Waypoints

1.0.1

Generated by Doxygen 1.9.5

Chapter 1

Namespace Index

1.1 Package List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 Waypoints Namespace Reference

Classes

- class **WaypointsEditorUtility**

  A public static class that provides some useful helper function(s) related to things like copying and instantiating visual-only representations of GameObjects.

Enumerations

- enum **InterpolationMode**
- enum **PathMode**
- enum **RotationMode**

4.2 Waypoints.Editor Namespace Reference

4.3 Waypoints.Editor.Pathing Namespace Reference

Classes

- class **PathMoveEditorSettings**

  A public static class that stores settings for the 'Path Move Editor'.
- class PathMoveEditorSettingsWindow

  A window where path move gizmo settings can be modified.
- class PathMoveEditorWindow

  A tool designed to make it even easier to edit waypoints.
- class **PathMoveGizmo**

  Draws editor gizmos for PathMoves.
- class RigidbodyPathMoveEditor

  A custom Editor for RigidbodyPathMove components.
- class TransformPathMoveEditor

  A custom Editor for TransformPathMove components.

4.4 Waypoints.Mathematics Namespace Reference

Classes

- class **MathUtility**

  A public static class that provides helper math-related functions.

4.5 Waypoints.Pathing Namespace Reference

Classes

- class BSpline
- class CatmullRom
- class IInterpolator
- class InvokeEventIfPathMoveProgressionIs

  A component that invokes an event when when InvokeIf is called only if the referenced PathMove's pathing progression field compared to the 'triggerValue' using the set operation yields a true result.
- class InvokeEventOnWaypointReachedByPathMove

  A component that invokes an event when a waypoint is reached by a pathing PathMove. NOTE: This component will only be triggered if it is enabled.
- class Linear
- class PathMove

  An abstract class that is the base class for all path movers that use the pathing library.
- class PathMovePauser

  A component that provides an easy way to pause a PathMove component for time.
- class RigidbodyPathMove

  A component for moving a rigidbody along waypoints.
- class TransformPathMove

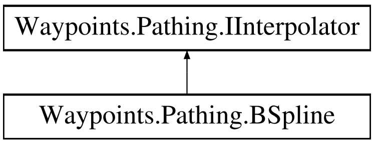
  A component for moving a Transform along waypoints.

Chapter 5

Class Documentation

5.1 Waypoints.Pathing.BSpline Class Reference

Inheritance diagram for Waypoints.Pathing.BSpline:



Public Member Functions

- **BSpline** (Vector3[ ] pPoints, bool pClosed=false)
- **BSpline** (Transform[ ] pPoints, bool pClosed=false)
- **BSpline** (List< Transform > pPoints, bool pClosed=false)
- override Vector3 Evaluate (float pU)
- override Vector3 Heading (float pU)

Additional Inherited Members

5.1.1 Member Function Documentation

5.1.1.1 Evaluate()

```
override Vector3 Waypoints.Pathing.BSpline.Evaluate (
             float pU )  [virtual]
```

Implements Waypoints.Pathing.IInterpolator.

5.1.1.2 Heading()

```
override Vector3 Waypoints.Pathing.BSpline.Heading (
            float pU )  [virtual]
```

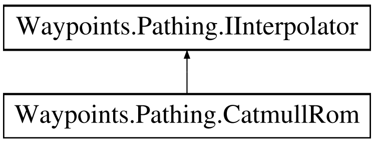
Implements Waypoints.Pathing.IInterpolator.

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

- BSpline.cs

5.2 Waypoints.Pathing.CatmullRom Class Reference

Inheritance diagram for Waypoints.Pathing.CatmullRom:



Public Member Functions

- **CatmullRom** (Vector3[ ] pPoints, bool pClosed)
- **CatmullRom** (Transform[ ] pPoints, bool pClosed)
- **CatmullRom** (List< Transform > pPoints, bool pClosed=false)
- override Vector3 Evaluate (float pU)
- override Vector3 Heading (float pU)

Additional Inherited Members

5.2.1 Member Function Documentation

5.2.1.1 Evaluate()

```
override Vector3 Waypoints.Pathing.CatmullRom.Evaluate (
            float pU )  [virtual]
```

Implements Waypoints.Pathing.IInterpolator.

5.2.1.2 Heading()

```
override Vector3 Waypoints.Pathing.CatmullRom.Heading (
            float pU )  [virtual]
```

Implements Waypoints.Pathing.IInterpolator.

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

- CatmullRom.cs

5.3 Waypoints.Pathing.RigidbodyPathMove.FreezeAxis Struct Reference

Public Attributes

- bool **x**
- bool **y**
- bool **z**

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

- RigidbodyPathMove.cs

5.4 Waypoints.Pathing.TransformPathMove.FreezeAxis Struct Reference

Public Attributes

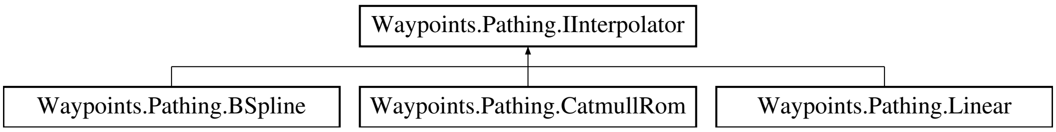
- bool **x**
- bool **y**
- bool **z**

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

- TransformPathMove.cs

5.5 Waypoints.Pathing.IInterpolator Class Reference

Inheritance diagram for Waypoints.Pathing.IInterpolator:


Waypoints.Pathing.IInterpolator
Waypoints.Pathing.BSpline
Waypoints.Pathing.CatmullRom
Waypoints.Pathing.Linear

Public Member Functions

- **IInterpolator** (Vector3[ ] pPoints, bool pClosed)
- **IInterpolator** (Transform[ ] pPoints, bool pClosed)
- **IInterpolator** (List< Transform > pPoints, bool pClosed)
- void **SetControl** (Vector3[ ] pPoints)
- void **SetControl** (Transform[ ] pPoints)
- void **SetControl** (List< Transform > pPoints)
- void **SetClosed** (bool pClosed)
- int **Limit** (int pIndex)
- abstract Vector3 **Evaluate** (float pU)
- abstract Vector3 **Heading** (float pU)

Public Attributes

- Vector3[ ] **control**

  An array of control point positions.
- bool **closed**

  Is the control points array a closed loop?
- int **length**

  The number of control points.

