Grab System

1.1.1

Generated by Doxygen 1.9.5

Chapter 1

Namespace Index

1.1 Package List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 GrabSystem Namespace Reference

Classes

- class ConditionalGrabber

  The base component for all grabbers that implement the 'ConditionalGrab()' method. Also implements repeat grab attempt routine functionality.
- class DistanceCatchAssist

  A component that is generally NOT attached to the same GameObject as a DistanceGrabber that must be referenced to enable assisted catching. This is NOT attached to the same GameObject as a DistanceGrabber to avoid triggering unwanted components.
- class DistanceGrabbable

  A component that can be attached to the same GameObject as a GrabbableObject that allows a Grabber to instantly grab it from a distance.
- class DistanceGrabber

  A component that is attached to the same GameObject as a Grabber to enable it to grab DistanceGrabbable objects from a distance.
- class ForceDistanceGrabbable

  A component that is derived from 'DistanceGrabbable' that allows for force based grabbing of a GrabbableObject that also has a Rigidbody component. Modes: Instant - Teleports the object to the grabber and grabs it instantly (as a normal DistanceGrabbable would). Arch - Uses physics forces to pull the GameObject as an arch.
- class GrabbableChildObject

  A component that makes a GrabbableObject grabbable via a child collider.
- class GrabbableObject

  A component that makes an object grabbable by a Grabber.
- class Grabber

  The base component for all grabbers.
- class GrabberUnityEvent

  Arg0: Grabber - The Grabber involved in this grab event.
- class GrabUnityEvent

  Arg0: Grabber - The Grabber involved in this grab event. Arg1: GrabbableObject - The GrabbableObject that was involved in the event.
- class RayGrabber

  A component that extends a Grabber by adding the ability to attempt grabs using a raycast.

4.2 GrabSystem.Collisions Namespace Reference

Classes

- class DelayUnignoreCollidersOnRelease

  A simple component that references a IgnoreCollidersWhileGrabbing component that uses the 'OverrideUnignore↩ Delegate' delegate event to delay the unginore collisions for some time.
- class IgnoreCollidersWhileGrabbing

  A component intended to be attached to a GrabbableObject that subscribes to the relevant 'Grabbed' and 'Released' events to ignore collisions between some colliders and a Grabber while it is holding an object.

4.3 GrabSystem.Delegates Namespace Reference

Functions

- delegate void ActionRef< T > (ref T pItem)

  A simple delegate for events where an argument is passed by reference.
- delegate void ActionRef< VALUE_ONE, T > (VALUE_ONE pValueOne, ref T pItem)

  A simple delegate for events where an argument is passed by reference.
- delegate void ActionRef< VALUE_ONE, VALUE_TWO, T > (VALUE_ONE pValueOne, VALUE_TWO p↩ ValueTwo, ref T pItem)

  A simple delegate for events where an argument is passed by reference.
- delegate void ActionRef< VALUE_ONE, VALUE_TWO, VALUE_THREE, T > (VALUE_ONE pValueOne, VALUE_TWO pValueTwo, VALUE_THREE pValueThree, ref T pItem)

  A simple delegate for events where an argument is passed by reference.

4.3.1 Function Documentation

4.3.1.1 ActionRef< T >()

```
delegate void GrabSystem.Delegates.ActionRef< T > (
            ref T pItem )
```

A simple delegate for events where an argument is passed by reference.

Template Parameters

T	The type of the passed reference.

Parameters

pItem	The reference that was passed.

4.3.1.2 ActionRef< VALUE_ONE, T >()

```
delegate void GrabSystem.Delegates.ActionRef< VALUE_ONE, T > (
            VALUE_ONE pValueOne,
            ref T pItem )
```

A simple delegate for events where an argument is passed by reference.

Template Parameters

VALUE_ONE	The type of the fist passed value.
T	The type of the passed reference.

Parameters

pValueOne	The first value that was passed.
pItem	The reference that was passed.

4.3.1.3 ActionRef< VALUE_ONE, VALUE_TWO, T >()

```
delegate void GrabSystem.Delegates.ActionRef< VALUE_ONE, VALUE_TWO, T > (
            VALUE_ONE pValueOne,
            VALUE_TWO pValueTwo,
            ref T pItem )
```

A simple delegate for events where an argument is passed by reference.

Template Parameters

VALUE_ONE	The type of the first passed value.
VALUE_TWO	The type of the second passed value.
T	The type of the passed reference.

Parameters

pValueOne	The first value that was passed.
pValueTwo	The second value that was passed.
pItem	The reference that was passed.

4.3.1.4 ActionRef< VALUE_ONE, VALUE_TWO, VALUE_THREE, T >()

```
delegate void GrabSystem.Delegates.ActionRef< VALUE_ONE, VALUE_TWO, VALUE_THREE, T > (
            VALUE_ONE pValueOne,
```

```
    VALUE_TWO pValueTwo,
    VALUE_THREE pValueThree,
    ref T pItem )
```

A simple delegate for events where an argument is passed by reference.

Template Parameters

VALUE_ONE	The type of the first passed value.
VALUE_TWO	The type of the second passed value.
VALUE_THREE	The type of the third passed value.
T	The type of the passed reference.

Parameters

pValueOne	The first value that was passed.
pValueTwo	The second value that was passed.
pValueThree	The third value that was passed.
pItem	The reference that was passed.

4.4 GrabSystem.Math Namespace Reference

Classes

- class **FloatMath**

  A public static class that provides helper functions for floating-point related math operations.
- class **VectorMath**

  A static class that contains vector math functions.

4.5 GrabSystem.Poses Namespace Reference

Classes

- class MaintainOffsetByGrabOffset

  A component that is attached to the same GameObject as a GrabbableObject (that is assumed to be using the 'Maintain Offset' grab mode) which overrides the maintain offset behaviour allowing more complex behaviours to be defined using the grab offset of the Grabber from the grabbable at the time it grabbed the GrabbableObject. NOTE: This component works in the grabbables local space by comparing the 'signed distance in direction' of the grabber from the grabbable at the time of grab using the comparator specified in the entry.
- class MaintainOffsetByProjectedAngle

  A component that is attached to the same GameObject as a GrabbableObject (that is assumed to be using the 'Maintain Offset' grab mode) which overrides the maintain offset behaviour allowing more complex behaviours to be defined using the relative angle of the GrabbableObject from the Grabber at the time it is grabbed. NOTE: This component works by projecting the 'grabber' onto the local 'axis' in local space of the grabbable and getting the hands angle around that axis.

4.6 GrabSystem.Structures Namespace Reference

Classes

- class AxisToggleSettings
- struct FloatMinMax

4.7 GrabSystem.Triggers Namespace Reference

Classes

- class GrabArea

  A simple component that uses the 'OnTriggerEnter' and 'OnTriggerExit' callbacks to allow a ConditionalGrabber that enters a trigger to automatically grab a specified GrabbableObject when it fails to grab anything else after a grab attempt while it remains in the trigger.
- class GrabberReleaseArea

  A simple component that uses the 'OnTriggerEnter' and 'OnTriggerExit' callbacks to force a grabber to release on object when it enters the relevnat trigger. Optionally prevents the hand from grabbing while in the release.

4.8 GrabSystem.Utility Namespace Reference

Classes

- class **ComponentUtility**

  A public static class that provides Compoennt related helper functions.

Chapter 5

Class Documentation

5.1 GrabSystem.Structures.AxisToggleSettings Class Reference

Public Attributes

- bool **x**

  Toggle the setting for the
- bool **y**

  Toggle the setting for the
- bool **z**

  Toggle the setting for the

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

- AxisToggleSettings.cs

5.2 GrabSystem.RayGrabber.BoxCastData Class Reference

Public Member Functions

- **BoxCastData** ()

  Instantiaets a BoxCastData instance.
- BoxCastData (BoxCastData pOther)

  Instantiates a copy of some other BoxCastData, pOther.

Public Attributes

- Transform **origin**

  The
- float **boxHeight**

  How high of a box to cast. This is the height of the box in units
- float **boxWidth**

  The width of the box to cast in units.
- float **boxLength**

  The length of the box to cast in units.

5.2.1 Constructor & Destructor Documentation

5.2.1.1 BoxCastData()

```
GrabSystem.RayGrabber.BoxCastData.BoxCastData (
            BoxCastData pOther )
```

Instantiates a copy of some other BoxCastData, pOther.

Parameters

pOther	

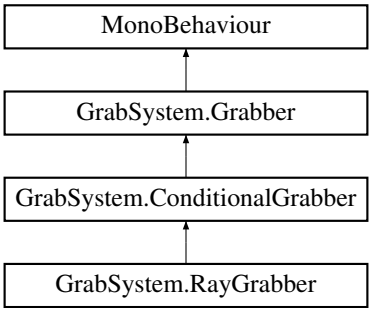
The documentation for this class was generated from the following file:

- RayGrabber.cs

5.3 GrabSystem.ConditionalGrabber Class Reference

The base component for all grabbers that implement the 'ConditionalGrab()' method. Also implements repeat grab attempt routine functionality.

