AR + GPS Location

3.0.0

Generated by Doxygen 1.8.14

Contents

Chapter 1

Unity AR+GPS Location

The `AR+GPS Location` package brings the ability to position 3D objects in real-world geographical locations via their GPS coordinates using Unity and Augmented-Reality. It currently works Unity's `AR Foundation`, but we are working on getting it to work with the `Unity ARKit` plugin in the next releases.

This project is in it's first versions and we need a lot of feedback to make it as useful as possible for everyone! Bug reports and feature requests are more than welcomed and will be implemented swiftly.

If you purchase this package you get full access to the github repository. Just send an email to `daniel.↩mbfm@gmail.com` with the code in the ACCESS_CODE.TXT file and your github username/email.

Main Features

- Place 3D Objects in geographical positions defined by their latitude, longitude and altitude.
- Place 3D Text markers on real-world points of interest (example using OpenStreetmaps is included.)
- Smooth movements on device location and heading updates.
- Move objects or place them along paths (Catmull-rom splines) on the map.
- Augmented reality floor shadows.
- Double precision vector structs, `DVector2` and `DVector3`.
- General purpose Catmull-rom curves and splines.

Sample Scenes

- **Scenes/ARLocation Basic**: A Basic scene with one positioned object.
- **Scenes/ARLocation 3D Text**: Shows how to place 3D on points of interest on the map. You can either add them manually on the inspector, load a xml file from OpenStreetMaps/Overpass, or fetch them from the internet via a Overpass API request.
- **Scenes/ARLocation Jet Fighter** and **Scenes/ARLocation Jet Fighter Squad**: Shows a jet fighter (a jet squad in the second) flight along a predefined route on the map.
- **Scenes/ARLocation Walking Dead**: A Zombie walking around your neighborhood!
- **Scenes/ARLocation Path Line Render**: Using a line-renderer to render a ARLocationPath.
- **Scenes/ARLocation Place At Locations**: Places a prefab in a number of predefined locations.

Limitations

- Altitude information is usually very imprecise so, currently, it's best to use heights relative to the device position.
- If the user is moving, after some distance the scene orientation and true north direction may start to deteriorate in quality. To bypass that, there is a option to reset the AR Session after the user has walked some distance from the initial position.
- Due to GPS precision, the position data can jump around a lot, making object jump round in the scene. We use movement smoothing to mitigate the effects of this.
- Movement smoothing must be used lightly on objects moving along paths. Use values around 10.0f.

Roadmap

- `Unity ARKit` plugin support.
- *AR Hotstpots*: Regular AR experiences (e.g., using plane detection) triggered at specific locations.
- Dynamic floor height/level calculation by using nearest detected planes.
- Double precision location data by using native modules.
- Add more curve/spline types (Only Catmull-rom splines currently.)
- Improve movement smoothing (i.e., of movement due to location changes) on object moving along paths.
- Implement closed curves/paths.

Documentation

Read the full documentation `here`.

Contact

If you have any questions, contact me via e-mail at `daniel.mbfm@gmail.com`, at twitter , or at my website `danielfortes.com`.

Chapter 2

Namespace Index

2.1 Packages

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

ARLocation . . . . . ??
ARLocation.Session . . . . . ??
ARLocation.UI . . . . . ??
ARLocation.Utils . . . . . ??

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

ARLocation.AngleLowPassFilter . . . ??
ARLocationEditorConfigManager . . . ??
ARLocation.Curve . . . ??
    ARLocation.CatmullRomCurve . . . ??
    ARLocation.Line . . . ??
ARLocation.CurvePointData . . . ??
DefineSymbols . . . ??
DefineSymbolsManager . . . ??
ARLocation.DVector2 . . . ??
ARLocation.DVector3 . . . ??
Editor
    ARLocation.ARLocationConfigInspector . . . ??
    ARLocation.ARLocationManagerInspector . . . ??
    ARLocation.ARLocationOrientationInspector . . . ??
    ARLocation.ARLocationProviderInspector . . . ??
    ARLocation.LocationPathInspector . . . ??
ARLocation.PlaceAtLocations.Entry . . . ??
ARLocation.HeadingReading . . . ??
ARLocation.Hotspot.HotspotSettingsData . . . ??
ARLocation.Session.IARSessionManager . . . ??
    ARLocation.Session.ARFoundationSessionManager . . . ??
ARLocation.ILocationProvider . . . ??
    ARLocation.AbstractLocationProvider . . . ??
        ARLocation.MockLocationProvider . . . ??
        ARLocation.UnityLocationProvider . . . ??
ARLocation.Location . . . ??
ARLocation.LocationPropertyData . . . ??
ARLocation.LocationProviderOptions . . . ??
ARLocation.LocationReading . . . ??
ARLocation.PlaceAtLocation.LocationSettingsData . . . ??
ARLocation.LowPassFilter . . . ??
ARLocation.Utils.Misc . . . ??
MonoBehaviour
    ARLocation.GroundHeight . . . ??

ARLocation.Hotspot . . . ??
ARLocation.MoveAlongPath . . . ??
ARLocation.PlaceAlongPath . . . ??
ARLocation.PlaceAtLocation . . . ??
ARLocation.PlaceAtLocations . . . ??
ARLocation.RenderPathLine . . . ??
ARLocation.UI.ARTrackingInfo . . . ??
ARLocation.UI.DebugInfoOverlay . . . ??
ARLocation.UI.LoadingBar . . . ??
ARLocation.UI.LocationProviderInfo . . . ??
ARLocation.UI.OrientationInfo . . . ??
ARLocation.Utils.CreatePointOfInterestTextMeshes . . . ??
ARLocation.Utils.DebugCanvas . . . ??
ARLocation.Utils.DevCameraController . . . ??
ARLocation.Utils.FaceCamera . . . ??
ARLocation.Utils.FadeOutTextMesh . . . ??
ARLocation.Utils.FollowCameraPosition . . . ??
ARLocation.Utils.RotateObject . . . ??
ARLocation.Utils.SelectScene . . . ??
ARLocation.Utils.ShowHideSelfOnPointerClick . . . ??
ARLocation.Utils.Singleton< T > . . . ??
ARLocation.Utils.SmoothMove . . . ??
ARLocation.Utils.MovingAveragePosition . . . ??
ARLocation.Utils.OpenStreetMapOptions . . . ??
ARLocation.Utils.OverpassRequestData . . . ??
ARLocation.OverrideAltitudeData . . . ??
ARLocation.MoveAlongPath.PathSettingsData . . . ??
ARLocation.PlaceAtLocation.PlaceAtOptions . . . ??
ARLocation.MoveAlongPath.PlacementSettingsData . . . ??
ARLocation.Utils.POIData . . . ??
PropertyDrawer
ARLocation.LocationPropertyDataDrawer . . . ??
ARLocation.OverrideAltitudeDataDrawer . . . ??
ScriptableObject
ARLocation.ARLocationConfig . . . ??
ARLocation.LocationData . . . ??
ARLocation.LocationPath . . . ??
ARLocation.GroundHeight.SettingsData . . . ??
ARLocation.Utils.Singleton< ARLocationManager > . . . ??
ARLocation.ARLocationManager . . . ??
ARLocation.Utils.Singleton< ARLocationOrientation > . . . ??
ARLocation.ARLocationOrientation . . . ??
ARLocation.Utils.Singleton< ARLocationProvider > . . . ??
ARLocation.ARLocationProvider . . . ??
ARLocation.Spline . . . ??
ARLocation.CatmullRomSpline . . . ??
ARLocation.LinearSpline . . . ??
ARLocation.MoveAlongPath.StateData . . . ??
ARLocation.PlaceAtLocation.StateData . . . ??
ARLocation.Hotspot.StateData . . . ??
ARLocation.GroundHeight.StateData . . . ??
UnityEvent
ARLocation.ARLocationOrientation.OnBeforeOrientationUpdatedEvent . . . ??
ARLocation.ARLocationProvider.CompassUpdatedUnityEvent . . . ??
ARLocation.ARLocationProvider.LocationEnabledUnityEvent . . . ??
ARLocation.ARLocationProvider.LocationUpdatedUnityEvent . . . ??
ARLocation.Hotspot.OnHotspotActivatedUnityEvent . . . ??

ARLocation.PlaceAtLocation.ObjectUpdatedEvent . . . . . ??

Chapter 4

Class Index

4.1 Class List

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

ARLocation.AbstractLocationProvider
: Abstract location provider. All concrete location provider implementations should derive from this. **??**

ARLocation.AngleLowPassFilter . . . . . **??**

ARLocation.Session.ARFoundationSessionManager . . . . . **??**

ARLocation.ARLocationConfig
: This scriptable object holds the global configuration data for the AR + GPS Location plugin. . . **??**

ARLocation.ARLocationConfigInspector
: Inspector for the ARLocationConfig. This inspector is the main configuration interface for the AR+GPS Location plugin. . . . . **??**

ARLocationEditorConfigManager
: This is a static class that makes sure that there always is a ARLocationConfig resource for the project. . . . . **??**

ARLocation.ARLocationManager
: This Component manages all positioned GameObjects, synchronizing their world position in the scene with their geographical coordinates. This is done by calculating their position relative to the device's position . . . . . **??**

ARLocation.ARLocationManagerInspector . . . . . **??**

ARLocation.ARLocationOrientation
: This component should be placed on the "ARLocationRoot" GameObject (which should be a child of the "AR Session Origin") for correctly aligning the coordinate system to the north/east geographical lines. . . . . **??**

ARLocation.ARLocationOrientationInspector . . . . . **??**

ARLocation.ARLocationProvider . . . . . **??**

ARLocation.ARLocationProviderInspector . . . . . **??**

ARLocation.UI.ARTrackingInfo . . . . . **??**

ARLocation.CatmullRomCurve
: A catmull-rom curve. . . . . **??**

ARLocation.CatmullRomSpline
: A (open-ended) catmull-rom spline, which interpolates a set points by joining catmull-rom curves together. . . . . **??**

ARLocation.ARLocationProvider.CompassUpdatedUnityEvent . . . . . **??**

ARLocation.Utils.CreatePointOfInterestTextMeshes . . . . . **??**

ARLocation.Curve . . . . . **??**

ARLocation.CurvePointData
: A struct holding a pair of point/tangent values. . . . . **??**