Protected Attributes

- int **m_LengthM1**

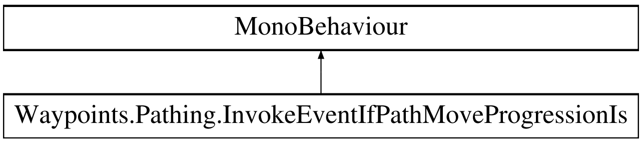
The documentation for this class was generated from the following file:

- IInterpolator.cs

5.6 Waypoints.Pathing.InvokeEventIfPathMoveProgressionIs Class Reference

A component that invokes an event when when InvokeIf is called only if the referenced PathMove's pathing progression field compared to the 'triggerValue' using the set operation yields a true result.

Inheritance diagram for Waypoints.Pathing.InvokeEventIfPathMoveProgressionIs:



Public Types

- enum **Condition**

Public Member Functions

- void **InvokeIf** ()

  A public method that invokes the 'Triggered' event if it is called when the referenced PathMove's pathing progression field compared to the 'triggerValue' using the set operation yields a true result.

Public Attributes

- Condition **compareCondition**

  The condition to use when comparing the referenced PathMove
- float **triggerValue**

  The value that the referenced PathMOve
- UnityEvent **Triggered**

  A unity event that is invoked when InvokeIf

5.6.1 Detailed Description

A component that invokes an event when when InvokeIf is called only if the referenced PathMove's pathing progression field compared to the 'triggerValue' using the set operation yields a true result.

Author: Mathew Aloisio

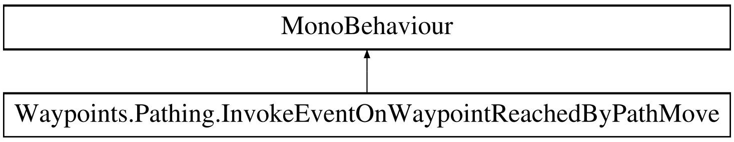
The documentation for this class was generated from the following file:

- InvokeEventIfPathMoveProgressionIs.cs

5.7 Waypoints.Pathing.InvokeEventOnWaypointReachedByPathMove Class Reference

A component that invokes an event when a waypoint is reached by a pathing PathMove. NOTE: This component will only be triggered if it is enabled.

Inheritance diagram for Waypoints.Pathing.InvokeEventOnWaypointReachedByPathMove:



Public Member Functions

- void NotifyWaypointReached (PathMove pPathMove)

  A public method that is used to notify this component when the associated waypoint is reached.

Public Attributes

- UnityEvent **WaypointReached**

  A unity event that is invoked when the waypoint this component is attached to is reached by any BaseAI.

5.7.1 Detailed Description

A component that invokes an event when a waypoint is reached by a pathing PathMove. NOTE: This component will only be triggered if it is enabled.

Author: Mathew Aloisio

5.7.2 Member Function Documentation

5.7.2.1 NotifyWaypointReached()

```
void Waypoints.Pathing.InvokeEventOnWaypointReachedByPathMove.NotifyWaypointReached (
            PathMove pPathMove )
```

A public method that is used to notify this component when the associated waypoint is reached.

Parameters

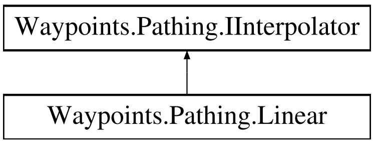
pPathMove	

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

- InvokeEventOnWaypointReachedByPathMove.cs

5.8 Waypoints.Pathing.Linear Class Reference

Inheritance diagram for Waypoints.Pathing.Linear:



Public Member Functions

- **Linear** (Vector3[ ] pPoints, bool pClosed=false)
- **Linear** (Transform[ ] pPoints, bool pClosed=false)
- **Linear** (List< Transform > pPoints, bool pClosed=false)
- override Vector3 Evaluate (float pU)
- override Vector3 Heading (float pU)

Additional Inherited Members

5.8.1 Member Function Documentation

5.8.1.1 Evaluate()

```
override Vector3 Waypoints.Pathing.Linear.Evaluate (
            float pU )  [virtual]
```

Implements Waypoints.Pathing.IInterpolator.

5.8.1.2 Heading()

```
override Vector3 Waypoints.Pathing.Linear.Heading (
            float pU )  [virtual]
```

Implements Waypoints.Pathing.IInterpolator.

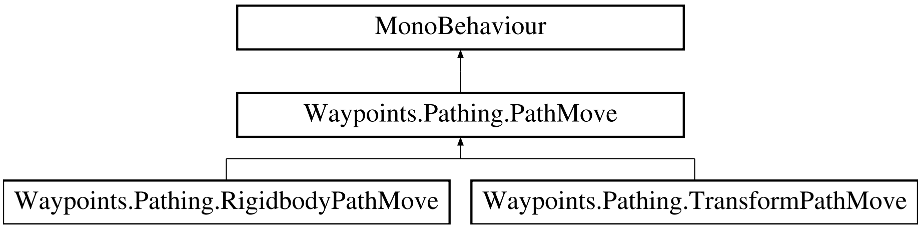
The documentation for this class was generated from the following file:

- Linear.cs

5.9 Waypoints.Pathing.PathMove Class Reference

An abstract class that is the base class for all path movers that use the pathing library.

