.LaserHit.WithHistory (
            IEnumerable< int > raycastHistory )
```

Builder pattern member that replaces the raycast history of this hit.

Parameters

raycastHistory	the new history

Returns

this hit instance to chain the builder

6.26.2.3 WithHitRelay()

```
LaserHit Laser.dto.LaserHit.WithHitRelay (
            int objectInstanceId )
```

Builder pattern member that adds the relay to the history of this hit.

Parameters

objectInstanceId	the InstanceID of the relay we add to the history of this hit

Returns

this hit instance to chain the builder

6.26.3 Property Documentation

6.26.3.1 Direction

```
Vector3 Laser.dto.LaserHit.Direction  [get]
```

Direction of the raycast.

6.26.3.2 Distance

```
float Laser.dto.LaserHit.Distance  [get]
```

The remaining distance this raycast has.

6.26.3.3 Emitter

```
LaserEmitter Laser.dto.LaserHit.Emitter  [get]
```

The emitter the raycast belongs to.

6.26.3.4 Hit

```
RaycastHit Laser.dto.LaserHit.Hit  [get]
```

The RaycastHit Unity returned after casting the ray.

6.26.3.5 Origin

```
Vector3 Laser.dto.LaserHit.Origin  [get]
```

Origin of the raycast.

6.26.3.6 RaycastHistory

```
ISet<int> Laser.dto.LaserHit.RaycastHistory  [get]
```

The laser actors (their InstanceIDs, to be precise) that this emitter has already hit as part of this chain.

Needed to prevent an infinite loop inside the chain.

6.26.3.7 SupportedLayers

```
int Laser.dto.LaserHit.SupportedLayers  [get]
```

The layers this raycast takes into account.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserHit.cs

6.27 Laser.dto.LaserLineEndpoints Struct Reference

Contains information for the laser beam to draw.

Static Public Member Functions

- static LaserLineEndpoints Create (Vector3 origin, Vector3 destination)

  Creates a new instance from a raycast origin and destination

Properties

- Vector3 Origin [get]

  The start point of the laser beam.
- Vector3 Destination [get]

  The end point of the laser beam.

6.27.1 Detailed Description

Contains information for the laser beam to draw.

6.27.2 Member Function Documentation

6.27.2.1 Create()

```
static LaserLineEndpoints Laser.dto.LaserLineEndpoints.Create (
            Vector3 origin,
            Vector3 destination )  [static]
```

Creates a new instance from a raycast origin and destination

Parameters

origin	raycast origin
destination	raycast destination

Returns

a new instance with the data included

6.27.3 Property Documentation

6.27.3.1 Destination

```
Vector3 Laser.dto.LaserLineEndpoints.Destination  [get]
```

The end point of the laser beam.

6.27.3.2 Origin

```
Vector3 Laser.dto.LaserLineEndpoints.Origin  [get]
```

The start point of the laser beam.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserLineEndpoints.cs

6.28 Laser.Logic.LaserMirror Class Reference

Forwards the incoming laser by reflecting it.

Inherits Laser.Logic.LaserRelay.

Protected Member Functions

- override Vector3 CalculateLaserDirection (LaserHit incomingHit)

  The direction to which the laser will be forwarded to.
- override Vector3 CalculateLaserOrigin (LaserHit incomingHit)

  The origin point of the forwarded laser.

Additional Inherited Members

6.28.1 Detailed Description

Forwards the incoming laser by reflecting it.

6.28.2 Member Function Documentation

6.28.2.1 CalculateLaserDirection()

```
override Vector3 Laser.Logic.LaserMirror.CalculateLaserDirection (
            LaserHit incomingHit )  [protected], [virtual]
```

The direction to which the laser will be forwarded to.

Parameters

incomingHit	the incoming laser hit

Returns

the direction of the forwarded laser

Implements Laser.Logic.LaserRelay.

6.28.2.2 CalculateLaserOrigin()

```
override Vector3 Laser.Logic.LaserMirror.CalculateLaserOrigin (
            LaserHit incomingHit )  [protected], [virtual]
```

The origin point of the forwarded laser.

When implementing this be mindful that the origin point should not be inside the collider of the relay GameObject.

Parameters

incomingHit	the incoming laser hit

Returns

the origin point of the forwarded laser

Implements Laser.Logic.LaserRelay.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserMirror.cs

6.29 Laser.Drawing.Controller.LaserParticleController Class Reference

Draws particles for the laser system. Includes an optional light that is switched on or off (if exists) when the particle effects are active.

Inherits MonoBehaviour.

Public Member Functions

- void Play (LaserHit hit, Color tint)

  Play the particle system at the hit point, pointing to its origin.
- void Play (Color tint)

  Play the particle system at the origin of this GameObject.
- void Stop ()

  Stop playing the particles and light immediately.

6.29.1 Detailed Description

Draws particles for the laser system. Includes an optional light that is switched on or off (if exists) when the particle effects are active.

6.29.2 Member Function Documentation

6.29.2.1 Play() [1/2]