ARLocation.Utils.DebugCanvas . . . ??
ARLocation.UI.DebugInfoOverlay . . . ??
DefineSymbols
Utility class to manage a list of symbol strings. . . . ??
DefineSymbolsManager
Utility class that manages Define Symbols for a given set of build targets. . . . ??
ARLocation.Utils.DevCameraController . . . ??
ARLocation.DVector2 . . . ??
ARLocation.DVector3 . . . ??
ARLocation.PlaceAtLocations.Entry . . . ??
ARLocation.Utils.FaceCamera . . . ??
ARLocation.Utils.FadeOutTextMesh . . . ??
ARLocation.Utils.FollowCameraPosition . . . ??
ARLocation.GroundHeight . . . ??
ARLocation.HeadingReading . . . ??
ARLocation.Hotspot . . . ??
ARLocation.Hotspot.HotspotSettingsData . . . ??
ARLocation.Session.IARSessionManager . . . ??
ARLocation.ILocationProvider . . . ??
ARLocation.Line . . . ??
ARLocation.LinearSpline . . . ??
ARLocation.UI.LoadingBar . . . ??
ARLocation.Location
Represents a geographical location. . . . ??
ARLocation.LocationData
Data used to construct a spline passing trough a set of geographical locations. . . . ??
ARLocation.ARLocationProvider.LocationEnabledUnityEvent . . . ??
ARLocation.LocationPath
Data used to construct a spline passing trough a set of geographical locations. . . . ??
ARLocation.LocationPathInspector . . . ??
ARLocation.LocationPropertyData . . . ??
ARLocation.LocationPropertyDataDrawer . . . ??
ARLocation.UI.LocationProviderInfo . . . ??
ARLocation.LocationProviderOptions . . . ??
ARLocation.LocationReading . . . ??
ARLocation.PlaceAtLocation.LocationSettingsData . . . ??
ARLocation.ARLocationProvider.LocationUpdatedUnityEvent . . . ??
ARLocation.LowPassFilter . . . ??
ARLocation.Utils.Misc . . . ??
ARLocation.MockLocationProvider . . . ??
ARLocation.MoveAlongPath
This component, when attached to a GameObject, makes it traverse a path that interpolates a given set of geographical locations. . . . ??
ARLocation.Utils.MovingAveragePosition . . . ??
ARLocation.PlaceAtLocation.ObjectUpdatedEvent . . . ??
ARLocation.ARLocationOrientation.OnBeforeOrientationUpdatedEvent . . . ??
ARLocation.Hotspot.OnHotspotActivatedUnityEvent . . . ??
ARLocation.Utils.OpenStreetMapOptions . . . ??
ARLocation.UI.OrientationInfo . . . ??
ARLocation.Utils.OverpassRequestData . . . ??
ARLocation.OverrideAltitudeData . . . ??
ARLocation.OverrideAltitudeDataDrawer . . . ??
ARLocation.MoveAlongPath.PathSettingsData . . . ??
ARLocation.PlaceAlongPath
This component places instances of a given prefab/GameObject along equally spaced positions in a LocationPath. Should be placed in the ARLocationRoot GameObject. . . . ??
ARLocation.PlaceAtLocation
Apply to a GameObject to place it at a specified geographic location. . . . ??

ARLocation.PlaceAtLocations
This class instantiates a prefab at the given GPS locations. Must be in the ARLocationRoot GameObject with a ARLocatedObjectsManager Component. . . . . ??
ARLocation.PlaceAtLocation.PlaceAtOptions . . . . . ??
ARLocation.MoveAlongPath.PlacementSettingsData . . . . . ??
ARLocation.Utils.POIData . . . . . ??
ARLocation.RenderPathLine
This component renders a LocationPath using a given LineRenderer. . . . . ??
ARLocation.Utils.RotateObject . . . . . ??
ARLocation.Utils.SelectScene . . . . . ??
ARLocation.GroundHeight.SettingsData . . . . . ??
ARLocation.Utils.ShowHideSelfOnPointerClick . . . . . ??
ARLocation.Utils.Singleton< T > . . . . . ??
ARLocation.Utils.SmoothMove . . . . . ??
ARLocation.Spline . . . . . ??
ARLocation.MoveAlongPath.StateData . . . . . ??
ARLocation.PlaceAtLocation.StateData . . . . . ??
ARLocation.Hotspot.StateData . . . . . ??
ARLocation.GroundHeight.StateData . . . . . ??
ARLocation.UnityLocationProvider . . . . . ??

Chapter 5

Namespace Documentation

5.1 ARLocation Namespace Reference

Namespaces

Classes

- class AbstractLocationProvider

  Abstract location provider. All concrete location provider implementations should derive from this.
- class **AndroidMagneticDeclination**
- class AngleLowPassFilter
- class **ARLocation**

  This static class loads the global configuration for the AR + GPS Location plugin.
- class ARLocationConfig

  This scriptable object holds the global configuration data for the AR + GPS Location plugin.
- class ARLocationConfigInspector

  Inspector for the ARLocationConfig. This inspector is the main configuration interface for the AR+GPS Location plugin.
- class ARLocationManager

  This Component manages all positioned GameObjects, synchronizing their world position in the scene with their geographical coordinates. This is done by calculating their position relative to the device's position.
- class ARLocationManagerInspector
- class ARLocationOrientation

  This component should be placed on the "ARLocationRoot" GameObject (which should be a child of the "AR Session Origin") for correctly aligning the coordinate system to the north/east geographical lines.
- class ARLocationOrientationInspector
- class ARLocationProvider
- class ARLocationProviderInspector
- class CatmullRomCurve

  A catmull-rom curve.
- class CatmullRomSpline

  A (open-ended) catmull-rom spline, which interpolates a set points by joining catmull-rom curves together.
- class Curve
- struct CurvePointData

  A struct holding a pair of point/tangent values.
- struct DVector2

- struct DVector3
- class **GameObjectMenuItems**
- class GroundHeight
- struct HeadingReading
- class Hotspot
- interface ILocationProvider
- class Line
- class LinearSpline
- class Location

  Represents a geographical location.
- class LocationData

  Data used to construct a spline passing trough a set of geographical locations.
- class LocationPath

  Data used to construct a spline passing trough a set of geographical locations.
- class LocationPathInspector
- class LocationPropertyData
- class LocationPropertyDataDrawer
- class LocationProviderOptions
- struct LocationReading
- class LowPassFilter
- class **MathUtils**
- class MockLocationProvider
- class MoveAlongPath

  This component, when attached to a GameObject, makes it traverse a path that interpolates a given set of geographical locations.
- class OverrideAltitudeData
- class OverrideAltitudeDataDrawer
- class PlaceAlongPath

  This component places instances of a given prefab/GameObject along equally spaced positions in a LocationPath. Should be placed in the ARLocationRoot GameObject.
- class PlaceAtLocation

  Apply to a GameObject to place it at a specified geographic location.
- class PlaceAtLocations

  This class instantiates a prefab at the given GPS locations. Must be in the `ARLocationRoot` *GameObject with a* `ARLocatedObjectsManager` *Component.*
- class RenderPathLine

  This component renders a LocationPath using a given LineRenderer.
- class Spline
- class UnityLocationProvider

Enumerations

- enum **LocationProviderStatus** { **Idle**, **Initializing**, **Started**, **Failed** }
- enum **AltitudeMode** { **Absolute**, **DeviceRelative**, **GroundRelative**, **Ignore** }
- enum **SplineType** { **CatmullromSpline**, **LinearSpline** }

Functions

- delegate void **LocationUpdatedDelegate** (LocationReading currentLocation, LocationReading lastLocation)
- delegate void **CompassUpdateDelegate** (HeadingReading heading, HeadingReading lastReading)
- delegate void **LocationEnabledDelegate** ()
- delegate void **LocationFailedDelegate** (string message)

5.2 ARLocation.Session Namespace Reference

Classes

- class ARFoundationSessionManager
- interface IARSessionManager

5.3 ARLocation.UI Namespace Reference

Classes

- class ARTrackingInfo
- class DebugInfoOverlay
- class LoadingBar
- class LocationProviderInfo
- class OrientationInfo

5.4 ARLocation.Utils Namespace Reference

Classes

- class CreatePointOfInterestTextMeshes
- class DebugCanvas
- class DevCameraController
- class FaceCamera
- class FadeOutTextMesh
- class FollowCameraPosition
- class **Logger**
- class Misc
- class MovingAveragePosition
- class OpenStreetMapOptions
- class OverpassRequestData
- class POIData
- class RotateObject
- class SelectScene
- class ShowHideSelfOnPointerClick
- class Singleton
- class SmoothMove

Chapter 6

Class Documentation

6.1 ARLocation.AbstractLocationProvider Class Reference

Abstract location provider. All concrete location provider implementations should derive from this.

Inheritance diagram for ARLocation.AbstractLocationProvider:

Public Member Functions

- virtual IEnumerator **Start** (uint maxWaitTime=10000, uint delay=0)
- void **ForceLocationUpdate** ()
- virtual void **Update** ()
- void **Restart** ()
- void **ResetStartPoint** ()
- void **SetCompassLowPassFactor** (double factor)
- string **GetStatusString** ()
- string **GetInfoString** ()
- void **OnEnabled** (LocationEnabledDelegate del)
- void **OnFail** (LocationFailedDelegate del)
- void Pause ()

  Pauses location updates
- void Resume ()

  Resumes location updates

Public Attributes

- bool **HasStarted** => Status == LocationProviderStatus.Started

Protected Member Functions

- abstract LocationReading ReadLocation ()

  Reads the location from the device; should be implemented by each provider.
- abstract HeadingReading ReadHeading ()

  Reads the heading from the device; should be implemented by each provider.
- abstract void RequestLocationAndCompassUpdates ()

  Requests the location and compass updates from the device; should be implemented by each provider.
- abstract void UpdateLocationRequestStatus ()

  Updates the location service status from the device; should be implemented by each provider.
- virtual void **InnerOnEnabled** ()
- void **EmitLocationUpdated** ()
- void **EmitLocationUpdatedRaw** ()
- void **EmitCompassUpdated** ()
- void **UpdateLocation** (LocationReading newLocation)
- void **UpdateHeading** (HeadingReading newHeading)
- HeadingReading **ApplyCompassLpFilters** (HeadingReading reading)
- bool **ShouldUpdateHeading** (HeadingReading newHeading)
- bool **ShouldUpdateLocation** (LocationReading newLocation)