Inheritance diagram for GrabSystem.ConditionalGrabber:



Public Member Functions

- void **StartGrabRoutine** ()

  Starts a grab routine that continuously attempts grabs til successful or stopped.
- void **StopGrabRoutine** ()

  Stops any ongoing continous grab attempt routine.
- void **AttemptGrab** ()

  A simple wrapper for TryGrab() that has no return value. Useful for use with Unity Editor events.
- GrabbableObject TryGrab ()

  Attempts a conditional grab using the overrideable 'ConditionalGrab()' method..
- bool CheckInterruptGrabRoutine ()

  Polls all 'ShouldInterruptGrabRoutineDelegate' event followers (and any other and returns true if this component will interrupt a grab based on the subscribers to the 'ShouldInterruptGrabRoutineDelegate' event, otherwise false. Note that interrupted grab routines will not end the routine, they will simply prevent the routine from attempting a grab.
- abstract GrabbableObject ConditionalGrab ()

  Performs a conditional grab attempt and returns the GrabbableObject that was grabbed, otherwise null.

Public Attributes

- GrabberUnityEvent **PreGrabAttempted**

  An event that is invoked just before a grab is attempted.
- GrabberUnityEvent **GrabAttemptRepeated**

  Invoked whenever a grab routine initiates a repeat grab attempt.
- GrabberUnityEvent **GrabAttemptSucceeded**

  Invoked whenever a grab attempt succeeds at grabbing a valid GrabbableObject.
- GrabberUnityEvent **GrabAttemptFailed**

  Invoked whenever a grab attempt fails at grabbing a valid GrabbableObject.

Properties

- Coroutine **GrabRoutine** `[get]`

  A reference to the coroutine for an ongoing continous grab attempt or null if none ongoing.

Events

- ActionRef< bool > **ShouldInterruptGrabRoutineDelegate**

  A simple event that has a single argument, a reference to a boolean value, if this boolean value becomes true the conditional grab is interrupted.

Additional Inherited Members

5.3.1 Detailed Description

The base component for all grabbers that implement the 'ConditionalGrab()' method. Also implements repeat grab attempt routine functionality.

The 'ShouldInterruptGrab' event uses a reference to a boolean value as its first argument, if a subscribing callback makes this boolean true any conditional grab will be interrupted. Note that interrupting a grab routine does not end the grab routine, it simply prevents it from attempting any grab. Author: Mathew Aloisio

5.3.2 Member Function Documentation

5.3.2.1 CheckInterruptGrabRoutine()

```
bool GrabSystem.ConditionalGrabber.CheckInterruptGrabRoutine ( )
```

Polls all 'ShouldInterruptGrabRoutineDelegate' event followers (and any other and returns true if this component will interrupt a grab based on the subscribers to the 'ShouldInterruptGrabRoutineDelegate' event, otherwise false. Note that interrupted grab routines will not end the routine, they will simply prevent the routine from attempting a grab.

Returns

true if this component will interrupt a grab routine based on the subscribers to the 'ShouldInterruptGrab' event, otherwise false.

5.3.2.2 ConditionalGrab()

```
abstract GrabbableObject GrabSystem.ConditionalGrabber.ConditionalGrab ( )  [pure virtual]
```

Performs a conditional grab attempt and returns the GrabbableObject that was grabbed, otherwise null.

Returns

the GrabbableObject that was grabbed, otherwise null.

Implemented in GrabSystem.RayGrabber.

5.3.2.3 TryGrab()

```
GrabbableObject GrabSystem.ConditionalGrabber.TryGrab ( )
```

Attempts a conditional grab using the overrideable 'ConditionalGrab()' method..

Returns

A reference to the GrabbableObject that was grabbed, otherwise null.

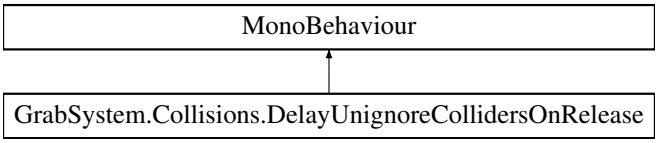
The documentation for this class was generated from the following file:

- ConditionalGrabber.cs

5.4 GrabSystem.Collisions.DelayUnignoreCollidersOnRelease Class Reference

A simple component that references a IgnoreCollidersWhileGrabbing component that uses the 'OverrideUnignore↩ Delegate' delegate event to delay the unginore collisions for some time.

Inheritance diagram for GrabSystem.Collisions.DelayUnignoreCollidersOnRelease:



Classes

- struct UnignoreEntry

Public Attributes

- float **unignoreIn** = 0.375f

  Unignore collisions this many seconds after
- IgnoreCollidersWhileGrabbing **ignoreComponent**

  A reference to the IgnoreCollidersWhileGrabbing component assocaited with this component.

5.4.1 Detailed Description

A simple component that references a IgnoreCollidersWhileGrabbing component that uses the 'OverrideUnignore↩ Delegate' delegate event to delay the unginore collisions for some time.

This component references an IgnoreCollidersWhileGrabbing component instead of being forced onto the same GameObject because frequently a developer may want to disable the relevant GameObject after release, this allows this component to be put on a separate non-deactivated GameObject while still allowing the unignore collisions delay. Author: Mathew Aloisio

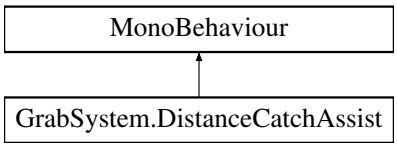
The documentation for this class was generated from the following file:

- DelayUnignoreCollidersOnRelease.cs

5.5 GrabSystem.DistanceCatchAssist Class Reference

A component that is generally NOT attached to the same GameObject as a DistanceGrabber that must be referenced to enable assisted catching. This is NOT attached to the same GameObject as a DistanceGrabber to avoid triggering unwanted components.

Inheritance diagram for GrabSystem.DistanceCatchAssist:



Public Attributes

- DistanceGrabber **distanceGrabber**

  The distance grabber this component assists with catching.
- bool **snapGrab**

  Should this component snap grabbable objects that is assisted catches to the grab target using the distance grab hit point?
- GrabUnityEvent **CatchAssisted**

  An event that is invoked when a catch assist is compelted.
  \nArg

5.5.1 Detailed Description

A component that is generally NOT attached to the same GameObject as a DistanceGrabber that must be referenced to enable assisted catching. This is NOT attached to the same GameObject as a DistanceGrabber to avoid triggering unwanted components.

When a distance grabbable that is being distance grabbed by the relevant distance grabber triggers this component an instantenous grab is performed. Author: Mathew Aloisio

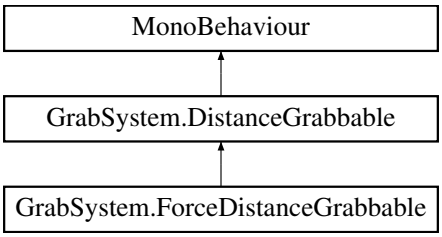
The documentation for this class was generated from the following file:

- DistanceCatchAssist.cs

5.6 GrabSystem.DistanceGrabbable Class Reference

A component that can be attached to the same GameObject as a GrabbableObject that allows a Grabber to instantly grab it from a distance.

Inheritance diagram for GrabSystem.DistanceGrabbable:



Public Member Functions

- bool CanDistanceGrab (Grabber pGrabber)

  Returns true if pGrabber can distance grab this, otherwise false. NOTE: This does not check all conditions for the Grabber, the Grabber may have its own conditions not specific to this Grabbable. See 'DistanceGrabber.CanDistanceGrab(DistanceGrabbable)'.
- void InstantGrab (Vector3 pHitPoint)

  Instantly grabs a distance grabbable.
- void SetTarget (DistanceGrabber pGrabber, Vector3 pHitPoint)

  Targets the specified DistanceGrabber, or none if null. The distance grabbable will immediate start being 'grabbed' (teleported or pulled).
- void **ClearTarget** ()

  Clears any target of the DistanceGrabbable.
- void Highlight (Grabber pGrabber)

  Tells this DistanceGrabbable that pGrabber has started highlighting it (if it is not already highlighting it).
- void Unhighlight (Grabber pGrabber)

  Tells this DistanceGrabbable that pGrabber has stopped highlighting it (if it is already highlighting it).
- Grabber GetHighlightedBy (int pIndex)

  Returns the Grabber that is highlighting this distance grabbable by index.