```
void Laser.Drawing.Controller.LaserParticleController.Play (
            Color tint )
```

Play the particle system at the origin of this GameObject.

Parameters

tint	the color of the particles and the light

6.29.2.2 Play() [2/2]

```
void Laser.Drawing.Controller.LaserParticleController.Play (
            LaserHit hit,
            Color tint )
```

Play the particle system at the hit point, pointing to its origin.

Parameters

hit	the hit data from the laser hit
tint	the color of the particles and the light

6.29.2.3 Stop()

```
void Laser.Drawing.Controller.LaserParticleController.Stop ( )
```

Stop playing the particles and light immediately.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Controller/LaserParticleController.cs

6.30 Laser.Cache.LaserReceiverCache Class Reference

Helper class for the LaserEmitter to avoid GetComponent invocations every frame.

Public Member Functions

- void Purge ()
  Clears the laser receiver cache.

Properties

- static LaserReceiverCache Instance `[get]`
  Returns the Singleton instance.

6.30.1 Detailed Description

Helper class for the LaserEmitter to avoid GetComponent invocations every frame.

6.30.2 Member Function Documentation

6.30.2.1 Purge()

```
void Laser.Cache.LaserReceiverCache.Purge ( )
```

Clears the laser receiver cache.

Should be called every time a new laser component is added to a GameObject that already existed in the scene. See FindOrCache for more info.

6.30.3 Property Documentation

6.30.3.1 Instance

```
LaserReceiverCache Laser.Cache.LaserReceiverCache.Instance  [static], [get]
```

Returns the Singleton instance.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Cache/LaserReceiverCache.cs

6.31 Laser.debug.LaserReceiverDebugger Class Reference

Simple debugger for the laser receiver.

Inherits MonoBehaviour.

6.31.1 Detailed Description

Simple debugger for the laser receiver.

Changes its color if it is being affect by a laser and reverts the change if the laser stops affecting it.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Debug/LaserReceiverDebugger.cs

6.32 Laser.Logic.LaserRelay Class Reference

Base class for any laser object that will forward the incoming laser.

Inherits MonoBehaviour, Laser.Logic.ILaserReceiver, and Laser.Logic.Query.IQueryableLaserRelay.

Inherited by Laser.Logic.LaserMirror, Laser.Logic.LaserRepeater, and Laser.Logic.NonBlockingLaserReceiver.

Public Member Functions

- bool Equals (IQueryableLaserActor other)
- Transform FindLaserRoot ()

  Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.
- override string ToString ()

  Overrides ToString method to help diagnose problems.

Protected Member Functions

- abstract Vector3 CalculateLaserDirection (LaserHit incomingHit)

  The direction to which the laser will be forwarded to.
- abstract Vector3 CalculateLaserOrigin (LaserHit incomingHit)

  The origin point of the forwarded laser.

Static Protected Attributes

- const float Delta = 0.05f

Properties

- ISet< LaserEmitter > AttachedEmitters [get]

  Contains the LaserEmitters that are currently affecting this target.
- int AttachedEmitterCount [get]

  The number of emitters attached to the target.

Events

- IQueryableLaserRelay.NotifyLaserHitActor LaserHitActor
- IQueryableLaserRelay.NotifyLaserHitNonActor LaserHitNonActor
- IQueryableLaserRelay.NotifyLaserMiss LaserMiss
- IQueryableLaserRelay.NotifyHitByLaser HitByLaser
- IQueryableLaserRelay.NotifyAllHitsCeased AllLaserHitsCeased
- ILaserReceiver.LaserEmitterAttached OnNewEmitterReceived
- ILaserReceiver.LaserEmitterDetached OnEmitterDetached

6.32.1 Detailed Description

Base class for any laser object that will forward the incoming laser.

6.32.2 Member Function Documentation

6.32.2.1 CalculateLaserDirection()

```
abstract Vector3 Laser.Logic.LaserRelay.CalculateLaserDirection (
            LaserHit incomingHit )  [protected], [pure virtual]
```

The direction to which the laser will be forwarded to.

Parameters

incomingHit	the incoming laser hit

Returns

the direction of the forwarded laser

Implemented in Laser.Logic.LaserMirror, Laser.Logic.LaserRepeater, and Laser.Logic.NonBlockingLaserReceiver.

6.32.2.2 CalculateLaserOrigin()