Inheritance diagram for Waypoints.Pathing.PathMove:


MonoBehaviour
Waypoints.Pathing.PathMove
Waypoints.Pathing.RigidbodyPathMove
Waypoints.Pathing.TransformPathMove

Classes

- class WaypointUnityEvent

Public Member Functions

- void **InitializeInterpolator** ()

  (Re)initializes the interpolator for this PathMove.
- void SetPathingPaused (bool pPaused)

  Controls whether or not Pathing is paused for a PathMove component.
- void SetPathingProgression (float pProgression)

  Allows the PathMove's pathing progression variable to be set. NOTE: pProgression is clamped between 0f and (float)waypoints.length
- void AddWaypoint (Transform pWaypoint)

  Adds a waypoint at the end of the waypoints array. NOTE: This method only adds a waypoint by reference it does not instantiate any object.
- void AddWaypointAtIndex (int pIndex, Transform pWaypoint)

  Adds a waypoint at the given index in the waypoints array, shifting array elements and expanding the array as neccesary. NOTE: pIndex can not be greater than WaypointCount but no error checking is performed. NOTE: When pIndex == WaypointCount this adds a waypoint at the end of the array and shifts no elements. NOTE: This method only adds a waypoint at the specified index by reference and shuffles the waypoints array, it does not instantiate any object.
- void SetWaypointByIndex (int pIndex, Transform pWaypoint)

  Sets the waypoint Transform reference at the given index, pIndex, in the waypoints array to pWaypoint.
- void RemoveWaypointByIndex (int pIndex)

  Removes the waypoint at the given index in the waypoints array, shifting elements above it down in the array. NOTE: This method DOES NOT destroy the waypoint object, it simply removes it from the waypoints array.
- Transform GetWaypointByIndex (int pIndex)

  Returns the Transform of the waypoint at the given index, pIndex, in the waypoints array. NOTE: This method performs no error checking.
- abstract void ForceMoveToWaypoint (int pPointIndex)

  Force the position of the transform of the PathMove to the position of the waypoint in the index. This only forces the position and will instantly be overridden by an active path move.
- virtual bool IsInterpolatorValid ()

  Returns true if the PathMove's interpolator is valid and matching the waypoint settings, otherwise false.
- virtual void **OnWaypointReached** (int pPointIndex)

  A callback that is intended to be invoked by a PathMove when it reaches a waypoint.
- virtual void TeleportToWaypoint (int pIndex)

  Allows the PathMove to be teleported directly to a waypoint instantly.

Public Attributes

- bool **pauseOnAwake**

  Should the PathMove start paused when Awake
- PathMode **pathMode**

  The mode that determines the order in which waypoints should be followed.
  \nClosed
- RotationMode **rotationMode** = RotationMode.ForwardToPoint

  The rotation mode to use when rotating the transform.
- InterpolationMode **interpolationMode**

  The interpolation method to use.
- WaypointUnityEvent **WaypointReached**

  An event that is fired when a waypoint is reached.
- UnityEvent **PathingPaused**

  An event that is invoked when pathing is paused for this PathMove.
- UnityEvent **PathingResumed**

  An event that is invoked when pathing is resumed for this PathMove.

Protected Member Functions

- virtual void **Awake** ()
- virtual void **OnDrawGizmos** ()
- virtual void **OnDrawGizmosSelected** ()
- void **Notify_WaypointReached** (int pIndex)

  Checks if a waypoint has been reached. This public method can be used in things like the teleport system to update the checkpoint reached status after teleporting.

Protected Attributes

- float **m_U**

  The variable that controls how the waypoint the PathMove is on and far along we are between waypoints.
- int **m_LastUFire** = -1

  The last whole number 'U' value that was fired, aka the last waypoint that was reached.

Properties

- int **WaypointCount** `[get]`

  Returns the number of waypoints this path move has configured.
- float **PathingProgression** `[get, set]`

  An accessor that allows the pathing progression (m_U) variable of a PathMove component to be both read and modified.
- IInterpolator **Interpolator** `[get, protected set]`

  Returns the IInterpolator associated with this component.
- InterpolationMode InterpolatorType `[get, protected set]`

  A reference to the InterpolationMode used to create the 'Interpolator'. Only consider this valid when the 'Interpolator' is non-null.
- bool **IsPathingPaused** `[get]`
- bool **ShouldDrawGizmo** `[get, set]`

  Only relevant in the editor, this is set to true every OnDrawGizmos() call... the gizmo drawer sets it to false, this allows gizmos to still be toggled using the editor dropdown.

5.9.1 Detailed Description

An abstract class that is the base class for all path movers that use the pathing library.

Only one PathMove is allowed per GameObject. Author: Mathew Aloisio

5.9.2 Member Function Documentation

5.9.2.1 AddWaypoint()

```
void Waypoints.Pathing.PathMove.AddWaypoint (
            Transform pWaypoint )
```

Adds a waypoint at the end of the waypoints array. NOTE: This method only adds a waypoint by reference it does not instantiate any object.

Parameters

pWaypoint	The waypoint to add.

5.9.2.2 AddWaypointAtIndex()

```
void Waypoints.Pathing.PathMove.AddWaypointAtIndex (
             int pIndex,
             Transform pWaypoint )
```

Adds a waypoint at the given index in the waypoints array, shifting array elements and expanding the array as neccesary. NOTE: pIndex can not be greater than WaypointCount but no error checking is performed. NOTE: When pIndex == WaypointCount this adds a waypoint at the end of the array and shifts no elements. NOTE: This method only adds a waypoint at the specified index by reference and shuffles the waypoints array, it does not instantiate any object.

Parameters

pIndex	
pWaypoint	The waypoint to add at pIndex.

5.9.2.3 ForceMoveToWaypoint()

```
abstract void Waypoints.Pathing.PathMove.ForceMoveToWaypoint (
             int pPointIndex )  [pure virtual]
```

Force the position of the transform of the PathMove to the position of the waypoint in the index. This only forces the position and will instantly be overridden by an active path move.

NOTE: This WILL NOT invoke any waypoint reached events.

Parameters

pPointIndex	

Implemented in Waypoints.Pathing.RigidbodyPathMove, and Waypoints.Pathing.TransformPathMove.

5.9.2.4 GetWaypointByIndex()

```
Transform Waypoints.Pathing.PathMove.GetWaypointByIndex (
             int pIndex )
```

Returns the Transform of the waypoint at the given index, pIndex, in the waypoints array. NOTE: This method performs no error checking.