Public Attributes

- bool **rotate**

  Should this distance grabbable rotate towards the specified rotateTo?
- Vector3 **rotateTo**

  The local space euler angles relative to the DistanceGrabber.GrabTarget doing the distance grabbing the object should rotate towards while being distance grabbed.
- GrabUnityEvent **DistanceGrabStarted**

  An event that is invoked whenever a distance grab is started.
  \nArg
- GrabUnityEvent **DistanceGrabStopped**

  An event that is invoked whenver a distance grab is stopped.
  \nArg
- GrabUnityEvent **InstantGrabbed**

  An event that is invoked whenever a distance grabbable is
- GrabberUnityEvent **StartedHighlighting**

  An event that is invoked when a Grabber starts highlighting this distance grabbable.
  \nArg
- GrabberUnityEvent **StoppedHighlighting**

  An event that is invoked when a Grabber stops highlighting this distance grabbable.
  \nArg
- GrabberUnityEvent **AllStoppedHighlighting**

  An event that is invoked when the last Grabber stops highlighting this distance grabbable.
  \nArg

Protected Member Functions

- virtual void **Awake** ()
- virtual void **OnEnable** ()
- virtual void **OnDisable** ()
- virtual void OnDistanceGrabStarted (Vector3 pHitPoint)

  Invoked when a distance grab is started.
- virtual void **OnDistanceGrabStopped** ()

  Invoked when a distance grab is stopped.

Properties

- GrabbableObject **Grabbable** `[get]`

  The GrabbableObject component that this component is converting into a distance grabbable.
- DistanceGrabber **Target** `[get]`

  A reference to the GrabbableObject being targetted by distance DistanceGrabbable, otherwise null.
- DistanceGrabber **LastTarget** `[get]`

  The last set target of the distance grabbable, otherwise null if there was none. This reference remains valid after the distance grab is stopped. (This updates immediately when a new valid target is set and will at that point be equal to Target.)
- Vector3 **TargetLocalHitPoint** `[get]`

  The point the target hit this distance grabbable at during selection in the distance grabbables local space. This is only valid while Target != null.
- int **HighlightedByCount** `[get]`

  Returns the number of Grabbers currently highlighting this distance grabbable.

Events

- ActionRef< Grabber, GrabbableObject, bool > **CanDistanceGrabDelegate**

  A delegate even that allows distance grab permissions to be overridden by subscribing to this event and modifying the 'ref bool' argument (argument 3). Arg0: Grabber - The Grabber doing the grabbing. This Grabber is guarenteed to have a DistanceGrabber component. Arg1: GrabbableObject - The GrabbableObject being distance grabbed. This GrabbableObject is guarenteed to have a DistanceGrabbable component. Arg2: ref bool - The reference to a boolean that when changed to false (or true) forces distance grabs for this object o be denied (or approved).

5.6.1 Detailed Description

A component that can be attached to the same GameObject as a GrabbableObject that allows a Grabber to instantly grab it from a distance.

Author: Mathew Aloisio

5.6.2 Member Function Documentation

5.6.2.1 CanDistanceGrab()

```
bool GrabSystem.DistanceGrabbable.CanDistanceGrab (
            Grabber pGrabber )
```

Returns true if pGrabber can distance grab this, otherwise false. NOTE: This does not check all conditions for the Grabber, the Grabber may have its own conditions not specific to this Grabbable. See 'DistanceGrabber.CanDistanceGrab(DistanceGrabbable)'.

By default this method uses 'pGrabber.CanGrab(Grabbable)' to check distance grab permission. This method invokes the callback 'CanDistanceGrabDelegate' which gives scripts an opportunity to override distance grab permissions.

Parameters

pGrabber	

Returns

true if pGrabber can distance grab this distance grabbable, otherwise false.

5.6.2.2 GetHighlightedBy()

```
Grabber GrabSystem.DistanceGrabbable.GetHighlightedBy (
            int pIndex )
```

Returns the Grabber that is highlighting this distance grabbable by index.

Parameters

pIndex	

5.6.2.3 Highlight()

```
void GrabSystem.DistanceGrabbable.Highlight (
            Grabber pGrabber )
```

Tells this DistanceGrabbable that pGrabber has started highlighting it (if it is not already highlighting it).

Parameters

pGrabber	

5.6.2.4 InstantGrab()

```
void GrabSystem.DistanceGrabbable.InstantGrab (
            Vector3 pHitPoint )
```

Instantly grabs a distance grabbable.

Parameters

pHitPoint	The 'hit point' the grabber targeted.

5.6.2.5 OnDistanceGrabStarted()

```
virtual void GrabSystem.DistanceGrabbable.OnDistanceGrabStarted (
            Vector3 pHitPoint )  [protected], [virtual]
```

Invoked when a distance grab is started.

Parameters

pHitPoint	The world space 'hit point' the grabber targeted in world space.

Reimplemented in GrabSystem.ForceDistanceGrabbable.

5.6.2.6 SetTarget()

```
void GrabSystem.DistanceGrabbable.SetTarget (
            DistanceGrabber pGrabber,
            Vector3 pHitPoint )
```

Targets the specified DistanceGrabber, or none if null. The distance grabbable will immediate start being 'grabbed' (teleported or pulled).

Parameters

pGrabber	The DistanceGrabber that is pulling this distance grabbable.
pHitPoint	The 'hit point' the grabber targeted.

5.6.2.7 Unhighlight()

```
void GrabSystem.DistanceGrabbable.Unhighlight (
            Grabber pGrabber )
```

Tells this DistanceGrabbable that pGrabber has stopped highlighting it (if it is already highlighting it).

Parameters

pGrabber	

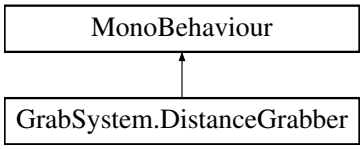
The documentation for this class was generated from the following file:

- DistanceGrabbable.cs

5.7 GrabSystem.DistanceGrabber Class Reference

A component that is attached to the same GameObject as a Grabber to enable it to grab DistanceGrabbable objects from a distance.

Inheritance diagram for GrabSystem.DistanceGrabber:



Public Member Functions

- bool CanDistanceGrab (DistanceGrabbable pGrabbable)

  Returns true if this Grabber can distance grab pGrabbable, otherwise false.
- void StartPull (DistanceGrabbable pGrabbable)

  Stops any existing pull operation and starts pulling the DistanceGrabbable, pGrabbable. NOTE: This method does not check grab permissions, see 'bool CanGrab(DistanceGrabbable)'.
- void StartPull (DistanceGrabbable pGrabbable, Vector3 pHitPoint)

  Stops any existing pull operation and starts pulling the DistanceGrabbable, pGrabbable. NOTE: This method does not check grab permissions, see 'bool CanGrab(DistanceGrabbable)'.
- void **StopPull** ()

  Stops any existing pull operation.

Public Attributes

- GrabUnityEvent **StartedPull**

  An event that is invoked when this component starts pulling a distance grabbable.
  \nArg
- GrabUnityEvent **StoppedPull**

  An event that is invoked when this component stops pulling a distance grabbable.
  \nArg
- GrabUnityEvent **InstantPulled**

  An event that is invoked if a distance grabbable is instant pulled to the grabber.
  \nArg

Properties

- Grabber **Grabber** `[get]`

  A reference to the Grabber associated with this DistanceGrabber.
- Transform **GrabTarget** `[get]`

  A reference to the Transform this distance grabber will teleport or pull distance grabbables to.
- DistanceGrabbable **Pulling** `[get]`

  A reference to the DistanceGrabbable currently being pulled by this component, or null.

Events

- ActionRef< Grabber, GrabbableObject, bool > **CanDistanceGrabDelegate**

  A delegate even that allows distance grab permissions to be overridden by subscribing to this event and modifying the 'ref bool' argument (argument 3). Arg0: Grabber - The Grabber doing the grabbing. This Grabber is guarenteed to have a DistanceGrabber component. Arg1: GrabbableObject - The GrabbableObject being distance grabbed. This GrabbableObject is guarenteed to have a DistanceGrabbable component. Arg2: ref bool - The reference to a boolean that when changed to false (or true) forces distance grabs for this object o be denied (or approved).

5.7.1 Detailed Description

A component that is attached to the same GameObject as a Grabber to enable it to grab DistanceGrabbable objects from a distance.

Author: Mathew Aloisio

5.7.2 Member Function Documentation

5.7.2.1 CanDistanceGrab()

```
bool GrabSystem.DistanceGrabber.CanDistanceGrab (
            DistanceGrabbable pGrabbable )
```

Returns true if this Grabber can distance grab pGrabbable, otherwise false.

Parameters

pGrabbable	

Returns

true if this Grabber can distance grab pGrabbable, otherwise false.

5.7.2.2 StartPull() [1/2]

```
void GrabSystem.DistanceGrabber.StartPull (
            DistanceGrabbable pGrabbable )
```

Stops any existing pull operation and starts pulling the DistanceGrabbable, pGrabbable. NOTE: This method does not check grab permissions, see 'bool CanGrab(DistanceGrabbable)'.

Parameters

pGrabbable	

5.7.2.3 StartPull() [2/2]

```
void GrabSystem.DistanceGrabber.StartPull (
            DistanceGrabbable pGrabbable,
            Vector3 pHitPoint )
```

Stops any existing pull operation and starts pulling the DistanceGrabbable, pGrabbable. NOTE: This method does not check grab permissions, see 'bool CanGrab(DistanceGrabbable)'.

Parameters

pGrabbable	
pHitPoint	The hit point the pull was started from.