```
abstract Vector3 Laser.Logic.LaserRelay.CalculateLaserOrigin (
            LaserHit incomingHit )  [protected], [pure virtual]
```

The origin point of the forwarded laser.

When implementing this be mindful that the origin point should not be inside the collider of the relay GameObject.

Parameters

incomingHit	the incoming laser hit

Returns

the origin point of the forwarded laser

Implemented in Laser.Logic.LaserMirror, Laser.Logic.LaserRepeater, and Laser.Logic.NonBlockingLaserReceiver.

6.32.2.3 Equals()

```
bool Laser.Logic.LaserRelay.Equals (
            IQueryableLaserActor other )
```

6.32.2.4 FindLaserRoot()

```
Transform Laser.Logic.LaserRelay.FindLaserRoot ( )
```

Finds the compound Laser Actor's root. Put this on e.g. a laser emitter cube so that when moving the cube the emitter doesn't fall apart.

Relatively expensive to execute so make sure to cache the result at startup.

Returns

the root of the laser actor as marked by the developer or null if no such script is added to any of the parents.

Implements Laser.Logic.Query.IQueryableLaserActor.

6.32.2.5 ToString()

```
override string Laser.Logic.LaserRelay.ToString ( )
```

Overrides ToString method to help diagnose problems.

Returns

Implements Laser.Logic.Query.IQueryableLaserActor.

6.32.3 Member Data Documentation

6.32.3.1 Delta

```
const float Laser.Logic.LaserRelay.Delta = 0.05f  [static], [protected]
```

6.32.4 Property Documentation

6.32.4.1 AttachedEmitterCount

```
int Laser.Logic.LaserRelay.AttachedEmitterCount  [get]
```

The number of emitters attached to the target.

Implements Laser.Logic.Query.IQueryableLaserReceiver.

6.32.4.2 AttachedEmitters

```
ISet<LaserEmitter> Laser.Logic.LaserRelay.AttachedEmitters  [get]
```

Contains the LaserEmitters that are currently affecting this target.

Note that it may not be accurate frame by frame due to the timing of the LaserEmitter.

Implements Laser.Logic.Query.IQueryableLaserReceiver.

6.32.5 Event Documentation

6.32.5.1 AllLaserHitsCeased

`IQueryableLaserRelay.NotifyAllHitsCeased Laser.Logic.LaserRelay.AllLaserHitsCeased`

6.32.5.2 HitByLaser

`IQueryableLaserRelay.NotifyHitByLaser Laser.Logic.LaserRelay.HitByLaser`

6.32.5.3 LaserHitActor

`IQueryableLaserRelay.NotifyLaserHitActor Laser.Logic.LaserRelay.LaserHitActor`

6.32.5.4 LaserHitNonActor

`IQueryableLaserRelay.NotifyLaserHitNonActor Laser.Logic.LaserRelay.LaserHitNonActor`

6.32.5.5 LaserMiss

`IQueryableLaserRelay.NotifyLaserMiss Laser.Logic.LaserRelay.LaserMiss`

6.32.5.6 OnEmitterDetached

`ILaserReceiver.LaserEmitterDetached Laser.Logic.LaserRelay.OnEmitterDetached`

6.32.5.7 OnNewEmitterReceived

`ILaserReceiver.LaserEmitterAttached Laser.Logic.LaserRelay.OnNewEmitterReceived`

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserRelay.cs

6.33 Laser.Logic.LaserRepeater Class Reference

Forwards the incoming laser in the GameObject's forward direction.

Inherits Laser.Logic.LaserRelay.

Protected Member Functions

- override Vector3 CalculateLaserDirection (LaserHit incomingHit)

  The direction to which the laser will be forwarded to.
- override Vector3 CalculateLaserOrigin (LaserHit incomingHit)

  The origin point of the forwarded laser.

Additional Inherited Members

6.33.1 Detailed Description

Forwards the incoming laser in the GameObject's forward direction.

6.33.2 Member Function Documentation

6.33.2.1 CalculateLaserDirection()

```
override Vector3 Laser.Logic.LaserRepeater.CalculateLaserDirection (
            LaserHit incomingHit )  [protected], [virtual]
```

The direction to which the laser will be forwarded to.

Parameters

incomingHit	the incoming laser hit

Returns

the direction of the forwarded laser

Implements Laser.Logic.LaserRelay.

6.33.2.2 CalculateLaserOrigin()