Properties

- abstract string Name `[get]`

  The name of the location provider.
- LocationProviderOptions Options `[get, set]`

  The options of the location provider.
- LocationReading CurrentLocation `[get, protected set]`

  Gets or sets the current location.
- LocationReading LastLocation `[get, protected set]`

  Gets or sets the previous location.
- LocationReading **LastLocationRaw** `[get, protected set]`
- LocationReading CurrentLocationRaw `[get, protected set]`

  Gets or sets the previous raw location reading.
- HeadingReading CurrentHeading `[get, protected set]`

  The current heading reading.
- HeadingReading LastHeading `[get, protected set]`

  The previous heading reading.
- LocationReading FirstLocation `[get, protected set]`

  The start point, i.e., the first measured location.
- LocationProviderStatus Status `[get, protected set]`

  Gets or sets the current status of the location provider.
- bool IsEnabled `[get, protected set]`

  If true, the location provider is enablied and getting regular location updated from the device.
- bool FirstReading `[get, protected set]`

  If true, the first reading has not occured yet.
- abstract bool IsCompassEnabled `[get]`

  If true, the provider has a functioning magnetic compass sensor.
- float StartTime `[get, protected set]`

  The start time of the location provider.
- bool Paused `[get, protected set]`

  If true, location updates are paused.
- int **LocationUpdateCount** `[get, protected set]`
- bool **ApplyCompassTiltCompensationOnAndroid** = true `[get, set]`
- double **DistanceFromStartPoint** `[get]`

Events

- LocationUpdatedDelegate LocationUpdated

  Event for when a new location data is received.
- CompassUpdateDelegate CompassUpdated

  Event for when a new compass data is received.
- LocationEnabledDelegate **LocationEnabled**
- LocationFailedDelegate **LocationFailed**
- LocationUpdatedDelegate **LocationUpdatedRaw**

6.1.1 Detailed Description

Abstract location provider. All concrete location provider implementations should derive from this.

Definition at line 14 of file AbstractLocationProvider.cs.

6.1.2 Member Function Documentation

6.1.2.1 Pause()

```
void ARLocation.AbstractLocationProvider.Pause ( )
```

Pauses location updates

Implements ARLocation.ILocationProvider.

Definition at line 468 of file AbstractLocationProvider.cs.

6.1.2.2 ReadHeading()

```
abstract HeadingReading ARLocation.AbstractLocationProvider.ReadHeading ( )  [protected], [pure virtual]
```

Reads the heading from the device; should be implemented by each provider.

Returns

The heading.

Implemented in ARLocation.UnityLocationProvider, and ARLocation.MockLocationProvider.

6.1.2.3 ReadLocation()

```
abstract LocationReading ARLocation.AbstractLocationProvider.ReadLocation ( )  [protected],
[pure virtual]
```

Reads the location from the device; should be implemented by each provider.

Returns

The location.

Implemented in ARLocation.UnityLocationProvider, and ARLocation.MockLocationProvider.

6.1.2.4 RequestLocationAndCompassUpdates()

```
abstract void ARLocation.AbstractLocationProvider.RequestLocationAndCompassUpdates ( )  [protected],
[pure virtual]
```

Requests the location and compass updates from the device; should be implemented by each provider.

Implemented in ARLocation.MockLocationProvider, and ARLocation.UnityLocationProvider.

6.1.2.5 Resume()

```
void ARLocation.AbstractLocationProvider.Resume ( )
```

Resumes location updates

Implements ARLocation.ILocationProvider.

Definition at line 476 of file AbstractLocationProvider.cs.

6.1.2.6 UpdateLocationRequestStatus()

```
abstract void ARLocation.AbstractLocationProvider.UpdateLocationRequestStatus ( )  [protected],
[pure virtual]
```

Updates the location service status from the device; should be implemented by each provider.

Implemented in ARLocation.MockLocationProvider, and ARLocation.UnityLocationProvider.

6.1.3 Property Documentation

6.1.3.1 CurrentHeading

```
HeadingReading ARLocation.AbstractLocationProvider.CurrentHeading [get], [protected set]
```

The current heading reading.

The current heading.

Definition at line 56 of file AbstractLocationProvider.cs.

6.1.3.2 CurrentLocation

```
LocationReading ARLocation.AbstractLocationProvider.CurrentLocation [get], [protected set]
```

Gets or sets the current location.

The current location.

Definition at line 36 of file AbstractLocationProvider.cs.

6.1.3.3 CurrentLocationRaw

```
LocationReading ARLocation.AbstractLocationProvider.CurrentLocationRaw [get], [protected set]
```

Gets or sets the previous raw location reading.

The raw location last.

Definition at line 50 of file AbstractLocationProvider.cs.

6.1.3.4 FirstLocation

```
LocationReading ARLocation.AbstractLocationProvider.FirstLocation [get], [protected set]
```

The start point, i.e., the first measured location.

The start point.

Definition at line 68 of file AbstractLocationProvider.cs.

6.1.3.5 FirstReading

```
bool ARLocation.AbstractLocationProvider.FirstReading  [get], [protected set]
```

If true, the first reading has not occured yet.

`true` if first reading; otherwise, `false`.

Definition at line 87 of file AbstractLocationProvider.cs.

6.1.3.6 IsCompassEnabled

```
abstract bool ARLocation.AbstractLocationProvider.IsCompassEnabled  [get]
```

If true, the provider has a functioning magnetic compass sensor.

`true` if is compass enabled; otherwise, `false`.

Definition at line 93 of file AbstractLocationProvider.cs.

6.1.3.7 IsEnabled

```
bool ARLocation.AbstractLocationProvider.IsEnabled  [get], [protected set]
```

If true, the location provider is enablied and getting regular location updated from the device.

`true` if is enabled; otherwise, `false`.

Definition at line 81 of file AbstractLocationProvider.cs.

6.1.3.8 LastHeading

```
HeadingReading ARLocation.AbstractLocationProvider.LastHeading  [get], [protected set]
```

The previous heading reading.

The last heading.

Definition at line 62 of file AbstractLocationProvider.cs.

6.1.3.9 LastLocation

```
LocationReading ARLocation.AbstractLocationProvider.LastLocation  [get], [protected set]
```

Gets or sets the previous location.

The last location.

Definition at line 42 of file AbstractLocationProvider.cs.

6.1.3.10 Name

```
abstract string ARLocation.AbstractLocationProvider.Name  [get]
```

The name of the location provider.

The name.

Definition at line 24 of file AbstractLocationProvider.cs.

6.1.3.11 Options

```
LocationProviderOptions ARLocation.AbstractLocationProvider.Options  [get], [set]
```

The options of the location provider.

The options.

Definition at line 30 of file AbstractLocationProvider.cs.

6.1.3.12 Paused

```
bool ARLocation.AbstractLocationProvider.Paused  [get], [protected set]
```

If true, location updates are paused.

Definition at line 104 of file AbstractLocationProvider.cs.

6.1.3.13 StartTime

```
float ARLocation.AbstractLocationProvider.StartTime  [get], [protected set]
```

The start time of the location provider.

The start time.

Definition at line 99 of file AbstractLocationProvider.cs.

6.1.3.14 Status

```
LocationProviderStatus ARLocation.AbstractLocationProvider.Status  [get], [protected set]
```

Gets or sets the current status of the location provider.

The status.

Definition at line 74 of file AbstractLocationProvider.cs.

6.1.4 Event Documentation

6.1.4.1 CompassUpdated

```
CompassUpdateDelegate ARLocation.AbstractLocationProvider.CompassUpdated
```

Event for when a new compass data is received.

Definition at line 123 of file AbstractLocationProvider.cs.

6.1.4.2 LocationUpdated

```
LocationUpdatedDelegate ARLocation.AbstractLocationProvider.LocationUpdated
```

Event for when a new location data is received.

Definition at line 118 of file AbstractLocationProvider.cs.

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

- Assets/ARLocation/Scripts/Location/AbstractLocationProvider.cs

6.2 ARLocation.AngleLowPassFilter Class Reference

Public Member Functions

- **AngleLowPassFilter** (double smoothFactor)
- double **Apply** (double angle)
- void **SetFactor** (double factor)

6.2.1 Detailed Description

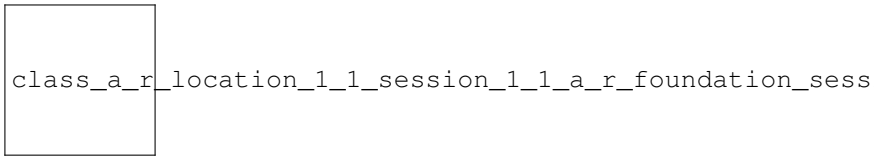
Definition at line 5 of file AngleLowPassFilter.cs.

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

- Assets/ARLocation/Scripts/Math/AngleLowPassFilter.cs

6.3 ARLocation.Session.ARFoundationSessionManager Class Reference

Inheritance diagram for ARLocation.Session.ARFoundationSessionManager:



Public Member Functions

- **ARFoundationSessionManager** (ARSession session)
- void **Reset** (Action callback)
- string **GetSessionInfoString** ()
- string **GetProviderString** ()
- void **OnARTrackingStarted** (Action callback)
- void **OnARTrackingRestored** (Action callback)
- void **OnARTrackingLost** (Action callback)

Properties

- bool **DebugMode** `[get, set]`

6.3.1 Detailed Description

Definition at line 42 of file ARFoundationSessionManager.cs.

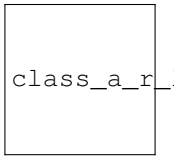
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARSession/ARFoundationSessionManager.cs

6.4 ARLocation.ARLocationConfig Class Reference

This scriptable object holds the global configuration data for the AR + GPS Location plugin.

Inheritance diagram for ARLocation.ARLocationConfig:



Public Types

- enum **ARLocationDistanceFunc** { **Haversine**, **PlaneSpherical**, **PlaneEllipsoidalFcc** }

Public Attributes

- double **EarthRadiusInKM** = 6372.8
- ARLocationDistanceFunc **DistanceFunction** = ARLocationDistanceFunc.Haversine
- float **InitialGroundHeightGuess** = 1.4f
- float **MinGroundHeight** = 0.4f
- float **MaxGroundHeight** = 3.0f
- float **VuforiaGroundHitTestDistance** = 4.0f
- bool **UseVuforia**

Properties

- static string **Version** `[get]`

6.4.1 Detailed Description

This scriptable object holds the global configuration data for the AR + GPS Location plugin.