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

- DistanceGrabber.cs

5.8 GrabSystem.Poses.MaintainOffsetByGrabOffset.Entry Class Reference

Public Attributes

- Transform **maintainOffsetTransform**

  The Transform to maintain offset using when this entrys conditions are met first.
- Vector3 **localOffset** = Vector3.zero

  The local space
- Vector3 **direction**

  The direction to test the grabber distance in
- FloatMath.Comparator **comparator**

  The comparator to use when testing distance
- float **distanceThreshold**

  The minimum distance in the specified direction the grabber must be to satisfy the conditions.
- GrabberUnityEvent **EntryUsed**

  An event that is invoked when this entry is used by a grabber.
  \nArg

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

- MaintainOffsetByGrabOffset.cs

5.9 GrabSystem.Poses.MaintainOffsetByProjectedAngle.Entry Class Reference

Public Attributes

- Transform **maintainOffsetTransform**

  The Transform to maintain offset using when this entrys conditions are met first.
- Vector3 **upAxis** = Vector3.up

  The upwards direction vector to test against.
- Vector3 **angleAxis** = Vector3.right

  The direction vector axis that defines the plane to project the grabber onto.
- Transform **centerPoint**

  The center point Transform to get the angle around. [If null
- FloatMinMax **angleRange**

  The angle range the
- GrabberUnityEvent **EntryUsed**

  An event that is invoked when this entry is used by a grabber.
  \nArg

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

- MaintainOffsetByProjectedAngle.cs

5.10 GrabSystem.Structures.FloatMinMax Struct Reference

Public Attributes

- float **minimum**

  The minimum value.
- float **maximum**

  The maximum value.

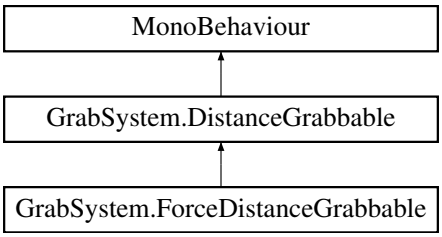
The documentation for this struct was generated from the following file:

- FloatMinMax.cs

5.11 GrabSystem.ForceDistanceGrabbable Class Reference

A component that is derived from 'DistanceGrabbable' that allows for force based grabbing of a GrabbableObject that also has a Rigidbody component. Modes: Instant - Teleports the object to the grabber and grabs it instantly (as a normal DistanceGrabbable would). Arch - Uses physics forces to pull the GameObject as an arch.

Inheritance diagram for GrabSystem.ForceDistanceGrabbable:



Public Types

- enum **PullMode**

Public Member Functions

- Vector3 CalculateTrajectory (Vector3 pOrigin, Vector3 pTarget, float pArch)

  Calculates the trajectory velocity from the target to the origin and returns it as a Vector3.

Public Attributes

- PullMode **pullMode** = PullMode.Arch

  The pull mode to use for the distance grabbable.
  \nInstant
- bool **stopPullOnCollision** = true

  Should this distance grabbable stop being pulled after a collision collisionGrace seconds after pulling starts?
- float **collisionEnterGrace** = 0.2f

  The number of seconds of collision enter grace granted before de
- float **collisionStayGrace** = 0.8f

  The number of seconds of collision stay grace granted before de
- float **archMultiplier** = 0.7f

  The arch strength multiplication factor for the distance grabbable.
- float **pullSpeed** = 1f

  The pull speed scalar for the distance grabbable.
- float **pullVelocityScalar** = 10f

  A scalar that can be used to framerate independently scale pull velocity.
- float **rotateSpeed** = 540f

  If

Protected Member Functions

- virtual void **FixedUpdate** ()
- virtual void **OnCollisionEnter** (Collision pCollision)
- virtual void **OnCollisionStay** (Collision pCollision)
- override void OnDistanceGrabStarted (Vector3 pHitPoint)

  Invoked when a distance grab is started.

Additional Inherited Members

5.11.1 Detailed Description

A component that is derived from 'DistanceGrabbable' that allows for force based grabbing of a GrabbableObject that also has a Rigidbody component. Modes: Instant - Teleports the object to the grabber and grabs it instantly (as a normal DistanceGrabbable would). Arch - Uses physics forces to pull the GameObject as an arch.

Author: Mathew Aloisio

5.11.2 Member Function Documentation

5.11.2.1 CalculateTrajectory()

```
Vector3 GrabSystem.ForceDistanceGrabbable.CalculateTrajectory (
            Vector3 pOrigin,
            Vector3 pTarget,
            float pArch )
```

Calculates the trajectory velocity from the target to the origin and returns it as a Vector3.

Parameters

pOrigin	
pTarget	
pArch	

Returns

a Vector3 representing the trajectory velocity from the target to the origin.

5.11.2.2 OnDistanceGrabStarted()

```
override void GrabSystem.ForceDistanceGrabbable.OnDistanceGrabStarted (
            Vector3 pHitPoint )  [protected], [virtual]
```

Invoked when a distance grab is started.

Parameters

pHitPoint	The world space 'hit point' the grabber targeted in world space.

Reimplemented from GrabSystem.DistanceGrabbable.

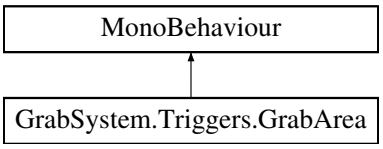
The documentation for this class was generated from the following file:

- ForceDistanceGrabbable.cs

5.12 GrabSystem.Triggers.GrabArea Class Reference

A simple component that uses the 'OnTriggerEnter' and 'OnTriggerExit' callbacks to allow a ConditionalGrabber that enters a trigger to automatically grab a specified GrabbableObject when it fails to grab anything else after a grab attempt while it remains in the trigger.

Inheritance diagram for GrabSystem.Triggers.GrabArea:



Public Attributes

- GrabbableObject **grabbable**

  A reference to the GrabbableObject that can be grabbed using this GrabArea.
- bool **checkRigidbody** = true

  If not found on the triggering Collider

5.12.1 Detailed Description

A simple component that uses the 'OnTriggerEnter' and 'OnTriggerExit' callbacks to allow a ConditionalGrabber that enters a trigger to automatically grab a specified GrabbableObject when it fails to grab anything else after a grab attempt while it remains in the trigger.

Author: Mathew Aloisio

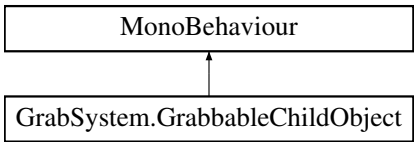
The documentation for this class was generated from the following file:

- GrabArea.cs

5.13 GrabSystem.GrabbableChildObject Class Reference

A component that makes a GrabbableObject grabbable via a child collider.

Inheritance diagram for GrabSystem.GrabbableChildObject:



Public Attributes

- GrabbableObject **grabbable**

  A reference to the GrabbableObject that this component handles grab events for.

5.13.1 Detailed Description

A component that makes a GrabbableObject grabbable via a child collider.

Author: Mathew Aloisio

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

- GrabbableChildObject.cs

5.14 GrabSystem.RayGrabber.GrabbableEntry Struct Reference

Public Attributes

- GrabbableObject **grabbable**

  The GrabbableObject reference for the pair.
- Collider **collider**

  The Collider that may be grabbed.
- float **distance**

  The distance of the grabbable according to the pair.

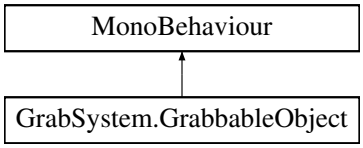
The documentation for this struct was generated from the following file:

- RayGrabber.cs

5.15 GrabSystem.GrabbableObject Class Reference

A component that makes an object grabbable by a Grabber.

Inheritance diagram for GrabSystem.GrabbableObject:



Classes

- class GrabModeChangedUnityEvent

  Arg0: GrabMode - The old grab mode. Arg1: GrabMode - The new grab mode.

Public Types

- enum GrabMode

  The mode used when grabbing this object.

Public Member Functions

- void SetPositionAndRotation (Vector3 pPosition, Quaternion pRotation)

  Sets the position and rotation of this GrabbableObject (and its Rigidbody(s) if relevant.)
- void **UpdateCachedMass** ()

  If this grabbable has a valid Rigidbody overrides the cached 'default' mass value for the object with 'Rigidbody.mass'.
- void **ForceGrabbersRelease** ()

  Forces all Grabbers to release this GrabbableObject.
- void **ForceGrabbersReleaseNoThrow** ()

  Forces all Grabbers to release this GrabbableObject without throwing.
- Vector3 GetGrabOffset (Grabber pGrabber)

  Returns a Vector3 representing the grab offset in the pGrabbers local space for this GrabbableObject. Takes into account axis flip settings.
- Vector3 GetGrabEulerOffset (Grabber pGrabber)

  Returns a Vector3 representing the grab euler offset in the pGrabbers local space for this GrabbableObject. Takes into account axis flip settings.
- Grabber GetHeldBy (int pIndex)

  Returns the Grabber in the given 'held by' index.
- void MaintainOffsetFrom (Grabber pGrabber, Transform pMaintainTransform)

  Maintains a pGrabber at pMaintainTransform.
- void SetGrabEnabled (bool pEnabled)

  Sets the 'grabEnabled' field for the grabbable object. Useful for use with Unity Editor events.
- void **SetGrabMode** (GrabMode pGrabMode)
- void **UseMoveObjectGrabMode** ()

  Forces the GrabbableObject to use the 'MoveObject' grab mode.
- void **UseMaintainOffsetGrabMode** ()

  Forces the GrabbableObject to use the 'MaintainOffset' grab mode.
- virtual bool CanGrab (Grabber pGrabber)

  Returns true if this GrabbableObject can be grabbed by pGrabber, otherwise false. (NOTE: This only checks conditions relevant to the grabbable object, the Grabber may have its own conditions.)