```
override Vector3 Laser.Logic.LaserRepeater.CalculateLaserOrigin (
            LaserHit incomingHit )  [protected], [virtual]
```

The origin point of the forwarded laser.

When implementing this be mindful that the origin point should not be inside the collider of the relay GameObject.

Parameters

incomingHit	the incoming laser hit

Returns

the origin point of the forwarded laser

Implements Laser.Logic.LaserRelay.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserRepeater.cs

6.34 Laser.dto.LaserResult Struct Reference

Contains information that is returned back to the emitter upon contact with a receiver.

Static Public Member Functions

- static LaserResult Create (ILaserTarget target)
  Used to create a new result instance based on a single hit target.
- static LaserResult Empty ()
  Used to create an empty instance.

Properties

- List< ILaserReceiver > AffectedReceivers `[get]`
  The list of receivers that is affected by this laser emitter (either directly or indirectly via relays).

6.34.1 Detailed Description

Contains information that is returned back to the emitter upon contact with a receiver.

6.34.2 Member Function Documentation

6.34.2.1 Create()

```
static LaserResult Laser.dto.LaserResult.Create (
            ILaserTarget target )  [static]
```

Used to create a new result instance based on a single hit target.

Parameters

target	the target that was hit. A new collection is created and the target is added to it.

Returns

a new LaserResult instance

6.34.2.2 Empty()

```
static LaserResult Laser.dto.LaserResult.Empty ( )  [static]
```

Used to create an empty instance.

The targets collection is instantiated for later use.

Returns

a new, empty instance

6.34.3 Property Documentation

6.34.3.1 AffectedReceivers

```
List<ILaserReceiver> Laser.dto.LaserResult.AffectedReceivers  [get]
```

The list of receivers that is affected by this laser emitter (either directly or indirectly via relays).

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserResult.cs

6.35 Laser.exceptions.MissingLaserAssetException Class Reference

Inherits Exception.

Public Member Functions

- MissingLaserAssetException (string message)

6.35.1 Constructor & Destructor Documentation

6.35.1.1 MissingLaserAssetException()

```
Laser.exceptions.MissingLaserAssetException.MissingLaserAssetException (
            string message )
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Exceptions/MissingLaserAssetException.cs

6.36 Laser.drawing.MultiTargetLaserDrawer Class Reference

Handles the drawing of Laser Actors that repeat multiple incoming beams, each with its own particle systems, like LaserMirror

Inherits Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

Public Attributes

- string mirrorParticlesPath = "Prefabs/mirror_particles"
- LaserParticleController mirrorParticlesTemplate

Protected Member Functions

- override void Start ()
- override void SetUpSubscriptions ()
- override void TearDownSubscriptions ()
- override void OnHitByLaser (IQueryableLaserReceiver sender, LaserHit incomingHit)
- override void OnLaserHitCeased (IQueryableLaserReceiver sender, LaserEmitter laserEmitter)

Additional Inherited Members

6.36.1 Detailed Description

Handles the drawing of Laser Actors that repeat multiple incoming beams, each with its own particle systems, like LaserMirror

6.36.2 Member Function Documentation

6.36.2.1 OnHitByLaser()

```
override void Laser.drawing.MultiTargetLaserDrawer.OnHitByLaser (
            IQueryableLaserReceiver sender,
            LaserHit incomingHit )  [protected]
```

6.36.2.2 OnLaserHitCeased()

```
override void Laser.drawing.MultiTargetLaserDrawer.OnLaserHitCeased (
            IQueryableLaserReceiver sender,
            LaserEmitter laserEmitter )  [protected]
```

6.36.2.3 SetUpSubscriptions()

```
override void Laser.drawing.MultiTargetLaserDrawer.SetUpSubscriptions ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

6.36.2.4 Start()

```
override void Laser.drawing.MultiTargetLaserDrawer.Start ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

6.36.2.5 TearDownSubscriptions()

```
override void Laser.drawing.MultiTargetLaserDrawer.TearDownSubscriptions ( )  [protected],
[virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

6.36.3 Member Data Documentation

6.36.3.1 mirrorParticlesPath

```
string Laser.drawing.MultiTargetLaserDrawer.mirrorParticlesPath = "Prefabs/mirror_particles"
```

6.36.3.2 mirrorParticlesTemplate

```
LaserParticleController Laser.drawing.MultiTargetLaserDrawer.mirrorParticlesTemplate
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/MultiTargetLaserDrawer.cs

6.37 Laser.Logic.NonBlockingLaserReceiver Class Reference

Target that forwards the incoming laser, creating an illusion that it is pass-through.

Inherits Laser.Logic.LaserRelay, and Laser.Logic.ILaserTarget.

Protected Member Functions

- override Vector3 CalculateLaserOrigin (LaserHit incomingHit)

  The origin point of the forwarded laser.
- override Vector3 CalculateLaserDirection (LaserHit incomingHit)

  The direction to which the laser will be forwarded to.