Definition at line 11 of file ARLocationConfig.cs.

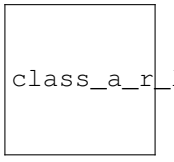
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ScriptableObjects/ARLocationConfig.cs

6.5 ARLocation.ARLocationConfigInspector Class Reference

Inspector for the ARLocationConfig. This inspector is the main configuration interface for the AR+GPS Location plugin.

Inheritance diagram for ARLocation.ARLocationConfigInspector:



Public Member Functions

- override void **OnInspectorGUI** ()

6.5.1 Detailed Description

Inspector for the ARLocationConfig. This inspector is the main configuration interface for the AR+GPS Location plugin.

Definition at line 14 of file ARLocationConfigInspector.cs.

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

- Assets/ARLocation/Editor/ARLocationConfigInspector.cs

6.6 ARLocationEditorConfigManager Class Reference

This is a static class that makes sure that there always is a ARLocationConfig resource for the project.

6.6.1 Detailed Description

This is a static class that makes sure that there always is a ARLocationConfig resource for the project.

Definition at line 10 of file ARLocationEditorConfigManager.cs.

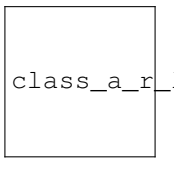
The documentation for this class was generated from the following file:

- Assets/ARLocation/Editor/ARLocationEditorConfigManager.cs

6.7 ARLocation.ARLocationManager Class Reference

This Component manages all positioned GameObjects, synchronizing their world position in the scene with their geographical coordinates. This is done by calculating their position relative to the device's position.

Inheritance diagram for ARLocation.ARLocationManager:



Public Member Functions

- override void **Awake** ()
- void ResetARSession (Action cb=null)

  This will reset the AR Session and the AR+GPS system, repositioning all objects.
- void Restart ()

  This will restart the AR+GPS system, repositioning all the objects.
- string **GetARSessionInfoString** ()
- string **GetARSessionProviderString** ()
- void OnARTrackingStarted (Action o)

  Add a event listener for when the AR Tracking starts.
- void OnARTrackingRestored (Action callback)

  Add a event listener for when the AR Tracking regained after it was lost.
- void OnARTrackingLost (Action callback)

  Add a event listener for when the AR Tracking is lost.

Public Attributes

- Camera **Camera**
- bool **WaitForARTrackingToStart** = true
- bool **RestartWhenARTrackingIsRestored**
- bool **SetTargetFrameRateTo60Mhz** = true
- bool **DebugMode**
- UnityEvent **OnTrackingStarted**
- UnityEvent **OnTrackingLost**
- UnityEvent **OnTrackingRestored**

Properties

- IARSessionManager SessionManager `[get]`

  The instance of the 'IARSessionManager'. Handles the interface with the underlying AR session (i.e., Vuforia or AR Foundation).
- Camera MainCamera `[get]`

  The 'MainCamera' that is being used for rendering the AR content.

6.7.1 Detailed Description

This Component manages all positioned GameObjects, synchronizing their world position in the scene with their geographical coordinates. This is done by calculating their position relative to the device's position.

Should be placed in a GameObject called "ARLocationRoot", whose parent is the "AR Session Origin".

Definition at line 28 of file ARLocationManager.cs.

6.7.2 Member Function Documentation

6.7.2.1 OnARTrackingLost()

```
void ARLocation.ARLocationManager.OnARTrackingLost (
            Action callback )
```

Add a event listener for when the AR Tracking is lost.

Parameters

callback	

Definition at line 206 of file ARLocationManager.cs.

6.7.2.2 OnARTrackingRestored()

```
void ARLocation.ARLocationManager.OnARTrackingRestored (
            Action callback )
```

Add a event listener for when the AR Tracking regained after it was lost.

Parameters

callback	

Definition at line 197 of file ARLocationManager.cs.

6.7.2.3 OnARTrackingStarted()

```
void ARLocation.ARLocationManager.OnARTrackingStarted (
            Action o )
```

Add a event listener for when the AR Tracking starts.

Parameters

o	

Definition at line 181 of file ARLocationManager.cs.

6.7.2.4 ResetARSession()

```
void ARLocation.ARLocationManager.ResetARSession (
            Action cb = null )
```

This will reset the AR Session and the AR+GPS system, repositioning all objects.

Parameters

cb	Optional callback, called when the system has restarted.

Definition at line 141 of file ARLocationManager.cs.

6.7.2.5 Restart()

```
void ARLocation.ARLocationManager.Restart ( )
```

This will restart the AR+GPS system, repositioning all the objects.

Definition at line 156 of file ARLocationManager.cs.

6.7.3 Property Documentation

6.7.3.1 MainCamera

```
Camera ARLocation.ARLocationManager.MainCamera  [get]
```

The 'MainCamera' that is being used for rendering the AR content.

Definition at line 60 of file ARLocationManager.cs.

6.7.3.2 SessionManager

```
IARSessionManager ARLocation.ARLocationManager.SessionManager  [get]
```

The instance of the 'IARSessionManager'. Handles the interface with the underlying AR session (i.e., Vuforia or AR Foundation).

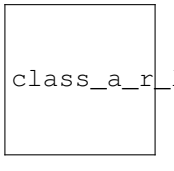
Definition at line 55 of file ARLocationManager.cs.

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

- Assets/ARLocation/Scripts/Components/ARLocationManager.cs

6.8 ARLocation.ARLocationManagerInspector Class Reference

Inheritance diagram for ARLocation.ARLocationManagerInspector:



Public Member Functions

- override void **OnInspectorGUI** ()

6.8.1 Detailed Description

Definition at line 7 of file ARLocationManagerInspector.cs.

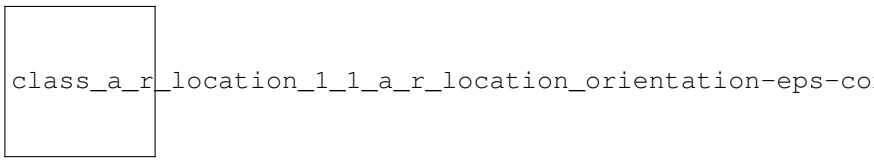
The documentation for this class was generated from the following file:

- Assets/ARLocation/Editor/ARLocationManagerInspector.cs

6.9 ARLocation.ARLocationOrientation Class Reference

This component should be placed on the "ARLocationRoot" GameObject (which should be a child of the "A↩ R Session Origin") for correctly aligning the coordinate system to the north/east geographical lines.

Inheritance diagram for ARLocation.ARLocationOrientation:



Classes

- class OnBeforeOrientationUpdatedEvent

Public Member Functions

- void Restart ()

  Restarts the orientation tracking.

Public Attributes

- uint **MaxNumberOfUpdates** = 4
- int AverageCount = 250

  Only update after measuring the heading N times, and take the average.
- double **LowPassFilterFactor** = 0.68f
- bool UseRawUntilFirstAverage = true

  If set to true, use raw heading values until measuring the first average.
- float MovementSmoothingFactor = 0.1f

  The smoothing factor. Zero means disabled. Values around 100 seem to give good results.
- float TrueNorthOffset

  A custom offset to the device-calculated true north direction.
- bool **ApplyCompassTiltCompensationOnAndroid** = true
- UnityEvent **OnOrientationUpdated**
- OnBeforeOrientationUpdatedEvent **OnBeforeOrientationUpdated**

Additional Inherited Members

6.9.1 Detailed Description

This component should be placed on the "ARLocationRoot" GameObject (which should be a child of the "AR Session Origin") for correctly aligning the coordinate system to the north/east geographical lines.

Definition at line 17 of file ARLocationOrientation.cs.

6.9.2 Member Function Documentation

6.9.2.1 Restart()

```
void ARLocation.ARLocationOrientation.Restart ( )
```

Restarts the orientation tracking.

Definition at line 87 of file ARLocationOrientation.cs.

6.9.3 Member Data Documentation

6.9.3.1 AverageCount

```
int ARLocation.ARLocationOrientation.AverageCount = 250
```

Only update after measuring the heading N times, and take the average.

Definition at line 34 of file ARLocationOrientation.cs.

6.9.3.2 MovementSmoothingFactor

```
float ARLocation.ARLocationOrientation.MovementSmoothingFactor = 0.1f
```

The smoothing factor. Zero means disabled. Values around 100 seem to give good results.

Definition at line 53 of file ARLocationOrientation.cs.

6.9.3.3 TrueNorthOffset

```
float ARLocation.ARLocationOrientation.TrueNorthOffset
```

A custom offset to the device-calculated true north direction.

Definition at line 61 of file ARLocationOrientation.cs.

6.9.3.4 UseRawUntilFirstAverage

```
bool ARLocation.ARLocationOrientation.UseRawUntilFirstAverage = true
```

If set to true, use raw heading values until measuring the first average.

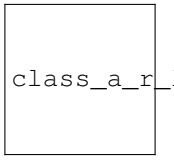
Definition at line 45 of file ARLocationOrientation.cs.

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

- Assets/ARLocation/Scripts/Components/ARLocationOrientation.cs

6.10 ARLocation.ARLocationOrientationInspector Class Reference

Inheritance diagram for ARLocation.ARLocationOrientationInspector:



Public Member Functions

- override void **OnInspectorGUI** ()

6.10.1 Detailed Description

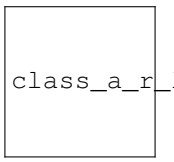
Definition at line 6 of file ARLocationOrientationInspector.cs.

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

- Assets/ARLocation/Editor/ARLocationOrientationInspector.cs

6.11 ARLocation.ARLocationProvider Class Reference

Inheritance diagram for ARLocation.ARLocationProvider:



Classes

- class CompassUpdatedUnityEvent
- class LocationEnabledUnityEvent
- class LocationUpdatedUnityEvent

Public Member Functions

- override void **Awake** ()
- void ForceLocationUpdate ()

  Force the provider to emit a location update event. This wont force a new read of location, just emit the last available measurement.
- void Pause ()

  Pauses location updates
- void Resume ()

  Resumes location updates
- void Restart ()

  Resets the location provider.
- void OnLocationUpdatedEvent (LocationUpdatedDelegate locationUpdatedDelegate)

  Register a delegate to location updates.
- void OnCompassUpdatedEvent (CompassUpdateDelegate compassUpdateDelegate)

  Register a delegate to compass/heading updates.
- void OnEnabledEvent (LocationEnabledDelegate del)

  RegisterRegister delegate for when the provider enables location updates.
- void OnFailedEvent (LocationFailedDelegate del)

  Register a delegate for when the provider fails to initialize location services.