Public Attributes

- bool **grabEnabled** = true

  Can this grabbable object be grabbed?
- bool **makeKinematic**

  Should this Grabbable always be kinematic when grabbed?
- bool **makeWeightless**

  Should this Grabbable be made weightless when grabbed?
- int **grabPriority** = 0

  The grab priority for this grabbable object. When multiple grabbables meet the grab conditions for a Grabber priority will be considered first
- GrabMode **grabMode** = GrabMode.MaintainOffset

  The mode used when this object is grabbed.
  \nNone
- bool **singleGrabberGrab**

  Should this object only be grabbable by a single Grabber at a time?
- bool **grabSwap**
- float **grabBreakForce** = 5000f

  The force required to break the grab joint.
- float **grabBreakTorque** = 5000f

  The torque required to break the grab joint.
- ConfigurableJoint **grabJointPrefab**
- bool **keepGrabOffset**

  When in
- Vector3 **grabOffset**

  When in
- Vector3 **grabEulerOffset**

  When in
- float **throwMultiplier** = 1f

  The multiplier for throw force being applied to this object.
- GrabUnityEvent **PreGrabbed**

  An event that is invoked just before this grabbable object is grabbed.
  \nArg
- GrabUnityEvent **Grabbed**

  An event that is invoked when this grabbable object is grabbed.
  \nArg
- GrabUnityEvent **PreReleased**

  An event that is invoked just before this grabbable object is released.
  \nArg
- GrabUnityEvent **Released**

  An event that is invoked when this grabbable object is released.
  \nArg
- GrabUnityEvent **LastGrabberReleased**

  An event that is invoked when the last Grabber that is holding this grabbable object releases it.
  \nArg
- GrabUnityEvent **GrabJointBroken**

  An event that is invoked after a
- GrabModeChangedUnityEvent **GrabModeChanged**

  An event that is invoked every time the grabMode of this GrabbableObject is changed using the
- ActionRef< Grabber, GrabbableObject, bool > **MaintainOffsetOverrideDelegate**

A delegate that allows the 'Maintain Offset' behaviour to be overridden for this GrabbableObject, when the reference-passed boolean is set to true the default 'Maintain Offset' behaviour will be skipped. This is useful for implementing things such as advanced maintained grab positions that go beyond the scope of the default settings of the GrabbableObject component. Arg0: Grabber - The Grabber who is grabbing the object. Arg1: GrabbableObject - The GrabbableObject being grabbed. Arg2: ref bool - The boolean that controls whether or not to skip the default 'Maintain Offset' behaviour. When true the default behaviour is skipped.

- ActionRef< Grabber, GrabbableObject, Vector3 > **OverrideGrabOffsetDelegate**

  A delegate that is invoked whenever GetGrabOffset(Grabber) is invoked. This is useful for implementing custom behaviours for overriding a grab offset for a GrabbableObject via code. Arg0: Grabber - The Grabber who is grabbing the object. Arg1: GrabbableObject - The GrabbableObject being grabbed. Arg2: ref Vector3 - A reference to the returned grab offset in the grabbers local space.
- ActionRef< Grabber, GrabbableObject, Vector3 > **OverrideGrabEulerOffsetDelegate**

  A delegate that is invoked whenever GetGrabEulerOffset(Grabber) is invoked. This is useful for implementing custom behaviours for overriding a grab euler angle offset for a GrabbableObject via code. Arg0: Grabber - The Grabber who is grabbing the object. Arg1: GrabbableObject - The GrabbableObject being grabbed. Arg2: ref Vector3 - A reference to the returned grab euler offset in the grabbers local space.
- ActionRef< Grabber, GrabbableObject, bool > **CanGrabDelegate**

  A delegate that is invoked whenever 'CanGrab(Grabber)' is invoked. This is useful for overiding grabbability for specific GrabbableObjects by subscribing to this event and modifying the 'ref bool' value. The value may be left unmodified to pass-through the default behaviour. Arg0: Grabber - The Grabber who is attempting to grab the object. Arg1: GrabbableObjet - The GrabbableObject being grabbed. Arg2: ref bool - A reference to the boolean that decides whether or not this grabber can be grabbed by the specified grabber (at least as far as the grabbable itself is concerned).

Protected Member Functions

- virtual void **Awake** ()
- virtual void **Start** ()
- virtual void **Update** ()

Properties

- Rigidbody **Rigidbody** `[get]`

  The Rigidbody component that on the same GameObject as this GrabbableObject, otherwise null if the grabbable is not a Rigidbody.
- int **HeldByCount** `[get]`

  the number of hands currently holding the GrabbableObject.
- Transform **MaintainGrabTransform** `[get]`

  A reference to the Transform a hand that grabs this object will maintain it's offset relative to when enabled. If null the grab point is used instead.
- bool **CanBeGrabbed** `[get]`

  Returns true if this GrabbableObject can be grabbed, otherwise false. (NOTE: Use CanGrab(Grabber) methods instead to test for grabbability by specific Grabbers, this only tests for global 'can grab' properties that apply to any Grabbers.)
- Vector3 **OffsetAtGrab** `[get]`

  Returns 'transform.position - grabber.transform.position' at the time of grab. This offset is in the grabbers local space.
- Quaternion **RotationOffsetAtGrab** `[get]`

  Returns the rotation offset at the time of grab. This offset is in the grabbers local space.

5.15.1 Detailed Description

A component that makes an object grabbable by a Grabber.

NOTE: While in GrabMode.MoveObject the object will move based on the most recent grabbing hand. Author: Mathew Aloisio

5.15.2 Member Function Documentation

5.15.2.1 CanGrab()

```
virtual bool GrabSystem.GrabbableObject.CanGrab (
            Grabber pGrabber )  [virtual]
```

Returns true if this GrabbableObject can be grabbed by pGrabber, otherwise false. (NOTE: This only checks conditions relevant to the grabbable object, the Grabber may have its own conditions.)

Parameters

pGrabber	

Returns

true if this GrabbableObject can be grabbed by pGrabber, otherwise false.

5.15.2.2 GetGrabEulerOffset()

```
Vector3 GrabSystem.GrabbableObject.GetGrabEulerOffset (
            Grabber pGrabber )
```

Returns a Vector3 representing the grab euler offset in the pGrabbers local space for this GrabbableObject. Takes into account axis flip settings.

NOTE: This method invokes 'OverrideGrabEulerOffsetDelegate(Grabber, GrabbableObject, ref Vector3)' for easy overriding.

Parameters

pGrabber	

Returns

a Vector3 representing the grab euler offset in the pGrabbers local space for this GrabbableObject.

5.15.2.3 GetGrabOffset()

```
Vector3 GrabSystem.GrabbableObject.GetGrabOffset (
            Grabber pGrabber )
```

Returns a Vector3 representing the grab offset in the pGrabbers local space for this GrabbableObject. Takes into account axis flip settings.

NOTE: This method invokes 'OverrideGrabOffsetDelegate(Grabber, GrabbableObject, ref Vector3)' for easy overriding.

Parameters

pGrabber	

Returns

a Vector3 representing the grab offset in the pGrabbers local space for this GrabbableObject.

5.15.2.4 GetHeldBy()

```
Grabber GrabSystem.GrabbableObject.GetHeldBy (
            int pIndex )
```

Returns the Grabber in the given 'held by' index.

Parameters

pIndex	

Returns

the Grabber in the given 'held by' index.

5.15.2.5 MaintainOffsetFrom()

```
void GrabSystem.GrabbableObject.MaintainOffsetFrom (
            Grabber pGrabber,
            Transform pMaintainTransform )
```

Maintains a pGrabber at pMaintainTransform.

Parameters

pGrabber	
pMaintainTransform	

5.15.2.6 SetGrabEnabled()

```
void GrabSystem.GrabbableObject.SetGrabEnabled (
            bool pEnabled )
```

Sets the 'grabEnabled' field for the grabbable object. Useful for use with Unity Editor events.

Parameters

pEnabled	

5.15.2.7 SetPositionAndRotation()

```
void GrabSystem.GrabbableObject.SetPositionAndRotation (
            Vector3 pPosition,
            Quaternion pRotation )
```

Sets the position and rotation of this GrabbableObject (and its Rigidbody(s) if relevant.)