Additional Inherited Members

6.37.1 Detailed Description

Target that forwards the incoming laser, creating an illusion that it is pass-through.

6.37.2 Member Function Documentation

6.37.2.1 CalculateLaserDirection()

```
override Vector3 Laser.Logic.NonBlockingLaserReceiver.CalculateLaserDirection (
            LaserHit incomingHit )  [protected], [virtual]
```

The direction to which the laser will be forwarded to.

Parameters

incomingHit	the incoming laser hit

Returns

the direction of the forwarded laser

Implements Laser.Logic.LaserRelay.

6.37.2.2 CalculateLaserOrigin()

```
override Vector3 Laser.Logic.NonBlockingLaserReceiver.CalculateLaserOrigin (
            LaserHit incomingHit )  [protected], [virtual]
```

The origin point of the forwarded laser.

When implementing this be mindful that the origin point should not be inside the collider of the relay GameObject.

Parameters

incomingHit	the incoming laser hit

Returns

the origin point of the forwarded laser

Implements Laser.Logic.LaserRelay.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/NonBlockingLaserReceiver.cs

6.38 Laser.drawing.NonEmittingLaserDrawer Class Reference

Handles the drawings for the BlockingLaserReceiver.

Inherits MonoBehaviour.

Public Attributes

- string activationParticlesAssetPath
- LaserParticleController activationParticlesTemplate
- Transform activationParticlesOrigin

6.38.1 Detailed Description

Handles the drawings for the BlockingLaserReceiver.

6.38.2 Member Data Documentation

6.38.2.1 activationParticlesAssetPath

```
string Laser.drawing.NonEmittingLaserDrawer.activationParticlesAssetPath
```

6.38.2.2 activationParticlesOrigin

```
Transform Laser.drawing.NonEmittingLaserDrawer.activationParticlesOrigin
```

6.38.2.3 activationParticlesTemplate

```
LaserParticleController Laser.drawing.NonEmittingLaserDrawer.activationParticlesTemplate
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/NonEmittingLaserDrawer.cs

6.39 Laser.Drawing.Coloring.ParticleColorBlender Class Reference

Blends between two particle colors and returns the mixed value as a combined particle color.

Public Member Functions

- ParticleColorBlender AddEmitterWithColor (int emitterId, Color tint)
  Adds a new color, identified by its emitter's InstanceID to the blend.
- ParticleColorBlender RemoveEmitter (int emitterId)
  Remove a new color, identified by its emitter's InstanceID from the blend.
- Color CalculateColor ()
  The build method of the builder: it calculates the mixed colors
- void PurgeRegistry ()
  Removes every color from the blend.

6.39.1 Detailed Description

Blends between two particle colors and returns the mixed value as a combined particle color.

Used when multiple laser beams affect a single activation particle system.

6.39.2 Member Function Documentation

6.39.2.1 AddEmitterWithColor()

```
ParticleColorBlender Laser.Drawing.Coloring.ParticleColorBlender.AddEmitterWithColor (
            int emitterId,
            Color tint )
```

Adds a new color, identified by its emitter's InstanceID to the blend.

Parameters

emitter↩ Id	the emitter Instance ID whose color is being added
tint	the color being added to the blender

Returns

the blender instance for the builder pattern

6.39.2.2 CalculateColor()

```
Color Laser.Drawing.Coloring.ParticleColorBlender.CalculateColor ( )
```

The build method of the builder: it calculates the mixed colors

Returns

the result of the color blend operation

6.39.2.3 PurgeRegistry()

```
void Laser.Drawing.Coloring.ParticleColorBlender.PurgeRegistry ( )
```

Removes every color from the blend.

6.39.2.4 RemoveEmitter()

```
ParticleColorBlender Laser.Drawing.Coloring.ParticleColorBlender.RemoveEmitter (
            int emitterId )
```

Remove a new color, identified by its emitter's InstanceID from the blend.

Parameters

emitter↩ Id	the emitter's InstanceID whose color will be removed

Returns

the blender instance for the builder pattern

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Coloring/ParticleColorBlender.cs

6.40 Laser.Scripting.ScriptingExample Class Reference

Example to show that by extending the LaserActorAware class all the queries and events inside it are made available. To see the full list of available queries, see LaserActorAware.

Inherits Laser.Scripting.LaserActorAware.

Protected Member Functions

- override void OnDestroy ()

Additional Inherited Members

6.40.1 Detailed Description

Example to show that by extending the LaserActorAware class all the queries and events inside it are made available. To see the full list of available queries, see LaserActorAware.

6.40.2 Member Function Documentation

6.40.2.1 OnDestroy()