Public Attributes

- LocationProviderOptions **LocationProviderSettings** = new LocationProviderOptions()
- LocationData **MockLocationData**
- uint **MaxWaitTime** = 20
- uint **StartUpDelay**
- bool **DebugMode**
- LocationEnabledUnityEvent **OnEnabled**
- LocationUpdatedUnityEvent **OnLocationUpdated**
- LocationUpdatedUnityEvent **OnRawLocationUpdated**
- CompassUpdatedUnityEvent **OnCompassUpdated**
- bool IsEnabled => Provider.IsEnabled

If true, the location provider has received the first location data.

- bool HasStarted => Provider.HasStarted

  If true, the location provider has started, but no location data has been read.
- int LocationUpdateCount => Provider.LocationUpdateCount

  The number of location updates so far.
- bool IsPaused => Provider.Paused

  If true, updates are paused.
- LocationReading CurrentLocation => Provider.CurrentLocation

  The latest location data.
- LocationReading LastLocation => Provider.LastLocation

  The previous location data.
- HeadingReading CurrentHeading => Provider.CurrentHeading

  The current heading data.
- float TimeSinceStart => Time.time - Provider.StartTime

  Time since the location provider has started.
- double DistanceFromStartPoint => Provider.DistanceFromStartPoint

  The distance from the initial measured position.

Properties

- ILocationProvider Provider `[get]`

  Returns the current location provider.

6.11.1 Detailed Description

Definition at line 15 of file ARLocationProvider.cs.

6.11.2 Member Function Documentation

6.11.2.1 ForceLocationUpdate()

```
void ARLocation.ARLocationProvider.ForceLocationUpdate ( )
```

Force the provider to emit a location update event. This wont force a new read of location, just emit the last available measurement.

Definition at line 208 of file ARLocationProvider.cs.

6.11.2.2 OnCompassUpdatedEvent()

```
void ARLocation.ARLocationProvider.OnCompassUpdatedEvent (
            CompassUpdateDelegate compassUpdateDelegate )
```

Register a delegate to compass/heading updates.

Parameters

compassUpdateDelegate	

Definition at line 269 of file ARLocationProvider.cs.

6.11.2.3 OnEnabledEvent()

```
void ARLocation.ARLocationProvider.OnEnabledEvent (
            LocationEnabledDelegate del )
```

RegisterRegister delegate for when the provider enables location updates.

Parameters

del	Del.

Definition at line 278 of file ARLocationProvider.cs.

6.11.2.4 OnFailedEvent()

```
void ARLocation.ARLocationProvider.OnFailedEvent (
            LocationFailedDelegate del )
```

Register a delegate for when the provider fails to initialize location services.

Parameters

del	Del.

Definition at line 287 of file ARLocationProvider.cs.

6.11.2.5 OnLocationUpdatedEvent()

```
void ARLocation.ARLocationProvider.OnLocationUpdatedEvent (
            LocationUpdatedDelegate locationUpdatedDelegate )
```

Register a delegate to location updates.

Parameters

locationUpdatedDelegate	

Definition at line 255 of file ARLocationProvider.cs.

6.11.2.6 Pause()

```
void ARLocation.ARLocationProvider.Pause ( )
```

Pauses location updates

Definition at line 227 of file ARLocationProvider.cs.

6.11.2.7 Restart()

```
void ARLocation.ARLocationProvider.Restart ( )
```

Resets the location provider.

Definition at line 245 of file ARLocationProvider.cs.

6.11.2.8 Resume()

```
void ARLocation.ARLocationProvider.Resume ( )
```

Resumes location updates

Definition at line 236 of file ARLocationProvider.cs.

6.11.3 Member Data Documentation

6.11.3.1 CurrentHeading

```
HeadingReading ARLocation.ARLocationProvider.CurrentHeading => Provider.CurrentHeading
```

The current heading data.

Definition at line 97 of file ARLocationProvider.cs.

6.11.3.2 CurrentLocation

```
LocationReading ARLocation.ARLocationProvider.CurrentLocation => Provider.CurrentLocation
```

The latest location data.

Definition at line 87 of file ARLocationProvider.cs.

6.11.3.3 DistanceFromStartPoint

```
double ARLocation.ARLocationProvider.DistanceFromStartPoint => Provider.DistanceFromStartPoint
```

The distance from the initial measured position.

Definition at line 108 of file ARLocationProvider.cs.

6.11.3.4 HasStarted

```
bool ARLocation.ARLocationProvider.HasStarted => Provider.HasStarted
```

If true, the location provider has started, but no location data has been read.

Definition at line 72 of file ARLocationProvider.cs.

6.11.3.5 IsEnabled

```
bool ARLocation.ARLocationProvider.IsEnabled => Provider.IsEnabled
```

If true, the location provider has received the first location data.

Definition at line 67 of file ARLocationProvider.cs.

6.11.3.6 IsPaused

```
bool ARLocation.ARLocationProvider.IsPaused => Provider.Paused
```

If true, updates are paused.

Definition at line 82 of file ARLocationProvider.cs.

6.11.3.7 LastLocation

```
LocationReading ARLocation.ARLocationProvider.LastLocation => Provider.LastLocation
```

The previous location data.

Definition at line 92 of file ARLocationProvider.cs.

6.11.3.8 LocationUpdateCount

```
int ARLocation.ARLocationProvider.LocationUpdateCount => Provider.LocationUpdateCount
```

The number of location updates so far.

Definition at line 77 of file ARLocationProvider.cs.

6.11.3.9 TimeSinceStart

```
float ARLocation.ARLocationProvider.TimeSinceStart => Time.time - Provider.StartTime
```

Time since the location provider has started.

Definition at line 103 of file ARLocationProvider.cs.

6.11.4 Property Documentation

6.11.4.1 Provider

```
ILocationProvider ARLocation.ARLocationProvider.Provider  [get]
```

Returns the current location provider.

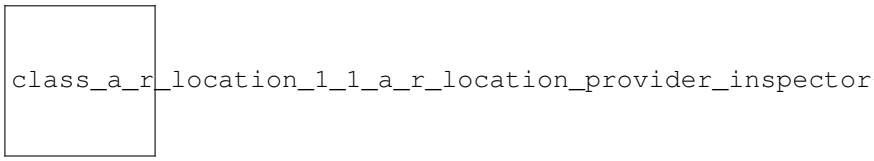
Definition at line 62 of file ARLocationProvider.cs.

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

- Assets/ARLocation/Scripts/Components/ARLocationProvider.cs

6.12 ARLocation.ARLocationProviderInspector Class Reference

Inheritance diagram for ARLocation.ARLocationProviderInspector:



Public Member Functions

- override void **OnInspectorGUI** ()

6.12.1 Detailed Description

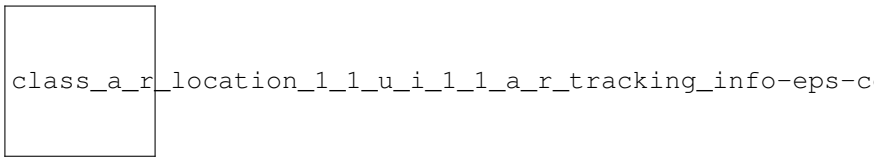
Definition at line 7 of file ARLocationProviderInspector.cs.

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

- Assets/ARLocation/Editor/ARLocationProviderInspector.cs

6.13 ARLocation.UI.ARTrackingInfo Class Reference

Inheritance diagram for ARLocation.UI.ARTrackingInfo:



Public Attributes

- Text **InfoValue**
- Text **ProviderValue**

6.13.1 Detailed Description

Definition at line 7 of file ARTrackingInfo.cs.

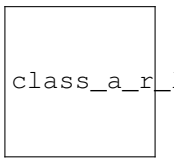
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Components/UI/ARTrackingInfo.cs

6.14 ARLocation.CatmullRomCurve Class Reference

A catmull-rom curve.

Inheritance diagram for ARLocation.CatmullRomCurve:



Public Member Functions

- CatmullRomCurve (Vector3 p0, Vector3 p1, Vector3 p2, Vector3 p3, float alpha)

  Creates a catmull-rom curve with control points p0, p1, p2 and p3, and with a given alpha/tension parameter.
- override Vector3 GetPoint (float u)

  Calculates the curve at a point u, where u is between 0 and 1.
- override CurvePointData GetPointAndTangent (float u)

  Calculates the point and the tangent of the curve.
- override Vector3 [ ] Sample (int n)

  Creates a sample of (N+2) points (i.e., N + start and end points) of the current curve. Also calculates the length estimate.
- override float EstimateLength (int n=100)

  Returns the estimated length.
- override float GetParameterForLength (float s)

  Gets the curve parameter for a given length.
- override Vector3 GetPointAtLength (float s)

  Gets the curve point at a given length.
- override CurvePointData GetPointAndTangentAtLength (float s)

  Gets the CurvePointData which stores the point and tangent at a given arc-length.

Properties

- float **T0** `[get]`
- float **T1** `[get]`
- float **T2** `[get]`
- float **T3** `[get]`
- float Alpha `[get, set]`

  Gets or sets the alpha.
- Vector3 **P0** `[get, set]`
- Vector3 **P1** `[get, set]`
- Vector3 **P2** `[get, set]`
- Vector3 **P3** `[get, set]`

6.14.1 Detailed Description

A catmull-rom curve.

Definition at line 8 of file CatmullRomCurve.cs.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 CatmullRomCurve()

```
ARLocation.CatmullRomCurve.CatmullRomCurve (
            Vector3 p0,
            Vector3 p1,
            Vector3 p2,
            Vector3 p3,
            float alpha )
```

Creates a catmull-rom curve with control points p0, p1, p2 and p3, and with a given alpha/tension parameter.

Parameters

p0	
p1	
p2	
p3	
alpha	

Definition at line 144 of file CatmullRomCurve.cs.

6.14.3 Member Function Documentation

6.14.3.1 EstimateLength()

```
override float ARLocation.CatmullRomCurve.EstimateLength (
            int n = 100 )  [virtual]
```

Returns the estimated length.

Returns

The length.

Parameters

n	N.

Implements ARLocation.Curve.