Parameters

pPosition	
pRotation	

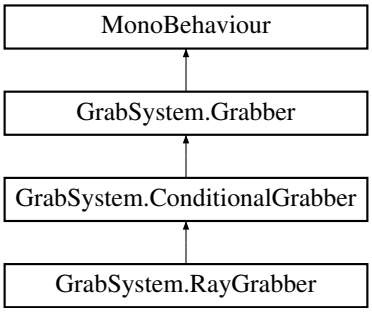
The documentation for this class was generated from the following file:

- GrabbableObject.cs

5.16 GrabSystem.Grabber Class Reference

The base component for all grabbers.

Inheritance diagram for GrabSystem.Grabber:



Public Member Functions

- void Grab (GrabbableObject pGrabbable)

  Forces a GrabbableObject to be grabbed, if an object is already grabbed it is (attempted to be) released. Same as calling 'Grab(pGrabbable, null)'.
- void Grab (GrabbableObject pGrabbable, Collider pGrabCollider)

  Forces a GrabbableObject to be grabbed, if an object is already being grabbed it is (attempted to be) released.
- void GrabAtOffset (GrabbableObject pGrabbable, Vector3 pOffset, Quaternion pRotation)

  If pGrabbable is a movable object, teleports it to pOffset and pRotation local space (angle) offsets relative to the Grabber's 'transform', then grabs.
- void **ReleaseNoThrow** ()

  Releases the currently grabbed object without throwing it only if 'CanRelease()' returns true.
- void **Release** ()

  Releases the currently grabbed object applying any relevant throw forces only if 'CanRelease()' returns true.
- void **ForceReleaseNoThrow** ()

  Releases the currently grabbed object without throwing it.
- void **ForceRelease** ()

  Releases the currently grabbed object applying any relevant throw forces.
- Vector3 GetOffsetFromGrabber (GrabbableObject pGrabbable)

  Returns the offset of pGrabbable from 'this' Grabber as a Vector3 in local space of 'this' Grabber.
- Quaternion GetRelativeRotationFromGrabber (GrabbableObject pGrabbable)

  Returns the relative rotation of pGrabbable from 'this' Grabber as a Quaternion.
- void ThrowRigidbody (Rigidbody pRigidbody, float pThrowForceMultiplier)

  Calculates and applies throw forces to the specified Rigidbody.
- Joint CreateJoint (GrabbableObject pGrabbable, Rigidbody pRigidbody, float pBreakForce, float pBreak↩ Torque, ConfigurableJoint pGrabJointPrefab)

  Creates a physics joint between a Rigidbody and the Grabber.
- Joint GetJointAtIndex (int pIndex)

  Returns the Joint at the given index of the joints array for this Grabber.
- void DestroyJoint (Joint pJoint)

  Destroys a Joint and removes it from this Grabbers tracked joint list if it is in it.
- void **DestroyAllJoints** ()

  Destroys all Joints that were created by this Grabber.
- void SetGrabLocked (bool pLocked)

  Sets the 'grabLocked' field of this component. Useful for use with Unity editor events.
- virtual bool CanGrab (GrabbableObject pGrabbable)

  Returns true if this Grabber can grab pGrabbable, otherwise false. NOTE: By default (when not overridden) it is guarenteed that this method takes into account 'CanGrabDelegate' event overrides.

Public Attributes

- bool **grabLocked**

  Lock this grabber when enabled preventing it from performing non
- Transform **overrideAnchor**

  Allows the grabbed object anchor Transform to be overridden with a specific Transform.
- AxisToggleSettings **flipGrabOffset**

  Allows grab offsets to be flipped per axis.
- AxisToggleSettings **flipGrabAngle**

  Allows grab angles to be flipped per axis.
- AxisToggleSettings **flipMaintainAngle**

  Allows maintain offset angles to be flipped per axis.

- Rigidbody **body**
- float **throwStrength** = 1f

  The force objects at thrown with.
- float **throwThreshold** = 0.01f

  The minimum velocity the grabber must have to apply throw forces to a Rigidbody.
- TransformVelocityTracker **velocityTracker**
- GrabUnityEvent **PreGrabbed**

  An event that is invoked just before a Grabbable object is grabbed by this grabber.
  \nArg
- GrabUnityEvent **Grabbed**

  An event that is invoked when a Grabbable object is grabbed by this grabber.
  \nArg
- GrabUnityEvent **PreReleased**

  An event that is invoked just before a Grabbable object is released by this grabber.
  \nArg
- GrabUnityEvent **Released**

  An event that is invoked when a Grabbable object is released by this grabber.
  \nArg
- GrabUnityEvent **GrabJointBroken**

  An event that is invoked after the

Protected Member Functions

- virtual void **Awake** ()
- virtual void **Reset** ()
- virtual void **LateUpdate** ()
- virtual void **OnDisable** ()
- virtual void **OnDestroy** ()
- virtual void **OnGrabbingJointBroken** ()

  Invoked when Grabber.GrabbingJoint breaks. When overriding this careful to ensure that the 'GrabJointBroken' event is still invoked and (if desired) that any non-null 'Grabbing' object is released.

Properties

- GrabbableObject **Grabbing** `[get]`

  A reference to the object being grabbed by this hand, otherwise null if nothing being grabbed.
- Joint **GrabbingJoint** `[get]`

  A reference to the Joint being used to connect a Rigidbody-containing GrabbableObject, otherwise null.
- Transform **AnchorTransform** `[get]`

  Returns the Transform grabbables are anchored to.
- Transform **GrabberPivot** `[get]`

  Returns the Transform that is used as the pivot point for the Grabber. This is mostly used when positioning the Grabber on a GrabbableObjects 'MaintainGrabTransform' reference.
- Transform **GrabPoint** `[get]`

  A reference to the Transform for the object that was created at the grabber position when the grab occured. Parented to the grabbed object, this means 'GrabPoint' maintains a reference of the hands relative position relative to the grabbed object..
- bool **IsGrabbingNonKinematicRigidbody** `[get]`

  Returns true if the Grabber is currently grabbing a non-kinematic Rigidbody, otherwise false.
- int **JointCount** `[get]`

  Returns the number of joints that are created by this Grabber (No guarentee that some joint entries will not be null if they are destroyed without using Grabber.DestroyJoint).
- bool **CanRelease** `[get]`

  Returns true if this Grabber may release the object it is 'Grabbing', otherwise false. The 'grabLocked' boolean or SetGrabLocked(bool) method may be used to prevent a release. NOTE: The 'CanReleaseDelegate' event may be used to override release permissions.

Events

- ActionRef< Grabber, GrabbableObject, bool > **CanGrabDelegate**

  A simple event that is passed a boolean by reference that may be overridden to override whether or not a grab can occur.
- ActionRef< Grabber, GrabbableObject, bool > **CanReleaseDelegate**

  A simple event that is passed a boolean by reference that may be overridden to override whether or not a non-forced release can occur.

5.16.1 Detailed Description

The base component for all grabbers.

Author: Mathew Aloisio

5.16.2 Member Function Documentation

5.16.2.1 CanGrab()

```
virtual bool GrabSystem.Grabber.CanGrab (
            GrabbableObject pGrabbable )  [virtual]
```

Returns true if this Grabber can grab pGrabbable, otherwise false. NOTE: By default (when not overridden) it is guarenteed that this method takes into account 'CanGrabDelegate' event overrides.

Parameters

pGrabbable	

Returns

true if this Grabber can grab pGrabbable, otherwise false.

5.16.2.2 CreateJoint()

```
Joint GrabSystem.Grabber.CreateJoint (
            GrabbableObject pGrabbable,
            Rigidbody pRigidbody,
            float pBreakForce,
            float pBreakTorque,
            ConfigurableJoint pGrabJointPrefab )
```

Creates a physics joint between a Rigidbody and the Grabber.

Parameters

pGrabbable	The GrabbableObject being jointed to.
pRigidbody	The Rigidbody that is being jointed to the Grabber.
pBreakForce	The force it takes to break the joint.
pBreakTorque	The torque it takes to break the joint.
pGrabJointPrefab	The custom grab joint prefab to use. If null the default joint configuration will be used.

Returns

The Joint that was created.

5.16.2.3 DestroyJoint()

```
void GrabSystem.Grabber.DestroyJoint (
            Joint pJoint )
```

Destroys a Joint and removes it from this Grabbers tracked joint list if it is in it.

Parameters

pJoint	

5.16.2.4 GetJointAtIndex()

```
Joint GrabSystem.Grabber.GetJointAtIndex (
            int pIndex )
```

Returns the Joint at the given index of the joints array for this Grabber.

Parameters

pIndex	

Returns

The Joint at the given index of the joints array for this Grabber.

5.16.2.5 GetOffsetFromGrabber()

```
Vector3 GrabSystem.Grabber.GetOffsetFromGrabber (
            GrabbableObject pGrabbable )
```

Returns the offset of pGrabbable from 'this' Grabber as a Vector3 in local space of 'this' Grabber.