Parameters

pIndex	

Returns

the Transform of the waypoint at the given index, pIndex, in the waypoints array.

5.9.2.5 IsInterpolatorValid()

```
virtual bool Waypoints.Pathing.PathMove.IsInterpolatorValid ( )  [virtual]
```

Returns true if the PathMove's interpolator is valid and matching the waypoint settings, otherwise false.

Returns

true if the PathMove's interpolator is valid and matching the waypoint settings, otherwise false.

Reimplemented in Waypoints.Pathing.RigidbodyPathMove, and Waypoints.Pathing.TransformPathMove.

5.9.2.6 RemoveWaypointByIndex()

```
void Waypoints.Pathing.PathMove.RemoveWaypointByIndex (
             int pIndex )
```

Removes the waypoint at the given index in the waypoints array, shifting elements above it down in the array. NOTE: This method DOES NOT destroy the waypoint object, it simply removes it from the waypoints array.

Parameters

pIndex	

5.9.2.7 SetPathingPaused()

```
void Waypoints.Pathing.PathMove.SetPathingPaused (
             bool pPaused )
```

Controls whether or not Pathing is paused for a PathMove component.

Parameters

pPaused	

5.9.2.8 SetPathingProgression()

```
void Waypoints.Pathing.PathMove.SetPathingProgression (
             float pProgression )
```

Allows the PathMove's pathing progression variable to be set. NOTE: pProgression is clamped between 0f and (float)waypoints.length

Parameters

pProgression	

5.9.2.9 SetWaypointByIndex()

```
void Waypoints.Pathing.PathMove.SetWaypointByIndex (
             int pIndex,
             Transform pWaypoint )
```

Sets the waypoint Transform reference at the given index, pIndex, in the waypoints array to pWaypoint.

Parameters

pIndex	
pWaypoint	

5.9.2.10 TeleportToWaypoint()

```
virtual void Waypoints.Pathing.PathMove.TeleportToWaypoint (
             int pIndex )  [virtual]
```

Allows the PathMove to be teleported directly to a waypoint instantly.

Parameters

pIndex	

Reimplemented in Waypoints.Pathing.RigidbodyPathMove, and Waypoints.Pathing.TransformPathMove.

5.9.3 Property Documentation

5.9.3.1 InterpolatorType

```
InterpolationMode Waypoints.Pathing.PathMove.InterpolatorType  [get], [protected set]
```

A reference to the InterpolationMode used to create the 'Interpolator'. Only consider this valid when the 'Interpolator' is non-null.

summary>Is the PathMove's pathing paused? (NOTE: This pauses only translations, not rotations.)

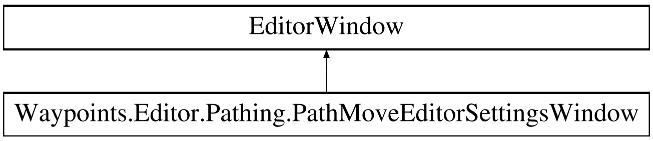
The documentation for this class was generated from the following file:

- PathMove.cs

5.10 Waypoints.Editor.Pathing.PathMoveEditorSettingsWindow Class Reference

A window where path move gizmo settings can be modified.

Inheritance diagram for Waypoints.Editor.Pathing.PathMoveEditorSettingsWindow:



Public Member Functions

- void **OverwriteGlobalSettings** ()

  Overwrites the PathMoveGizmo and PathMoveEditorSettings static classes' settings with the ones from this PathMoveEditorSettingsWindow.
- void **OverwriteLocalSettings** ()

  Overwrites the editor window's settings with the relevant static classes' settings.
- void **ResetGlobalSettings** ()

  Resets and overwrites the PathMoveGizmo and PathMoveEditorSettings static classes' settings with the ones from this PathMoveEditorSettingsWindow.

Public Attributes

- bool **autoRenameWaypoints** = PathMoveEditorSettings.Global.autoRenameWaypoints

  Should waypoints be automatically renamed after adding or removing waypoints using the editor?
- Material **autoGenWaypointMaterial** = PathMoveEditorSettings.Global.autoGenWaypointMaterial

  The Material to use for auto-generated waypoints.
- float **waypointSphereRadius** = PathMoveGizmo.Settings.waypointSphereRadius

  The radius used to render PathMove Waypoint sphere gizmos.
- Color **waypointSphereColor** = PathMoveGizmo.Settings.waypointSphereColor

  The color used to render PathMove Waypoint sphere gizmos.
- Color **highlightedWaypointSphereColor** = PathMoveGizmo.Settings.highlightedWaypointSphereColor

  The color used to render highlighted PathMove Waypoint sphere gizmos.

- int **waypointLabelFontSize** = PathMoveGizmo.Settings.waypointLabelFontSize

  The font size used to render PathMove Waypoint label gizmos.
- Color **waypointLabelFontColor** = PathMoveGizmo.Settings.waypointLabelFontColor

  The font color used to render PathMove Waypoint label gizmos.
- Color **highlightedWaypointLabelFontColor** = PathMoveGizmo.Settings.highlightedWaypointLabelFont←Color

  The font color used to render highlighted PathMove Waypoint label gizmos.
- int **pathDotsPerUnit** = PathMoveGizmo.Settings.pathDotsPerUnit

  The number of path visualization dots per distance unit.
- float **pathDotSphereGizmoRadius** = PathMoveGizmo.Settings.pathDotSphereGizmoRadius

  The radius of the path visualization dot sphere gizmos.
- Color **pathDotSphereStartColor** = PathMoveGizmo.Settings.pathDotSphereStartColor

  The color of the path visualization dot sphere gizmos at the start of a path.
- Color **pathDotSphereEndColor** = PathMoveGizmo.Settings.pathDotSphereEndColor

  The color of the path visualization dot sphere gizmos at the end of a path.
- Color **closedPathDotSphereGizmoColor** = PathMoveGizmo.Settings.closedPathDotSphereGizmoColor

  The color of the path visualization dot sphere gizmos that mark a closed pathing loop.