```
override void Laser.Scripting.ScriptingExample.OnDestroy ( )  [protected], [virtual]
```

Reimplemented from Laser.Scripting.LaserActorAware.

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Scripting/ScriptingExample.cs

6.41 Laser.drawing.SingleTargetLaserDrawer Class Reference

Handles the drawings of Laser Actors that only use one activation particle system when hit, like LaserRepeater and NonBlockingLaserReceiver.

Inherits Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

Public Attributes

- string activationParticlesPath
- LaserParticleController activationParticlesTemplate
- Transform activationParticlesOrigin

Protected Member Functions

- override void Start ()
- override void OnHitByLaser (IQueryableLaserReceiver sender, LaserHit incomingHit)
- override void OnLaserHitCeased (IQueryableLaserReceiver sender, LaserEmitter emitter)
- override void SetUpSubscriptions ()
- override void TearDownSubscriptions ()

Additional Inherited Members

6.41.1 Detailed Description

Handles the drawings of Laser Actors that only use one activation particle system when hit, like LaserRepeater and NonBlockingLaserReceiver.

Has an optional locator to set as a location for the particle system. Must be set at editor, or the particles will reset to the GameObject's location.

6.41.2 Member Function Documentation

6.41.2.1 OnHitByLaser()

```
override void Laser.drawing.SingleTargetLaserDrawer.OnHitByLaser (
            IQueryableLaserReceiver sender,
            LaserHit incomingHit )  [protected]
```

6.41.2.2 OnLaserHitCeased()

```
override void Laser.drawing.SingleTargetLaserDrawer.OnLaserHitCeased (
            IQueryableLaserReceiver sender,
            LaserEmitter emitter )  [protected]
```

6.41.2.3 SetUpSubscriptions()

```
override void Laser.drawing.SingleTargetLaserDrawer.SetUpSubscriptions ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

6.41.2.4 Start()

```
override void Laser.drawing.SingleTargetLaserDrawer.Start ( )  [protected], [virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

6.41.2.5 TearDownSubscriptions()

```
override void Laser.drawing.SingleTargetLaserDrawer.TearDownSubscriptions ( )  [protected],
[virtual]
```

Reimplemented from Laser.drawing.ForwarderLaserDrawer< IQueryableLaserRelay >.

6.41.3 Member Data Documentation

6.41.3.1 activationParticlesOrigin

```
Transform Laser.drawing.SingleTargetLaserDrawer.activationParticlesOrigin
```

6.41.3.2 activationParticlesPath

```
string Laser.drawing.SingleTargetLaserDrawer.activationParticlesPath
```

6.41.3.3 activationParticlesTemplate