Parameters

pGrabbable	The grabbable object.

Returns

the offset of pGrabbable from 'this' Grabber as a Vector3 in local space of 'this' Grabber.

5.16.2.6 GetRelativeRotationFromGrabber()

```
Quaternion GrabSystem.Grabber.GetRelativeRotationFromGrabber (
            GrabbableObject pGrabbable )
```

Returns the relative rotation of pGrabbable from 'this' Grabber as a Quaternion.

Parameters

pGrabbable	The grabbable object.

Returns

the relative rotation of pGrabbable from 'this' Grabber as a Quaternion.

5.16.2.7 Grab() [1/2]

```
void GrabSystem.Grabber.Grab (
            GrabbableObject pGrabbable )
```

Forces a GrabbableObject to be grabbed, if an object is already grabbed it is (attempted to be) released. Same as calling 'Grab(pGrabbable, null)'.

Parameters

pGrabbable	

5.16.2.8 Grab() [2/2]

```
void GrabSystem.Grabber.Grab (
            GrabbableObject pGrabbable,
            Collider pGrabCollider )
```

Forces a GrabbableObject to be grabbed, if an object is already being grabbed it is (attempted to be) released.

Parameters

pGrabbable	The Grabbable that was grabbed.
pGrabCollider	The Collider that was grabbed.

5.16.2.9 GrabAtOffset()

```
void GrabSystem.Grabber.GrabAtOffset (
            GrabbableObject pGrabbable,
            Vector3 pOffset,
            Quaternion pRotation )
```

If pGrabbable is a movable object, teleports it to pOffset and pRotation local space (angle) offsets relative to the Grabber's 'transform', then grabs.

Parameters

pGrabbable	The grabbable object.
pOffset	The local space offset.
pRotation	The local space rotation.

5.16.2.10 SetGrabLocked()

```
void GrabSystem.Grabber.SetGrabLocked (
            bool pLocked )
```

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

Parameters

pLocked	

5.16.2.11 ThrowRigidbody()

```
void GrabSystem.Grabber.ThrowRigidbody (
            Rigidbody pRigidbody,
            float pThrowForceMultiplier )
```

Calculates and applies throw forces to the specified Rigidbody.

Parameters

pRigidbody	
pThrowForceMultiplier	A throw force multiplier to apply to the calculated throw force. Useful for scaling force for specific objects.

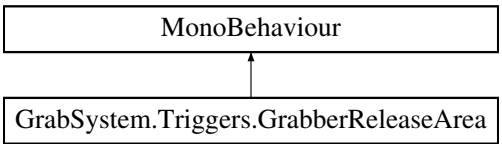
The documentation for this class was generated from the following file:

- Grabber.cs

5.17 GrabSystem.Triggers.GrabberReleaseArea Class Reference

A simple component that uses the 'OnTriggerEnter' and 'OnTriggerExit' callbacks to force a grabber to release on object when it enters the relevnat trigger. Optionally prevents the hand from grabbing while in the release.

Inheritance diagram for GrabSystem.Triggers.GrabberReleaseArea:



Public Types

- enum **ReleaseMode**

Public Attributes

- ReleaseMode **releaseMode** = ReleaseMode.Release

  The release mode to use when a Grabber enters this area.
  \nNone
- bool **forceRelease**

  Should
- bool **checkRigidbody**

  If not found on the triggering Collider
- bool **preventGrabbing**

  Should grabbing be prevented for Grabbers in this area?
- GrabberUnityEvent **Triggered**

  An event that is invoked when a Grabber triggers this component.
  \nArg
- GrabUnityEvent **GrabPrevented**

  An event that is invoked whenever a Grabber is prevented from making a grab due to this component.
  \nArg

5.17.1 Detailed Description

A simple component that uses the 'OnTriggerEnter' and 'OnTriggerExit' callbacks to force a grabber to release on object when it enters the relevnat trigger. Optionally prevents the hand from grabbing while in the release.

Author: Mathew Aloisio

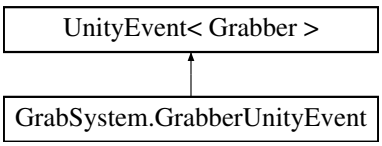
The documentation for this class was generated from the following file:

- GrabberReleaseArea.cs

5.18 GrabSystem.GrabberUnityEvent Class Reference

Arg0: Grabber - The Grabber involved in this grab event.

Inheritance diagram for GrabSystem.GrabberUnityEvent:



5.18.1 Detailed Description

Arg0: Grabber - The Grabber involved in this grab event.

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

- GrabberUnityEvent.cs

5.19 GrabSystem.Poses.MaintainOffsetByGrabOffset.GrabInfo Class Reference

Public Attributes

- Entry **entry**

  A reference to the entry whose conditions were met.

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

- MaintainOffsetByGrabOffset.cs

5.20 GrabSystem.Poses.MaintainOffsetByProjectedAngle.GrabInfo Class Reference

Public Attributes

- Entry **entry**

  A reference to the entry whose conditions were met.

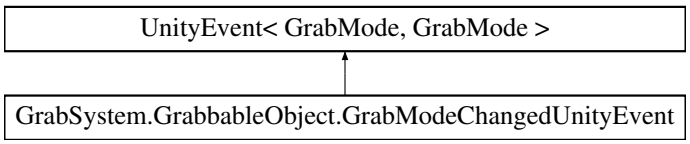
The documentation for this class was generated from the following file:

- MaintainOffsetByProjectedAngle.cs

5.21 GrabSystem.GrabbableObject.GrabModeChangedUnityEvent Class Reference

Arg0: GrabMode - The old grab mode. Arg1: GrabMode - The new grab mode.

Inheritance diagram for GrabSystem.GrabbableObject.GrabModeChangedUnityEvent:



5.21.1 Detailed Description

Arg0: GrabMode - The old grab mode. Arg1: GrabMode - The new grab mode.

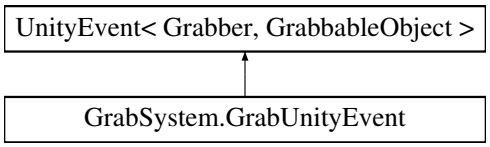
The documentation for this class was generated from the following file:

- GrabbableObject.cs

5.22 GrabSystem.GrabUnityEvent Class Reference

Arg0: Grabber - The Grabber involved in this grab event. Arg1: GrabbableObject - The GrabbableObject that was involved in the event.

Inheritance diagram for GrabSystem.GrabUnityEvent:



5.22.1 Detailed Description

Arg0: Grabber - The Grabber involved in this grab event. Arg1: GrabbableObject - The GrabbableObject that was involved in the event.

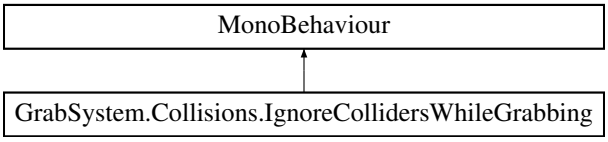
The documentation for this class was generated from the following file:

- GrabUnityEvent.cs

5.23 GrabSystem.Collisions.IgnoreCollidersWhileGrabbing Class Reference

A component intended to be attached to a GrabbableObject that subscribes to the relevant 'Grabbed' and 'Released' events to ignore collisions between some colliders and a Grabber while it is holding an object.

Inheritance diagram for GrabSystem.Collisions.IgnoreCollidersWhileGrabbing:



Public Attributes

- Collider[ ] **colliders**

  An array of colliders to enable/disable collisions against the hand for.
- ActionRef< Grabber, GrabbableObject, Collider[ ], bool > **OverrideUnignoreDelegate**

  A delegate event that allows the 'unignore collisions' behaviour to be overridden by anything that subscribes to this event. The boolean argument is passed as a reference, when flipped to true the original unignore collisions behaviour will be skipped. Arg0: Grabber - The grabber whose collisions are ignored with 'colliders'. Arg1: GrabbableObject - The grabbable object who is associated with the collision ignore event. Arg2: Collider[] - The array of Colliders the grabber is ignoring. Arg3: ref bool - The 'skip unignore collisions' boolean reference that when made true will make the 'unignore collisions' behaviour of this component get skipped.

Properties

- GrabbableObject **Grabbable** `[get]`

  A reference to the GrabbableObject associated with this component.

5.23.1 Detailed Description

A component intended to be attached to a GrabbableObject that subscribes to the relevant 'Grabbed' and 'Released' events to ignore collisions between some colliders and a Grabber while it is holding an object.

NOTE: This component does not manage triggers. Author: Mathew Aloisio

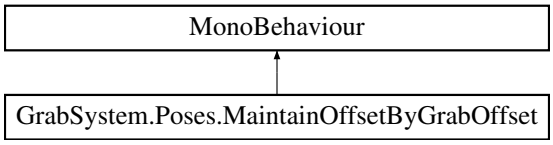
The documentation for this class was generated from the following file:

- IgnoreCollidersWhileGrabbing.cs

5.24 GrabSystem.Poses.MaintainOffsetByGrabOffset Class Reference

A component that is attached to the same GameObject as a GrabbableObject (that is assumed to be using the 'Maintain Offset' grab mode) which overrides the maintain offset behaviour allowing more complex behaviours to be defined using the grab offset of the Grabber from the grabbable at the time it grabbed the GrabbableObject. NOTE: This component works in the grabbables local space by comparing the 'signed distance in direction' of the grabber from the grabbable at the time of grab using the comparator specified in the entry.