Static Public Attributes

- const int **MINIMUM_FONT_SIZE** = 1

  The minimum font size for gizmo labels.
- const int **MAXIMUM_FONT_SIZE** = 64

  The maximum font size for gizmo labels.
- const int **MINIMUM_DOTS_PER_UNIT** = 0

  The minimum number of path visualization dots per unit.
- const int **MAXIMUM_DOTS_PER_UNIT** = 1024

  The maximum number of path visualization dots per unit.

Events

- static Action< PathMoveEditorSettingsWindow > **Initialized**

  A C# delegate event that is invoked when the PathMoveEditorSettingsWindow is intialized.

5.10.1 Detailed Description

A window where path move gizmo settings can be modified.

Author: Mathew Aloisio

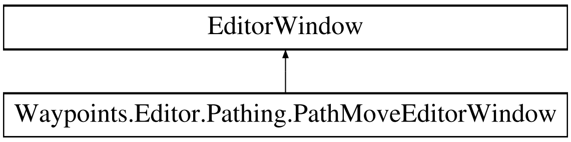
The documentation for this class was generated from the following file:

- PathMoveEditorSettingsWindow.cs

5.11 Waypoints.Editor.Pathing.PathMoveEditorWindow Class Reference

A tool designed to make it even easier to edit waypoints.

Inheritance diagram for Waypoints.Editor.Pathing.PathMoveEditorWindow:



Public Types

- enum **WaypointType**
- enum **WaypointSpawnPoint**

Public Member Functions

- void AddWaypoint (bool pSelectWaypoint)

  Adds a waypoint at the end of 'EditingPathMove.waypoints' array.
- void AddWaypointAtIndex (int pIndex, bool pSelectWaypoint)

  Adds a waypoint at the given index in the 'EditingPathMove.waypoints' array, shifting array elements and expanding the array as neccesary. NOTE: pIndex can not be greater than EditingPathMove.WaypointCount. NOTE: When pIndex == EditingPathMove.WaypointCount this adds a waypoint at the end of the array and shifts no elements.
- void RemoveWaypointByIndex (int pIndex, bool pSelectWaypoint)

  Removes the waypoint at the given index in the 'EditingPathMove.waypoints' array, shifting elements above it down in the array.
- void **SelectPreviousWaypoint** ()

  Selects the previous waypoint in 'EditingPathMove.waypoints' before EditingWaypointIndex. NOTE: Does nothing if 'EditingPathMove' is null, or if the waypoints array is null or empty. NOTE: This method automatically wraps the selection.
- void **SelectNextWaypoint** ()

  Selects the next waypoint in 'EditingPathMove.waypoints' after EditingWaypointIndex. NOTE: Does nothing if 'EditingPathMove' is null, or if the waypoints array is null or empty. NOTE: This method automatically wraps the selection.
- void **ValidateContainerComponents** ()

  Ensures the 'waypointContainer' contains the components specifeid in the editor. If they are missing they are added. Registers undo group upon adding component.
- void **ValidateWaypointComponents** ()

  Ensures all waypoints of 'EditingPathMove' contain the components specified in the editor. If they are missing they are added. Registers undo group upon adding component.
- void **RenameWaypoints** ()

  Renames all waypoints in 'EditingPathMove' by number while also registering an undo group.
- GameObject InstantiateWaypoint (int pIndex)

  Instantiates a waypoint based on the current settings and returns the Transform, or null if failed.

Static Public Member Functions

- static void **Open** ()

Public Attributes

- string **waypointName** = DEFAULT_WAYPOINT_NAME

  The name to give to waypoints created using the editor.
- Transform **waypointContainer**

  A reference to a Transform to store created waypoints in.
- ClassTypeReference[ ] **containerComponents**

  An array of MonoBehaviours that will automatically be added to the
- ClassTypeReference[ ] **waypointComponents**

  An array of MonoBehaviours that will automatically be added to waypoints created with the editor if not already found on the waypoint object that was instantiated.
- WaypointSpawnPoint **waypointSpawn**

  Controls how waypoints are spawned. Mover - Spawn the waypoint positioned and rotated to match the mover. PreviousWaypoint - Spawn the waypoint positioned and rotated to match the previous waypoint if possible, otherwise falls back to mover.
- WaypointType **waypointType**

  The type of waypoint to create. AutoGenerate - Automatically generate visual waypoint. Prefab - Use prefab to place visual waypoint. Empty - A waypoint with no visual.
- GameObject **waypointPrefab**

  A reference to the prefab that is used when making 'Prefab' waypoints.

Static Public Attributes

- const string **DEFAULT_WAYPOINT_NAME** = "Waypoint"

  The default name for waypoints.

Properties

- PathMove **EditingPathMove** `[get]`

  A reference to the PathMove currently being edited, or null.
- int **EditingWaypointIndex** `[get, set]`

  Returns the index of the waypoint currently being edited in the EditingPathMove.waypoints array, or -1.
- Transform **EditingWaypoint** `[get]`

  Returns the Transform for the waypoint currently being edited, or null.

Events

- static Action< PathMoveEditorWindow > **Initialized**

  An event that is invoked when the PathMoveEditorWindow is intialized.

5.11.1 Detailed Description

A tool designed to make it even easier to edit waypoints.

Author: Mathew Aloisio

5.11.2 Member Function Documentation

5.11.2.1 AddWaypoint()

```
void Waypoints.Editor.Pathing.PathMoveEditorWindow.AddWaypoint (
            bool pSelectWaypoint )
```

Adds a waypoint at the end of 'EditingPathMove.waypoints' array.

Parameters

pSelectWaypoint	Make the newly added waypoint the active selection?

5.11.2.2 AddWaypointAtIndex()

```
void Waypoints.Editor.Pathing.PathMoveEditorWindow.AddWaypointAtIndex (
            int pIndex,
            bool pSelectWaypoint )
```

Adds a waypoint at the given index in the 'EditingPathMove.waypoints' array, shifting array elements and expanding the array as neccesary. NOTE: pIndex can not be greater than EditingPathMove.WaypointCount. NOTE: When pIndex == EditingPathMove.WaypointCount this adds a waypoint at the end of the array and shifts no elements.