```
LaserParticleController Laser.drawing.SingleTargetLaserDrawer.activationParticlesTemplate
```

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

- D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/SingleTargetLaserDrawer.cs

Chapter 7

File Documentation

7.1 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Cache/LaserDrawerCache.cs File Reference

Classes

- class Laser.Cache.LaserDrawerCache< T >

  Provides reuse utility for laser assets so that there is no new object created for every laser interaction. Laser assets are identified by the LaserEmitter they are connected with.

Namespaces

- namespace Laser
- namespace Laser.Cache

7.2 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Cache/LaserReceiverCache.cs File Reference

Classes

- class Laser.Cache.LaserReceiverCache

  Helper class for the LaserEmitter to avoid GetComponent invocations every frame.

Namespaces

- namespace Laser
- namespace Laser.Cache

7.3 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Debug/DebugLaserDrawer.cs File Reference

Classes

- class Laser.debug.DebugLaserDrawer

  Draws debug rays to show where the lasers are cast.

Namespaces

- namespace Laser
- namespace Laser.debug

7.4 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Debug/LaserReceiverDebugger.cs File Reference

Classes

- class Laser.debug.LaserReceiverDebugger

  Simple debugger for the laser receiver.

Namespaces

- namespace Laser
- namespace Laser.debug

7.5 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Coloring/IgnoreColoring.cs File Reference

Classes

- class Laser.Drawing.Coloring.IgnoreColoring

  Particle systems with this component will be ignored by the LaserParticleController when finding the particle materials to color.

Namespaces

- namespace Laser
- namespace Laser.Drawing
- namespace Laser.Drawing.Coloring

7.6 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/↩ Coloring/LaserColorRegistry.cs File Reference

Classes

- class Laser.Drawing.Coloring.LaserColorRegistry

  Contains info on the selected color of a given EmitterLaserDrawer and its matching LaserEmitter.

Namespaces

- namespace Laser
- namespace Laser.Drawing
- namespace Laser.Drawing.Coloring

7.7 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/↩ Coloring/ParticleColorBlender.cs File Reference

Classes

- class Laser.Drawing.Coloring.ParticleColorBlender

  Blends between two particle colors and returns the mixed value as a combined particle color.

Namespaces

- namespace Laser
- namespace Laser.Drawing
- namespace Laser.Drawing.Coloring

7.8 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/↩ Controller/LaserBeamController.cs File Reference

Classes

- class Laser.Drawing.Controller.LaserBeamController

  Encapsulates the logic needed to interact with the laser beam's shader.

Namespaces

- namespace Laser
- namespace Laser.Drawing
- namespace Laser.Drawing.Controller

7.9 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/↩ Controller/LaserParticleController.cs File Reference

Classes

- class Laser.Drawing.Controller.LaserParticleController

  Draws particles for the laser system. Includes an optional light that is switched on or off (if exists) when the particle effects are active.

Namespaces

- namespace Laser
- namespace Laser.Drawing
- namespace Laser.Drawing.Controller

7.10 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Emitter↩ LaserDrawer.cs File Reference

Classes

- class Laser.drawing.EmitterLaserDrawer

  Draws beams and activation particle effects for a LaserEmitter.

Namespaces

- namespace Laser
- namespace Laser.drawing

7.11 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/↩ ForwarderLaserDrawer.cs File Reference

Classes

- class Laser.drawing.ForwarderLaserDrawer< T >

  Base class for all laser forwarders (and Laser Actor that can shoot a laser - either its own like a LaserEmitter or someone else's like a LaserRelay).

Namespaces

- namespace Laser
- namespace Laser.drawing

7.12 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/Helper/LaserDtoTransformer.cs File Reference

Classes

- class Laser.drawing.helper.LaserDtoTransformer

  Utility class to transform between laser dto-s.

Namespaces

- namespace Laser
- namespace Laser.drawing
- namespace Laser.drawing.helper

7.13 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/MultiTargetLaserDrawer.cs File Reference

Classes

- class Laser.drawing.MultiTargetLaserDrawer

  Handles the drawing of Laser Actors that repeat multiple incoming beams, each with its own particle systems, like LaserMirror

Namespaces

- namespace Laser
- namespace Laser.drawing

7.14 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/NonEmittingLaserDrawer.cs File Reference

Classes

- class Laser.drawing.NonEmittingLaserDrawer

  Handles the drawings for the BlockingLaserReceiver.

Namespaces

- namespace Laser
- namespace Laser.drawing

7.15 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Drawing/SingleTargetLaserDrawer.cs File Reference

Classes

- class Laser.drawing.SingleTargetLaserDrawer

  Handles the drawings of Laser Actors that only use one activation particle system when hit, like LaserRepeater and NonBlockingLaserReceiver.

Namespaces

- namespace Laser
- namespace Laser.drawing

7.16 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserDirection.cs File Reference

Classes

- struct Laser.dto.LaserDirection

  Contains information on a laser hit from

Namespaces

- namespace Laser
- namespace Laser.dto

7.17 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserHit.cs File Reference

Classes

- struct Laser.dto.LaserHit

  Contains information on a laser hit from the emitter to the target it seeks.

Namespaces

- namespace Laser
- namespace Laser.dto

7.18 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/LaserLine↩ Endpoints.cs File Reference

Classes

- struct Laser.dto.LaserLineEndpoints

  Contains information for the laser beam to draw.

Namespaces

- namespace Laser
- namespace Laser.dto

7.19 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Dto/Laser↩ Result.cs File Reference

Classes

- struct Laser.dto.LaserResult

  Contains information that is returned back to the emitter upon contact with a receiver.

Namespaces

- namespace Laser
- namespace Laser.dto

7.20 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Examples/↩ EmitterActivationSwitcher.cs File Reference

Classes

- class Laser.Examples.EmitterActivationSwitcher

Namespaces

- namespace Laser
- namespace Laser.Examples