Inheritance diagram for GrabSystem.Poses.MaintainOffsetByGrabOffset:



Classes

- class Entry
- class GrabInfo

Public Member Functions

- bool CheckEntryConditions (Entry pEntry, Grabber pGrabber, GrabbableObject pGrabbable)

  A public method that checks if the conditions are met for the given entry, pEntry, based on the relative offset of pGrabber from pGrabbable in pGrabbable local space.

Public Attributes

- Entry[ ] **entries**

  An array of MaintainOffsetByGrabOffset.Entrys that contain rules that decide which grab pose is used.

Properties

- GrabbableObject **Grabbable** `[get]`

  Returns the GrabbableObject reference associated with this component.

5.24.1 Detailed Description

A component that is attached to the same GameObject as a GrabbableObject (that is assumed to be using the 'Maintain Offset' grab mode) which overrides the maintain offset behaviour allowing more complex behaviours to be defined using the grab offset of the Grabber from the grabbable at the time it grabbed the GrabbableObject. NOTE: This component works in the grabbables local space by comparing the 'signed distance in direction' of the grabber from the grabbable at the time of grab using the comparator specified in the entry.

Author: Intuitive Gaming Solutions

5.24.2 Member Function Documentation

5.24.2.1 CheckEntryConditions()

```
bool GrabSystem.Poses.MaintainOffsetByGrabOffset.CheckEntryConditions (
            Entry pEntry,
            Grabber pGrabber,
            GrabbableObject pGrabbable )
```

A public method that checks if the conditions are met for the given entry, pEntry, based on the relative offset of pGrabber from pGrabbable in pGrabbable local space.

Parameters

pEntry	
pGrabber	
pGrabbable	

Returns

true if the conditions for this entry are met, otherwise false.

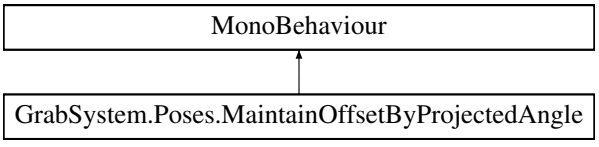
The documentation for this class was generated from the following file:

- MaintainOffsetByGrabOffset.cs

5.25 GrabSystem.Poses.MaintainOffsetByProjectedAngle Class Reference

A component that is attached to the same GameObject as a GrabbableObject (that is assumed to be using the 'Maintain Offset' grab mode) which overrides the maintain offset behaviour allowing more complex behaviours to be defined using the relative angle of the GrabbableObject from the Grabber at the time it is grabbed. NOTE: This component works by projecting the 'grabber' onto the local 'axis' in local space of the grabbable and getting the hands angle around that axis.

Inheritance diagram for GrabSystem.Poses.MaintainOffsetByProjectedAngle:



Classes

- class Entry
- class GrabInfo

Public Member Functions

- bool CheckEntryConditions (Entry pEntry, Grabber pGrabber, GrabbableObject pGrabbable)

  A public method that checks if the conditions are met for the given entry, pEntry, based on the relative offset of pGrabber from pGrabbable in pGrabbable local space.

Public Attributes

- Entry[ ] **entries**

  An array of MaintainOffsetRelativeAngle.Entrys that contain rules that decide which grab pose is used.

Properties

- GrabbableObject **Grabbable** `[get]`

  Returns the GrabbableObject reference associated with this component.

5.25.1 Detailed Description

A component that is attached to the same GameObject as a GrabbableObject (that is assumed to be using the 'Maintain Offset' grab mode) which overrides the maintain offset behaviour allowing more complex behaviours to be defined using the relative angle of the GrabbableObject from the Grabber at the time it is grabbed. NOTE: This component works by projecting the 'grabber' onto the local 'axis' in local space of the grabbable and getting the hands angle around that axis.

Author: Intuitive Gaming Solutions

5.25.2 Member Function Documentation

5.25.2.1 CheckEntryConditions()

```
bool GrabSystem.Poses.MaintainOffsetByProjectedAngle.CheckEntryConditions (
            Entry pEntry,
            Grabber pGrabber,
            GrabbableObject pGrabbable )
```

A public method that checks if the conditions are met for the given entry, pEntry, based on the relative offset of pGrabber from pGrabbable in pGrabbable local space.

Parameters

pEntry	
pGrabber	
pGrabbable	

Returns

true if the conditions for this entry are met, otherwise false.

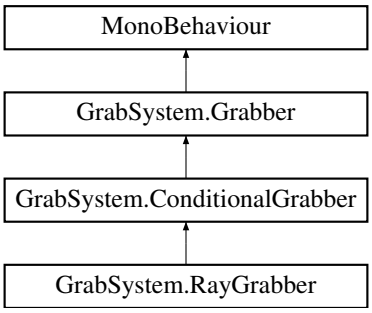
The documentation for this class was generated from the following file:

- MaintainOffsetByProjectedAngle.cs

5.26 GrabSystem.RayGrabber Class Reference

A component that extends a Grabber by adding the ability to attempt grabs using a raycast.

Inheritance diagram for GrabSystem.RayGrabber:



Classes

- class BoxCastData
- struct GrabbableEntry

Public Member Functions

- GrabbableObject GrabByPalmTrace ()

  Attempts to grab any nearby objects using a ray trace from the palm. Returns the GrabbableObject that was grabbed or null if nothing was grabbed.
- override GrabbableObject ConditionalGrab ()

  Attempts to grab using a raytrace.

Static Public Member Functions

- static void DrawBoxCastDataGizmo (BoxCastData pBoxCastData, Color pColor)

  Draws the given box cast data. Uses pBoxCastData.origin to determine position, rotation, and direction.
- static void DrawBoxCastDataGizmoAt (BoxCastData pBoxCastData, Color pColor, Vector3 pPosition, Quaternion pRotation, Vector3 pDirection)

  Draws the given box cast data at the given position and rotation in the given direction.

Public Attributes

- LayerMask **ignoreGrabLayers**

  A LayerMask of layers the grab system should ignore.
- BoxCastData **boxCastSettings** = new BoxCastData() { origin = null, boxHeight = 0.04f, boxWidth = 0.01f, boxLength = 0.01f }

  The settings that control the grab

Additional Inherited Members

5.26.1 Detailed Description

A component that extends a Grabber by adding the ability to attempt grabs using a raycast.

Author: Mathew Aloisio

5.26.2 Member Function Documentation

5.26.2.1 ConditionalGrab()

```
override GrabbableObject GrabSystem.RayGrabber.ConditionalGrab ( )  [virtual]
```

Attempts to grab using a raytrace.

Returns

The GrabbableObject that was grabbed, otherwise null.

Implements GrabSystem.ConditionalGrabber.

5.26.2.2 DrawBoxCastDataGizmo()

```
static void GrabSystem.RayGrabber.DrawBoxCastDataGizmo (
            BoxCastData pBoxCastData,
            Color pColor )  [static]
```

Draws the given box cast data. Uses pBoxCastData.origin to determine position, rotation, and direction.

Parameters

pBoxCastData	
pColor	The color to draw the gizmo in.

5.26.2.3 DrawBoxCastDataGizmoAt()

```
static void GrabSystem.RayGrabber.DrawBoxCastDataGizmoAt (
            BoxCastData pBoxCastData,
            Color pColor,
            Vector3 pPosition,
            Quaternion pRotation,
            Vector3 pDirection )  [static]
```

Draws the given box cast data at the given position and rotation in the given direction.

Parameters

pBoxCastData	
pColor	The color to draw the gizmo in.
pPosition	The position to draw the box cast at.
pRotation	The rotation to draw the box cast with.
pDirection	The box cast direction.

5.26.2.4 GrabByPalmTrace()

```
GrabbableObject GrabSystem.RayGrabber.GrabByPalmTrace ( )
```

Attempts to grab any nearby objects using a ray trace from the palm. Returns the GrabbableObject that was grabbed or null if nothing was grabbed.

Returns

A reference to the GrabbableObject that was grabbed, otherwise null if nothing.

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

- RayGrabber.cs

5.27 GrabSystem.Collisions.DelayUnignoreCollidersOnRelease.↩ UnignoreEntry Struct Reference

Public Attributes

- Grabber **grabber**

  The Grabber to unignore collisions with.
- GrabbableObject **grabbable**

  The GrabbableObject that was associated with the collision ignore event.
- float **unignoreTime**

  The Time.time to unignore the collision at.
- Collider[ ] **unignoreColliders**

  The array of Colliders to unignore.

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

- DelayUnignoreCollidersOnRelease.cs

Index