Parameters

pIndex	
pSelectWaypoint	Make the newly added waypoint the active selection?

5.11.2.3 InstantiateWaypoint()

```
GameObject Waypoints.Editor.Pathing.PathMoveEditorWindow.InstantiateWaypoint (
            int pIndex )
```

Instantiates a waypoint based on the current settings and returns the Transform, or null if failed.

Parameters

pIndex	The index the instantiated waypoint will be in.

Returns

The Tranasform of the instantiated waypoint, or null.

5.11.2.4 RemoveWaypointByIndex()

```
void Waypoints.Editor.Pathing.PathMoveEditorWindow.RemoveWaypointByIndex (
            int pIndex,
            bool pSelectWaypoint )
```

Removes the waypoint at the given index in the 'EditingPathMove.waypoints' array, shifting elements above it down in the array.

Parameters

pIndex	
pSelectWaypoint	Automatically make the previous waypoint (if valid) or next waypoint (if valid) the active selection?

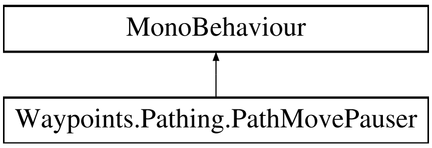
The documentation for this class was generated from the following file:

- PathMoveEditorWindow.cs

5.12 Waypoints.Pathing.PathMovePauser Class Reference

A component that provides an easy way to pause a PathMove component for time.

Inheritance diagram for Waypoints.Pathing.PathMovePauser:



Public Member Functions

- void **Pause** ()

  Pauses the referenced PathMove til it's manually unpaused.
- void PauseFor (float pSeconds)

  Pauses the referenced PathMove for a given number of seconds.
- void **Unpause** ()

  Unpauses the referenced PathMove.

Public Attributes

- PathMove **pathMove**

  A reference to the

5.12.1 Detailed Description

A component that provides an easy way to pause a PathMove component for time.

Author: Mathew Aloisio

5.12.2 Member Function Documentation

5.12.2.1 PauseFor()

```
void Waypoints.Pathing.PathMovePauser.PauseFor (
             float pSeconds )
```

Pauses the referenced PathMove for a given number of seconds.

Parameters

pSeconds	

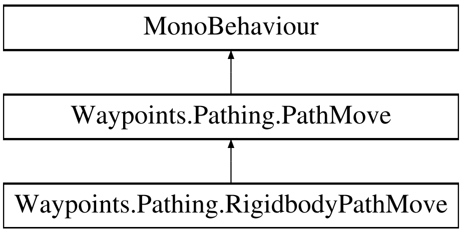
The documentation for this class was generated from the following file:

- PathMovePauser.cs

5.13 Waypoints.Pathing.RigidbodyPathMove Class Reference

A component for moving a rigidbody along waypoints.

Inheritance diagram for Waypoints.Pathing.RigidbodyPathMove:


MonoBehaviour
Waypoints.Pathing.PathMove
Waypoints.Pathing.RigidbodyPathMove

Classes

- struct FreezeAxis
- class Vector3UnityEvent

Public Member Functions

- override bool IsInterpolatorValid ()

  Returns true if the RigidbodyPathMove's interpolator is valid and matching the waypoint settings, otherwise false.
- override void ForceMoveToWaypoint (int pPointIndex)

  Force the position of the transform of the PathMove to the position of the waypoint in the index. This only forces the position and will instantly be overridden by an active path move.
- override void TeleportToWaypoint (int pIndex)

  Allows the RigidbodyPathMove to be teleported directly to a waypoint instantly.
- void **UseForwardToPointRotation** ()

  Sets the rotation mode to 'forward to point'.
- void **UseCopyPointRotation** ()

  Sets the rotation mode to 'copy waypoint'.
- void **UseInterpolateWaypointRotation** ()

  Sets the rotation mode to 'interpolate waypoint'.
- void SetMoveSpeed (float pSpeed)

  Sets the movement speed of the RigidbodyPathMove in meters per second.
- void SetSpeedMultiplier (float pMultiplier)

  Sets the speedMutliplier field of this component. Useful for use with Unity editor events.
- void SetRotationSpeed (float pDegreesPerSecond)

  Sets the rotation speed of the RigidbodyPathMove in degrees per second.

Public Attributes

- float **speed**

  The speed at which the body will move along the waypoints in meters per second in meters per second.
- float **speedMultiplier** = 1f

  The multiplier to use for current speed.
- float **rotationSpeed**

  Rotation speed in degrees/second.
- FreezeAxis **freezePosition**

  Allows the position to be frozen on a specific axis.
- FreezeAxis **freezeRotation**

  Allows the rotation to be frozen on a specific axis.
- Vector3UnityEvent **Moved**

  An event that is invoked when the RigidbodyPathMove moves.

Protected Attributes

- Rigidbody **m_Rigidbody**
- float **m_PathingDirectionMultiplier** = 1f
- Vector3 **m_StartPosition**
- Quaternion **m_StartRotation**

Additional Inherited Members

5.13.1 Detailed Description

A component for moving a rigidbody along waypoints.

Author: Mathew Aloisio

5.13.2 Member Function Documentation

5.13.2.1 ForceMoveToWaypoint()

```
override void Waypoints.Pathing.RigidbodyPathMove.ForceMoveToWaypoint (
            int pPointIndex )  [virtual]
```

Force the position of the transform of the PathMove to the position of the waypoint in the index. This only forces the position and will instantly be overridden by an active path move.

NOTE: This WILL NOT invoke any waypoint reached events.

Parameters

pPointIndex	

Implements Waypoints.Pathing.PathMove.