Definition at line 269 of file CatmullRomCurve.cs.

6.14.3.2 GetParameterForLength()

```
override float ARLocation.CatmullRomCurve.GetParameterForLength (
            float s )  [virtual]
```

Gets the curve parameter for a given length.

Returns

The parameter for length.

Parameters

s	S.

Implements ARLocation.Curve.

Definition at line 284 of file CatmullRomCurve.cs.

6.14.3.3 GetPoint()

```
override Vector3 ARLocation.CatmullRomCurve.GetPoint (
            float u )  [virtual]
```

Calculates the curve at a point u, where u is between 0 and 1.

Parameters

u	The curve parameter in the [0, 1] interval.

Returns

Implements ARLocation.Curve.

Definition at line 172 of file CatmullRomCurve.cs.

6.14.3.4 GetPointAndTangent()

```
override CurvePointData ARLocation.CatmullRomCurve.GetPointAndTangent (
            float u )  [virtual]
```

Calculates the point and the tangent of the curve.

Parameters

u	The curve parameter in the [0, 1] interval.

Returns

Implements ARLocation.Curve.

Definition at line 194 of file CatmullRomCurve.cs.

6.14.3.5 GetPointAndTangentAtLength()

```
override CurvePointData ARLocation.CatmullRomCurve.GetPointAndTangentAtLength (
            float s )  [virtual]
```

Gets the CurvePointData which stores the point and tangent at a given arc-length.

Parameters

s	

Returns

Implements ARLocation.Curve.

Definition at line 319 of file CatmullRomCurve.cs.

6.14.3.6 GetPointAtLength()

```
override Vector3 ARLocation.CatmullRomCurve.GetPointAtLength (
            float s )  [virtual]
```

Gets the curve point at a given length.

Returns

The point at length.

Parameters

s	S.

Implements ARLocation.Curve.

Definition at line 308 of file CatmullRomCurve.cs.

6.14.3.7 Sample()

```
override Vector3 [] ARLocation.CatmullRomCurve.Sample (
            int n )  [virtual]
```

Creates a sample of (N+2) points (i.e., N + start and end points) of the current curve. Also calculates the length estimate.

Returns

The sample.

Parameters

n	N.

Implements ARLocation.Curve.

Definition at line 230 of file CatmullRomCurve.cs.

6.14.4 Property Documentation

6.14.4.1 Alpha

```
float ARLocation.CatmullRomCurve.Alpha  [get], [set]
```

Gets or sets the alpha.

The alpha.

Definition at line 61 of file CatmullRomCurve.cs.

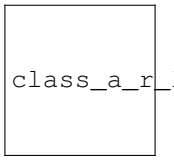
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Math/CatmullRomCurve.cs

6.15 ARLocation.CatmullRomSpline Class Reference

A (open-ended) catmull-rom spline, which interpolates a set points by joining catmull-rom curves together.

Inheritance diagram for ARLocation.CatmullRomSpline:



Public Member Functions

- CatmullRomSpline (Vector3[] points, int n, float alpha)

  Creates a new Catmull-rom spline.
- sealed override void CalculateSegments (int n)

  Calculate the catmull-rom segments. Also estimates the curve's length.

Properties

- float Alpha `[get, set]`

  The alpha/tension parameter of the spline.

Additional Inherited Members

6.15.1 Detailed Description

A (open-ended) catmull-rom spline, which interpolates a set points by joining catmull-rom curves together.

Definition at line 10 of file CatmullRomSpline.cs.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 CatmullRomSpline()

```
ARLocation.CatmullRomSpline.CatmullRomSpline (
            Vector3 [] points,
            int n,
            float alpha )
```

Creates a new Catmull-rom spline.

Parameters

points	The interpolated points.
n	The number of samples used in each segment of the spline.
alpha	

Definition at line 48 of file CatmullRomSpline.cs.

6.15.3 Member Function Documentation

6.15.3.1 CalculateSegments()

```
sealed override void ARLocation.CatmullRomSpline.CalculateSegments (
            int n )  [virtual]
```

Calculate the catmull-rom segments. Also estimates the curve's length.

Parameters

n	The number sample points used to estimate each segment's length.

Implements ARLocation.Spline.

Definition at line 60 of file CatmullRomSpline.cs.

6.15.4 Property Documentation

6.15.4.1 Alpha

```
float ARLocation.CatmullRomSpline.Alpha  [get], [set]
```

The alpha/tension parameter of the spline.

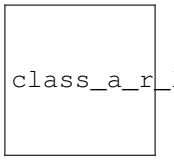
Definition at line 27 of file CatmullRomSpline.cs.

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

- Assets/ARLocation/Scripts/Math/CatmullRomSpline.cs

6.16 ARLocation.ARLocationProvider.CompassUpdatedUnityEvent Class Reference

Inheritance diagram for ARLocation.ARLocationProvider.CompassUpdatedUnityEvent:



6.16.1 Detailed Description

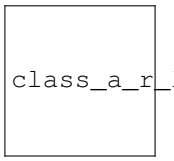
Definition at line 23 of file ARLocationProvider.cs.

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

- Assets/ARLocation/Scripts/Components/ARLocationProvider.cs

6.17 ARLocation.Utils.CreatePointOfInterestTextMeshes Class Reference

Inheritance diagram for ARLocation.Utils.CreatePointOfInterestTextMeshes:



Public Member Functions

- string **GetNodeTagValue** (XmlNode node, string tagName)
- string **GetNodeName** (XmlNode node)

Public Attributes

- float **height** = 1f
- TextMesh **textPrefab**
- float **movementSmoothingFactor** = 100.0f
- Location [ ] **locations**
- OpenStreetMapOptions **openStreetMapOptions**

6.17.1 Detailed Description

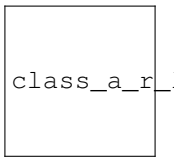
Definition at line 40 of file CreatePointOfInterestTextMeshes.cs.

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

- Assets/ARLocation/Scripts/Utils/CreatePointOfInterestTextMeshes.cs

6.18 ARLocation.Curve Class Reference

Inheritance diagram for ARLocation.Curve:



Public Member Functions

- abstract Vector3 **GetPoint** (float u)
- abstract CurvePointData **GetPointAndTangent** (float u)
- abstract Vector3 [ ] **Sample** (int n)
- abstract float **EstimateLength** (int n=100)
- abstract float **GetParameterForLength** (float s)
- abstract Vector3 **GetPointAtLength** (float s)
- abstract CurvePointData **GetPointAndTangentAtLength** (float s)

6.18.1 Detailed Description

Definition at line 16 of file Curve.cs.

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

- Assets/ARLocation/Scripts/Math/Curve.cs

6.19 ARLocation.CurvePointData Struct Reference

A struct holding a pair of point/tangent values.

Public Attributes

- Vector3 **point**
- Vector3 **tangent**

6.19.1 Detailed Description

A struct holding a pair of point/tangent values.

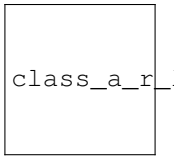
Definition at line 9 of file Curve.cs.

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

- Assets/ARLocation/Scripts/Math/Curve.cs

6.20 ARLocation.Utils.DebugCanvas Class Reference

Inheritance diagram for ARLocation.Utils.DebugCanvas:



Public Member Functions

- void **SetDebugText** (string val)

6.20.1 Detailed Description

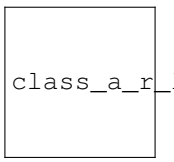
Definition at line 9 of file DebugCanvas.cs.

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

- Assets/ARLocation/Scripts/Utils/DebugCanvas.cs

6.21 ARLocation.UI.DebugInfoOverlay Class Reference

Inheritance diagram for ARLocation.UI.DebugInfoOverlay:



Public Member Functions

- void **Toggle** ()
- void **ToggleObjectInfo** ()

Public Attributes

- bool **Show**
- bool **ShowObjectInfo**

6.21.1 Detailed Description

Definition at line 7 of file DebugInfoOverlay.cs.

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

- Assets/ARLocation/Scripts/Components/UI/DebugInfoOverlay.cs

6.22 DefineSymbols Class Reference

Utility class to manage a list of symbol strings.

Public Member Functions

- **DefineSymbols** (string symbols)
- void **Set** (string sym)
- bool **Has** (string symbol)
- void **Add** (string symbol)
- void **Remove** (string symbol)
- string **Get** ()

6.22.1 Detailed Description

Utility class to manage a list of symbol strings.

Definition at line 7 of file DefineSymbols.cs.

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

- Assets/ARLocation/Editor/DefineSymbols.cs

6.23 DefineSymbolsManager Class Reference

Utility class that manages Define Symbols for a given set of build targets.

Public Member Functions

- **DefineSymbolsManager** (BuildTargetGroup[ ] groups)
- void **UpdateFromBuildSettings** ()
- void **ApplyToBuildSettings** ()
- void **Add** (string symbol)
- void **Remove** (string symbol)
- bool **Has** (string symbol)
- override string **ToString** ()

6.23.1 Detailed Description

Utility class that manages Define Symbols for a given set of build targets.

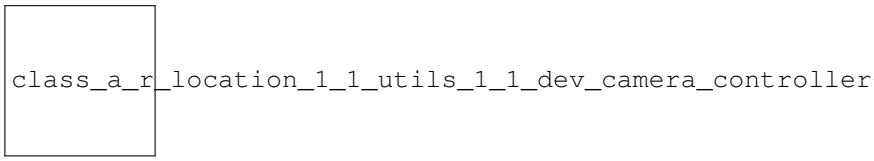
Definition at line 8 of file DefineSymbolsManager.cs.

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

- Assets/ARLocation/Editor/DefineSymbolsManager.cs

6.24 ARLocation.Utils.DevCameraController Class Reference

Inheritance diagram for ARLocation.Utils.DevCameraController:



Public Attributes

- float MouseSensitivity = 1.0f

  The mouse look/rotation sensitivity.
- float Speed = 1.0f

  The walking speed

6.24.1 Detailed Description

Definition at line 7 of file DevCameraController.cs.

6.24.2 Member Data Documentation

6.24.2.1 MouseSensitivity

```
float ARLocation.Utils.DevCameraController.MouseSensitivity = 1.0f
```

The mouse look/rotation sensitivity.

Definition at line 12 of file DevCameraController.cs.