7.21 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Exceptions/↩ MissingLaserAssetException.cs File Reference

Classes

- class Laser.exceptions.MissingLaserAssetException

Namespaces

- namespace Laser
- namespace Laser.exceptions

7.22 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Blocking↩ LaserReceiver.cs File Reference

Classes

- class Laser.Logic.BlockingLaserReceiver

  A laser target that does not forward the incoming ray (is not pass through).

Namespaces

- namespace Laser
- namespace Laser.Logic

7.23 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/ILaser↩ Emitter.cs File Reference

Classes

- interface Laser.Logic.ILaserEmitter

  A modifiable contract that expresses a laser emitter.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.24 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/ILaser↩ Receiver.cs File Reference

Classes

- interface Laser.Logic.ILaserReceiver

  Contains methods for the emitters to call when they hit this receiver. Interface exposed for custom receiver implementations.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.25 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/ILaser↩ Target.cs File Reference

Classes

- interface Laser.Logic.ILaserTarget

  Denotes that this receiver is an end target for an emitter. As an ILaserReceiver, contains methods to be called by the emitter to modify this target. Exposed for custom target implementations.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.26 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Laser↩ ActorRoot.cs File Reference

Classes

- class Laser.Logic.LaserActorRoot

  Marks a complex hierarchy's root that contains a Laser Actor. Useful when we want to move e.g. the whole emitter cube (whose children contain the actual emitter script).

Namespaces

- namespace Laser
- namespace Laser.Logic

7.27 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Laser↩ Emitter.cs File Reference

Classes

- class Laser.Logic.LaserEmitter

  Emits laser forward that interacts with ILaserReceivers and ILaserTargets.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.28 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserMirror.cs File Reference

Classes

- class Laser.Logic.LaserMirror

  Forwards the incoming laser by reflecting it.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.29 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserRelay.cs File Reference

Classes

- class Laser.Logic.LaserRelay

  Base class for any laser object that will forward the incoming laser.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.30 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/LaserRepeater.cs File Reference

Classes

- class Laser.Logic.LaserRepeater

  Forwards the incoming laser in the GameObject's forward direction.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.31 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/NonBlockingLaserReceiver.cs File Reference

Classes

- class Laser.Logic.NonBlockingLaserReceiver

  Target that forwards the incoming laser, creating an illusion that it is pass-through.

Namespaces

- namespace Laser
- namespace Laser.Logic

7.32 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/↩ IQueryableLaserActor.cs File Reference

Classes

- interface Laser.Logic.Query.IQueryableLaserActor

  Contains the most basic information about Laser Actors and serves as a base.

Namespaces

- namespace Laser
- namespace Laser.Logic
- namespace Laser.Logic.Query

7.33 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/↩ IQueryableLaserEmitter.cs File Reference

Classes

- interface Laser.Logic.Query.IQueryableLaserEmitter

  Provides information about a LaserEmitter.

Namespaces

- namespace Laser
- namespace Laser.Logic
- namespace Laser.Logic.Query

7.34 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/↩ IQueryableLaserForwarder.cs File Reference

Classes

- interface Laser.Logic.Query.IQueryableLaserForwarder

  Provides information about a Laser Actor that can shoot a laser (either its own like a LaserEmitter or someone else's like a LaserRelay)

Namespaces

- namespace Laser
- namespace Laser.Logic
- namespace Laser.Logic.Query

7.35 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/↩ IQueryableLaserReceiver.cs File Reference

Classes

- interface Laser.Logic.Query.IQueryableLaserReceiver

  Provides information about a Laser Actor that can be affected by the system's lasers.

Namespaces

- namespace Laser
- namespace Laser.Logic
- namespace Laser.Logic.Query

7.36 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/↩ IQueryableLaserRelay.cs File Reference

Classes

- interface Laser.Logic.Query.IQueryableLaserRelay

  Provides a type to query for information about relays.

Namespaces

- namespace Laser
- namespace Laser.Logic
- namespace Laser.Logic.Query

7.37 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Logic/Query/↩ IQueryableLaserTarget.cs File Reference

Classes

- interface Laser.Logic.Query.IQueryableLaserTarget

  Differentiates between regular receivers and targets (constructs that are a final target for an emitter).

Namespaces

- namespace Laser
- namespace Laser.Logic
- namespace Laser.Logic.Query

7.38 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/ResourceLoading/LaserComponentLoader.cs File Reference

Classes

- class Laser.ResourceLoading.LaserComponentLoader< T >

  Utility class to get a component from an object by either
- class Laser.ResourceLoading.LaserComponentLoader< T >.Builder

  Builder class for the resource loader.

Namespaces

- namespace Laser
- namespace Laser.ResourceLoading

Typedefs

- using Object = UnityEngine.Object

7.38.1 Typedef Documentation

7.38.1.1 Object

```
using Object = UnityEngine.Object
```

7.39 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Scripting/LaserActorAware.cs File Reference

Classes

- class Laser.Scripting.LaserActorAware

  Utility to speed up the work with Laser Actors. Contains various queries implemented about laser actors in the scene. Ideally you want to rely on the events found in interfaces and run these queries on event invocation.

Namespaces

- namespace Laser
- namespace Laser.Scripting

7.40 D:/ProjectFiles/unity/Lasers/Assets/Scripts/Laser/Scripting/↩ ScriptingExample.cs File Reference

Classes

- class Laser.Scripting.ScriptingExample

  Example to show that by extending the LaserActorAware class all the queries and events inside it are made available. To see the full list of available queries, see LaserActorAware.

Namespaces

- namespace Laser
- namespace Laser.Scripting

Index