5.13.2.2 IsInterpolatorValid()

```
override bool Waypoints.Pathing.RigidbodyPathMove.IsInterpolatorValid ( )  [virtual]
```

Returns true if the RigidbodyPathMove's interpolator is valid and matching the waypoint settings, otherwise false.

Returns

true if the RigidbodyPathMove's interpolator is valid and matching the waypoint settings, otherwise false.

Reimplemented from Waypoints.Pathing.PathMove.

5.13.2.3 SetMoveSpeed()

```
void Waypoints.Pathing.RigidbodyPathMove.SetMoveSpeed (
             float pSpeed )
```

Sets the movement speed of the RigidbodyPathMove in meters per second.

Parameters

pSpeed	

5.13.2.4 SetRotationSpeed()

```
void Waypoints.Pathing.RigidbodyPathMove.SetRotationSpeed (
             float pDegreesPerSecond )
```

Sets the rotation speed of the RigidbodyPathMove in degrees per second.

Parameters

pDegreesPerSecond	

5.13.2.5 SetSpeedMultiplier()

```
void Waypoints.Pathing.RigidbodyPathMove.SetSpeedMultiplier (
             float pMultiplier )
```

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

Parameters

pMultiplier	

5.13.2.6 TeleportToWaypoint()

```
override void Waypoints.Pathing.RigidbodyPathMove.TeleportToWaypoint (
             int pIndex )  [virtual]
```

Allows the RigidbodyPathMove to be teleported directly to a waypoint instantly.

Parameters

pIndex	

Reimplemented from Waypoints.Pathing.PathMove.

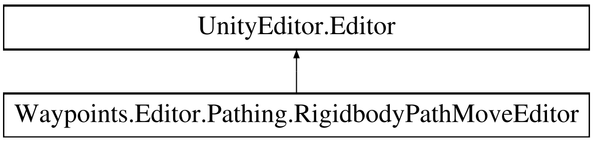
The documentation for this class was generated from the following file:

- RigidbodyPathMove.cs

5.14 Waypoints.Editor.Pathing.RigidbodyPathMoveEditor Class Reference

A custom Editor for RigidbodyPathMove components.

Inheritance diagram for Waypoints.Editor.Pathing.RigidbodyPathMoveEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

5.14.1 Detailed Description

A custom Editor for RigidbodyPathMove components.

Author: Mathew Aloisio

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

- RigidbodyPathMoveEditor.cs

5.15 Waypoints.Editor.Pathing.PathMoveEditorSettings.SettingsData Class Reference

Defines the settings for the PathMoveEditorSettings static class.

Public Attributes

- bool **autoRenameWaypoints** = false

  Should waypoints be automatically renamed after adding or removing waypoints using the editor?
- Material **autoGenWaypointMaterial**

  The material to use for automatically generated waypoints..

5.15.1 Detailed Description

Defines the settings for the PathMoveEditorSettings static class.

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

- PathMoveEditorSettings.cs

5.16 Waypoints.Editor.Pathing.PathMoveGizmo.SettingsData Class Reference

Defines the settings for the PathMoveEditorSettings static class.

Public Attributes

- float **waypointSphereRadius** = 0.05f

  The radius used to render PathMove Waypoint sphere gizmos.
- Color **waypointSphereColor** = new Color(1, 1, 1, 0.2f)

  The color used to render PathMove Waypoint sphere gizmos.
- Color **highlightedWaypointSphereColor** = new Color(0, 1, 0, 0.2f)

  The color used to render highlighted PathMove Waypoint sphere gizmos.
- int **waypointLabelFontSize** = 12

  The font size used to render PathMove Waypoint label gizmos.
- Color **waypointLabelFontColor** = Color.black

  The font color used to render PathMove Waypoint label gizmos.
- Color **highlightedWaypointLabelFontColor** = Color.white

  The font color used to render highlighted PathMove Waypoint label gizmos.
- int **pathDotsPerUnit** = 6

  How many dots should their be per distance unit? A value of 0 means no path visualization.
- float **pathDotSphereGizmoRadius** = 0.025f

  The radius of path visualization sphere gizmo dots.
- Color **pathDotSphereStartColor** = Color.blue

  The color of path visualization sphere gizmo dots at the start of a path.
- Color **pathDotSphereEndColor** = Color.cyan

  The color of path visualization sphere gizmo dots at the end of a path.
- Color **closedPathDotSphereGizmoColor** = Color.red

  The color of closed loop path visualization sphere gizmo dots.

5.16.1 Detailed Description

Defines the settings for the PathMoveEditorSettings static class.

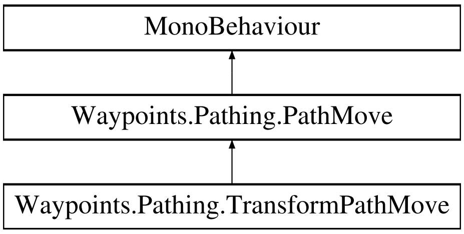
The documentation for this class was generated from the following file:

- PathMoveGizmo.cs

5.17 Waypoints.Pathing.TransformPathMove Class Reference

A component for moving a Transform along waypoints.

Inheritance diagram for Waypoints.Pathing.TransformPathMove:


MonoBehaviour
Waypoints.Pathing.PathMove
Waypoints.Pathing.TransformPathMove

Classes

- struct FreezeAxis
- class Vector3UnityEvent

Public Member Functions

- override bool IsInterpolatorValid ()

  Returns true if the TransformPathMove's interpolator is valid and matching the waypoint settings, otherwise false.
- override void ForceMoveToWaypoint (int pPointIndex)

  Force the position of the transform of the PathMove to the position of the waypoint in the index. This only forces the position and will instantly be overridden by an active path move.
- override void TeleportToWaypoint (int pIndex)

  Allows the TransformPathMove to be teleported directly to a waypoint instantly.
- void **UseForwardToPointRotation** ()

  Sets the rotation mode to 'forward to point'.
- void **UseCopyPointRotation** ()

  Sets the rotation mode to 'copy waypoint'.
- void **UseInterpolateWaypointRotation** ()

  Sets the rotation mode to 'interpolate waypoint'.
- void SetMoveSpeed (float pSpeed)

  Sets the movement speed of the TransformPathMove in meters per second.
- void SetSpeedMultiplier (float pMultiplier)

  Sets the speedMutliplier field of this component. Useful for use with Unity editor events.
- void SetRotationSpeed (float pDegreesPerSecond)

  Sets the rotation speed of the TransformPathMove in degrees per second.