6.24.2.2 Speed

```
float ARLocation.Utils.DevCameraController.Speed = 1.0f
```

The walking speed

Definition at line 17 of file DevCameraController.cs.

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

- Assets/ARLocation/Scripts/Utils/DevCameraController.cs

6.25 ARLocation.DVector2 Struct Reference

Public Member Functions

- DVector2 **Clone** ()
- DVector2 (double x=0.0, double y=0.0)

  Initializes a new instance of the T:DVector2 struct.
- Vector2 toVector2 ()

  Converts to a Vector2.
- bool Equals (DVector2 v, double e=0.00005)

  Equals the specified v and e.
- void Normalize ()

  Normalize this instance.
- void Set (double newX=0.0, double newY=0.0)

  Set the specified x and y.
- override string ToString ()

  Returns a T:System.String that represents the current T:DVector2.

Static Public Member Functions

- static double Dot (DVector2 a, DVector2 b)

  Dot the specified a and b.
- static double Distance (DVector2 a, DVector2 b)

  Distance the specified a and b.
- static DVector2 Lerp (DVector2 a, DVector2 b, double t)

  Lerp the specified a, b and t.
- static DVector2 operator* (DVector2 a, double b)

  Computes the product of `a` and `b`, yielding a new T:DVector2.
- static DVector2 operator/ (DVector2 a, double b)

  Computes the division of `a` and `b`, yielding a new T:DVector2.
- static DVector2 operator+ (DVector2 a, DVector2 b)

  Adds a DVector2 to a DVector2, yielding a new T:DVector2.
- static DVector2 operator- (DVector2 a, DVector2 b)

  Subtracts a DVector2 from a DVector2, yielding a new T:DVector2.

Public Attributes

- double **x**
- double **y**

Properties

- double magnitude [get]

  Gets the magnitude of the vector.
- DVector2 normalized [get]

  Gets the normalized version of this vector.

6.25.1 Detailed Description

Definition at line 8 of file DVector2.cs.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 DVector2()

```
ARLocation.DVector2.DVector2 (
            double x = 0.0,
            double y = 0.0 )
```

Initializes a new instance of the T:DVector2 struct.

Parameters

x	The x coordinate.
y	The y coordinate.

Definition at line 54 of file DVector2.cs.

6.25.3 Member Function Documentation

6.25.3.1 Distance()

```
static double ARLocation.DVector2.Distance (
            DVector2 a,
            DVector2 b )  [static]
```

Distance the specified a and b.

Returns

The distance.

Parameters

a	The alpha component.
b	The blue component.

Definition at line 127 of file DVector2.cs.

6.25.3.2 Dot()

```
static double ARLocation.DVector2.Dot (
            DVector2 a,
            DVector2 b )  [static]
```

Dot the specified a and b.

Returns

The dot.

Parameters

a	The alpha component.
b	The blue component.

Definition at line 116 of file DVector2.cs.

6.25.3.3 Equals()

```
bool ARLocation.DVector2.Equals (
            DVector2 v,
            double e = 0.00005 )
```

Equals the specified v and e.

Returns

The equals.

Parameters

v	V.
e	E.

Definition at line 75 of file DVector2.cs.

6.25.3.4 Lerp()

```
static DVector2 ARLocation.DVector2.Lerp (
            DVector2 a,
            DVector2 b,
            double t )  [static]
```

Lerp the specified a, b and t.

Returns

The lerp.

Parameters

a	The alpha component.
b	The blue component.
t	T.

Definition at line 139 of file DVector2.cs.

6.25.3.5 Normalize()

```
void ARLocation.DVector2.Normalize ( )
```

Normalize this instance.

Definition at line 83 of file DVector2.cs.

6.25.3.6 operator∗()

```
static DVector2 ARLocation.DVector2.operator* (
            DVector2 a,
            double b )  [static]
```

Computes the product of `a` and `b`, yielding a new T:DVector2.

Parameters

a	The DVector2 to multiply.
b	The double to multiply.

Returns

The T:DVector2 that is the `a * b`.

Definition at line 151 of file DVector2.cs.

6.25.3.7 operator+()

```
static DVector2 ARLocation.DVector2.operator+ (
            DVector2 a,
            DVector2 b )  [static]
```

Adds a DVector2 to a DVector2, yielding a new T:DVector2.

Parameters

a	The first DVector2 to add.
b	The second DVector2 to add.

Returns

The T:DVector2 that is the sum of the values of `a` and `b`.

Definition at line 179 of file DVector2.cs.

6.25.3.8 operator-()

```
static DVector2 ARLocation.DVector2.operator- (
            DVector2 a,
            DVector2 b )  [static]
```

Subtracts a DVector2 from a DVector2, yielding a new T:DVector2.

Parameters

a	The DVector2 to subtract from (the minuend).
b	The DVector2 to subtract (the subtrahend).

Returns

The T:DVector2 that is the `a` minus `b`.

Definition at line 193 of file DVector2.cs.

6.25.3.9 operator/()

```
static DVector2 ARLocation.DVector2.operator/ (
            DVector2 a,
            double b )  [static]
```

Computes the division of a and b, yielding a new T:DVector2.

Parameters

a	The DVector2 to divide (the divident).
b	The double to divide (the divisor).

Returns

The T:DVector2 that is the a / b.

Definition at line 165 of file DVector2.cs.

6.25.3.10 Set()

```
void ARLocation.DVector2.Set (
            double newX = 0.0,
            double newY = 0.0 )
```

Set the specified x and y.

Parameters

newX	
newY	

Definition at line 95 of file DVector2.cs.

6.25.3.11 ToString()

```
override string ARLocation.DVector2.ToString ( )
```

Returns a T:System.String that represents the current T:DVector2.

Returns

A T:System.String that represents the current T:DVector2.

Definition at line 105 of file DVector2.cs.

6.25.3.12 toVector2()

```
Vector2 ARLocation.DVector2.toVector2 ( )
```

Converts to a Vector2.

Returns

The vector2.

Definition at line 64 of file DVector2.cs.

6.25.4 Property Documentation

6.25.4.1 magnitude

```
double ARLocation.DVector2.magnitude  [get]
```

Gets the magnitude of the vector.

The magnitude.

Definition at line 18 of file DVector2.cs.

6.25.4.2 normalized

```
DVector2 ARLocation.DVector2.normalized  [get]
```

Gets the normalized version of this vector.

The normalized.

Definition at line 30 of file DVector2.cs.

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

- Assets/ARLocation/Scripts/Math/DVector2.cs

6.26 ARLocation.DVector3 Struct Reference

Public Member Functions

- **DVector3** (Vector3 v)
- DVector3 (double newX=0.0, double newY=0.0, double newZ=0.0)

  Initializes a new instance of the T:DVector3 struct.
- Vector3 toVector3 ()

  Converts to a Vector3.
- bool Equals (DVector3 v, double e=0.00005)

  Equals the specified v and e.
- void Normalize ()

  Normalize this instance.
- void Set (double xx=0.0, double yy=0.0, double zz=0.0)

  Set the specified x and y.
- override string ToString ()

  Returns a T:System.String that represents the current T:DVector3.

Static Public Member Functions

- static double Dot (DVector3 a, DVector3 b)

  Dot the specified a and b.
- static double Distance (DVector3 a, DVector3 b)

  Distance the specified a and b.
- static DVector3 Lerp (DVector3 a, DVector3 b, double t)

  Lerp the specified a, b and t.
- static DVector3 operator∗ (DVector3 a, double b)

  Computes the product of `a` and `b`, yielding a new T:DVector3.
- static DVector3 operator∗ (double b, DVector3 a)

  Computes the product of `a` and `b`, yielding a new T:DVector3.
- static DVector3 operator/ (DVector3 a, double b)

  Computes the division of `a` and `b`, yielding a new T:DVector3.
- static DVector3 operator+ (DVector3 a, DVector3 b)

  Adds a DVector3 to a DVector3, yielding a new T:DVector3.
- static DVector3 operator- (DVector3 a, DVector3 b)

  Subtracts a DVector3 from a DVector3, yielding a new T:DVector3.

Public Attributes

- double **x**
- double **y**
- double **z**

Properties

- double magnitude `[get]`

  Gets the magnitude of the vector.
- DVector3 normalized `[get]`

  Gets the normalized version of this vector.

6.26.1 Detailed Description

Definition at line 8 of file DVector3.cs.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 DVector3()

```
ARLocation.DVector3.DVector3 (
            double newX = 0.0,
            double newY = 0.0,
            double newZ = 0.0 )
```

Initializes a new instance of the T:DVector3 struct.

Definition at line 55 of file DVector3.cs.

6.26.3 Member Function Documentation

6.26.3.1 Distance()

```
static double ARLocation.DVector3.Distance (
            DVector3 a,
            DVector3 b )  [static]
```

Distance the specified a and b.

Returns

The distance.

Parameters

a	The alpha component.
b	The blue component.

Definition at line 132 of file DVector3.cs.

6.26.3.2 Dot()

```
static double ARLocation.DVector3.Dot (
            DVector3 a,
            DVector3 b )  [static]
```

Dot the specified a and b.

Returns

The dot.

Parameters

a	The alpha component.
b	The blue component.

Definition at line 121 of file DVector3.cs.

6.26.3.3 Equals()

```
bool ARLocation.DVector3.Equals (
            DVector3 v,
            double e = 0.00005 )
```

Equals the specified v and e.

Returns

The equals.

Parameters

v	V.
e	E.

Definition at line 77 of file DVector3.cs.

6.26.3.4 Lerp()

```
static DVector3 ARLocation.DVector3.Lerp (
            DVector3 a,
            DVector3 b,
            double t )  [static]
```

Lerp the specified a, b and t.

Returns

The lerp.

Parameters

a	The alpha component.
b	The blue component.
t	T.

Definition at line 144 of file DVector3.cs.

6.26.3.5 Normalize()

```
void ARLocation.DVector3.Normalize ( )
```

Normalize this instance.

Definition at line 85 of file DVector3.cs.

6.26.3.6 operator∗() [1/2]

```
static DVector3 ARLocation.DVector3.operator* (
            DVector3 a,
            double b )  [static]
```

Computes the product of `a` and `b`, yielding a new T:DVector3.

Parameters

a	The DVector3 to multiply.
b	The double to multiply.

Returns

The T:DVector3 that is the `a * b`.

Definition at line 156 of file DVector3.cs.