Public Attributes

- Transform **moveTransform**
- float **speed**

  The speed at which the transform will move along the waypoints in meters per second.
- float **speedMultiplier** = 1f

  The multiplier to use for current speed.
- float **rotationSpeed**

  Rotation speed in degrees/second.
- FreezeAxis **freezePosition**

  Allows the position to be frozen on a specific axis.
- FreezeAxis **freezeRotation**

  Allows the rotation to be frozen on a specific axis.
- Vector3UnityEvent **Moved**

  An event that is invoked when the TransformPathMove moves.

Protected Attributes

- float **m_PathingDirectionMultiplier** = 1f
- Vector3 **m_StartPosition**
- Quaternion **m_StartRotation**

Properties

- Transform **MoveTransform** `[get]`

  Returns the Transform that will be moved by this component.
- Vector3 **LastTargetDirection** `[get]`

  Returns the last calculated target heading / target direction for this component.

Additional Inherited Members

5.17.1 Detailed Description

A component for moving a Transform along waypoints.

Author: Mathew Aloisio

5.17.2 Member Function Documentation

5.17.2.1 ForceMoveToWaypoint()

```
override void Waypoints.Pathing.TransformPathMove.ForceMoveToWaypoint (
            int pPointIndex )  [virtual]
```

Force the position of the transform of the PathMove to the position of the waypoint in the index. This only forces the position and will instantly be overridden by an active path move.

NOTE: This WILL NOT invoke any waypoint reached events.

Parameters

pPointIndex	

Implements Waypoints.Pathing.PathMove.

5.17.2.2 IsInterpolatorValid()

```
override bool Waypoints.Pathing.TransformPathMove.IsInterpolatorValid ( )  [virtual]
```

Returns true if the TransformPathMove's interpolator is valid and matching the waypoint settings, otherwise false.

Returns

true if the TransformPathMove's interpolator is valid and matching the waypoint settings, otherwise false.

Reimplemented from Waypoints.Pathing.PathMove.

5.17.2.3 SetMoveSpeed()

```
void Waypoints.Pathing.TransformPathMove.SetMoveSpeed (
            float pSpeed )
```

Sets the movement speed of the TransformPathMove in meters per second.

Parameters

pSpeed	

5.17.2.4 SetRotationSpeed()

```
void Waypoints.Pathing.TransformPathMove.SetRotationSpeed (
            float pDegreesPerSecond )
```

Sets the rotation speed of the TransformPathMove in degrees per second.

Parameters

pDegreesPerSecond	

5.17.2.5 SetSpeedMultiplier()

```
void Waypoints.Pathing.TransformPathMove.SetSpeedMultiplier (
             float pMultiplier )
```

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

Parameters

pMultiplier	

5.17.2.6 TeleportToWaypoint()

```
override void Waypoints.Pathing.TransformPathMove.TeleportToWaypoint (
             int pIndex )  [virtual]
```

Allows the TransformPathMove to be teleported directly to a waypoint instantly.

Parameters

pIndex	

Reimplemented from Waypoints.Pathing.PathMove.

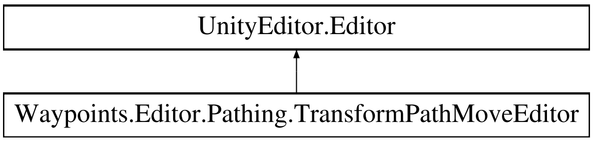
The documentation for this class was generated from the following file:

- TransformPathMove.cs

5.18 Waypoints.Editor.Pathing.TransformPathMoveEditor Class Reference

A custom Editor for TransformPathMove components.

Inheritance diagram for Waypoints.Editor.Pathing.TransformPathMoveEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

5.18.1 Detailed Description

A custom Editor for TransformPathMove components.

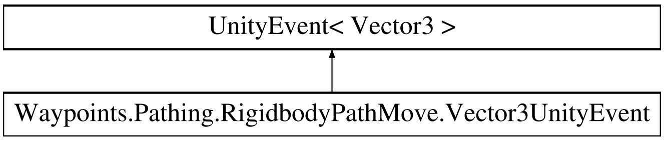
Author: Mathew Aloisio

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

- TransformPathMoveEditor.cs

5.19 Waypoints.Pathing.RigidbodyPathMove.Vector3UnityEvent Class Reference

Inheritance diagram for Waypoints.Pathing.RigidbodyPathMove.Vector3UnityEvent:

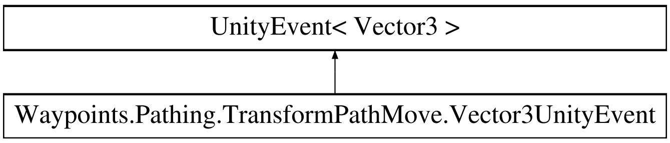


The documentation for this class was generated from the following file:

- RigidbodyPathMove.cs

5.20 Waypoints.Pathing.TransformPathMove.Vector3UnityEvent Class Reference

Inheritance diagram for Waypoints.Pathing.TransformPathMove.Vector3UnityEvent:

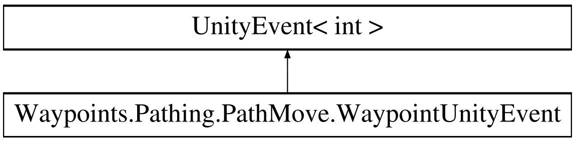


The documentation for this class was generated from the following file:

- TransformPathMove.cs

5.21 Waypoints.Pathing.PathMove.WaypointUnityEvent Class Reference

Inheritance diagram for Waypoints.Pathing.PathMove.WaypointUnityEvent:



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

- PathMove.cs

Index