6.26.3.7 operator∗() [2/2]

```
static DVector3 ARLocation.DVector3.operator* (
            double b,
            DVector3 a )  [static]
```

Computes the product of `a` and `b`, yielding a new T:DVector3.

Parameters

a	The DVector3 to multiply.
b	The double to multiply.

Returns

The T:DVector3 that is the `a * b`.

Definition at line 171 of file DVector3.cs.

6.26.3.8 operator+()

```
static DVector3 ARLocation.DVector3.operator+ (
            DVector3 a,
            DVector3 b )  [static]
```

Adds a DVector3 to a DVector3, yielding a new T:DVector3.

Parameters

a	The first DVector3 to add.
b	The second DVector3 to add.

Returns

The T:DVector3 that is the sum of the values of `a` and `b`.

Definition at line 201 of file DVector3.cs.

6.26.3.9 operator-()

```
static DVector3 ARLocation.DVector3.operator- (
            DVector3 a,
            DVector3 b )  [static]
```

Subtracts a DVector3 from a DVector3, yielding a new T:DVector3.

Parameters

a	The DVector3 to subtract from (the minuend).
b	The DVector3 to subtract (the subtrahend).

Returns

The T:DVector3 that is the `a` minus `b`.

Definition at line 216 of file DVector3.cs.

6.26.3.10 operator/()

```
static DVector3 ARLocation.DVector3.operator/ (
            DVector3 a,
            double b )  [static]
```

Computes the division of `a` and `b`, yielding a new T:DVector3.

Parameters

a	The DVector3 to divide (the divident).
b	The double to divide (the divisor).

Returns

The T:DVector3 that is the `a` / `b`.

Definition at line 186 of file DVector3.cs.

6.26.3.11 Set()

```
void ARLocation.DVector3.Set (
            double xx = 0.0,
            double yy = 0.0,
            double zz = 0.0 )
```

Set the specified x and y.

Parameters

xx	
yy	
zz	

Definition at line 99 of file DVector3.cs.

6.26.3.12 ToString()

```
override string ARLocation.DVector3.ToString ( )
```

Returns a T:System.String that represents the current T:DVector3.

Returns

A T:System.String that represents the current T:DVector3.

Definition at line 110 of file DVector3.cs.

6.26.3.13 toVector3()

```
Vector3 ARLocation.DVector3.toVector3 ( )
```

Converts to a Vector3.

Returns

The vector2.

Definition at line 66 of file DVector3.cs.

6.26.4 Property Documentation

6.26.4.1 magnitude

```
double ARLocation.DVector3.magnitude  [get]
```

Gets the magnitude of the vector.

The magnitude.

Definition at line 26 of file DVector3.cs.

6.26.4.2 normalized

```
DVector3 ARLocation.DVector3.normalized  [get]
```

Gets the normalized version of this vector.

The normalized.

Definition at line 38 of file DVector3.cs.

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

- Assets/ARLocation/Scripts/Math/DVector3.cs

6.27 ARLocation.PlaceAtLocations.Entry Class Reference

Public Attributes

- LocationData **ObjectLocation**
- OverrideAltitudeData **OverrideAltitude** = new OverrideAltitudeData()

6.27.1 Detailed Description

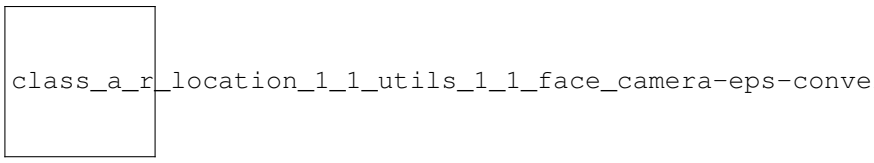
Definition at line 19 of file PlaceAtLocations.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocations.cs

6.28 ARLocation.Utils.FaceCamera Class Reference

Inheritance diagram for ARLocation.Utils.FaceCamera:



6.28.1 Detailed Description

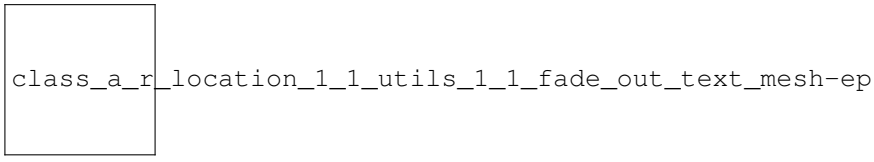
Definition at line 6 of file FaceCamera.cs.

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

- Assets/ARLocation/Scripts/Utils/FaceCamera.cs

6.29 ARLocation.Utils.FadeOutTextMesh Class Reference

Inheritance diagram for ARLocation.Utils.FadeOutTextMesh:



Public Attributes

- float **Duration** = 2.0f

6.29.1 Detailed Description

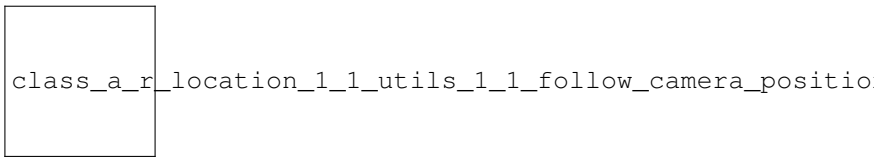
Definition at line 8 of file FadeOutTextMesh.cs.

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

- Assets/ARLocation/Scripts/Utils/FadeOutTextMesh.cs

6.30 ARLocation.Utils.FollowCameraPosition Class Reference

Inheritance diagram for ARLocation.Utils.FollowCameraPosition:



6.30.1 Detailed Description

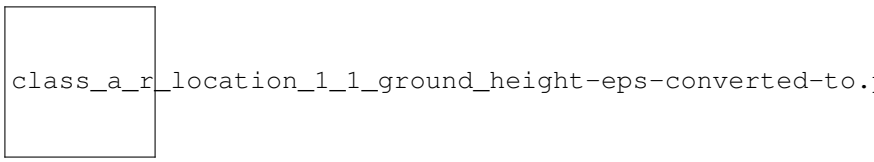
Definition at line 6 of file FollowCameraPosition.cs.

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

- Assets/ARLocation/Scripts/Utils/FollowCameraPosition.cs

6.31 ARLocation.GroundHeight Class Reference

Inheritance diagram for ARLocation.GroundHeight:



Classes

- class SettingsData
- class StateData

Public Attributes

- SettingsData **Settings** = new SettingsData()

6.31.1 Detailed Description

Definition at line 19 of file GroundHeight.cs.

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

- Assets/ARLocation/Scripts/Components/GroundHeight.cs

6.32 ARLocation.HeadingReading Struct Reference

Public Member Functions

- override string **ToString** ()

Public Attributes

- double **heading**
- double **magneticHeading**
- double **accuracy**
- long **timestamp**
- bool **isMagneticHeadingAvailable**

6.32.1 Detailed Description

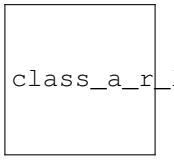
Definition at line 4 of file HeadingReading.cs.

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

- Assets/ARLocation/Scripts/Location/HeadingReading.cs

6.33 ARLocation.Hotspot Class Reference

Inheritance diagram for ARLocation.Hotspot:



Classes

- class HotspotSettingsData
- class OnHotspotActivatedUnityEvent
- class StateData

Public Types

- enum **PositionModes** { **HotspotCenter**, **CameraPosition** }

Public Member Functions

- void **Restart** ()

Static Public Member Functions

- static Hotspot **AddHotspotComponent** (GameObject go, Location location, HotspotSettingsData settings)
- static GameObject **CreateHotspotGameObject** (Location location, HotspotSettingsData settings, string name="GPS Hotspot")

Public Attributes

- PlaceAtLocation.LocationSettingsData **LocationSettings** = new PlaceAtLocation.LocationSettingsData()
- HotspotSettingsData **HotspotSettings** = new HotspotSettingsData()
- bool **DebugMode**
- OnHotspotActivatedUnityEvent **OnHotspotActivated**
- GameObject **Instance** => state.Instance
- float CurrentDistance => (float) currentDistance

  Returns the current user distance to the Hotspot center.

Properties

- Location **Location** `[get, set]`

6.33.1 Detailed Description

Definition at line 10 of file Hotspot.cs.

6.33.2 Member Data Documentation

6.33.2.1 CurrentDistance

```
float ARLocation.Hotspot.CurrentDistance => (float) currentDistance
```

Returns the current user distance to the Hotspot center.

Definition at line 80 of file Hotspot.cs.

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

- Assets/ARLocation/Scripts/Components/Hotspot.cs

6.34 ARLocation.Hotspot.HotspotSettingsData Class Reference

Public Attributes

- GameObject **Prefab**
- PositionModes **PositionMode**
- float **ActivationRadius** = 4.0f
- bool **AlignToCamera** = true
- float **DistanceFromCamera** = 3.0f

6.34.1 Detailed Description

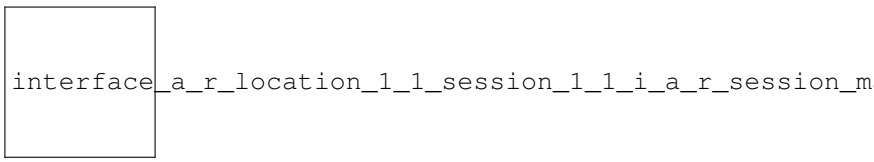
Definition at line 22 of file Hotspot.cs.

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

- Assets/ARLocation/Scripts/Components/Hotspot.cs

6.35 ARLocation.Session.IARSessionManager Interface Reference

Inheritance diagram for ARLocation.Session.IARSessionManager:



Public Member Functions

- void **Reset** (Action callback)
- string **GetSessionInfoString** ()
- string **GetProviderString** ()
- void **OnARTrackingStarted** (Action callback)
- void **OnARTrackingRestored** (Action callback)
- void **OnARTrackingLost** (Action callback)

Properties

- bool **DebugMode** [get, set]

6.35.1 Detailed Description

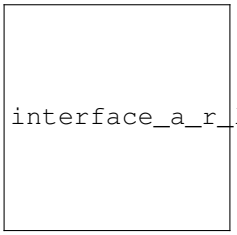
Definition at line 5 of file IARSessionManager.cs.

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

- Assets/ARLocation/Scripts/ARSession/IARSessionManager.cs

6.36 ARLocation.ILocationProvider Interface Reference

Inheritance diagram for ARLocation.ILocationProvider:



Public Member Functions

- IEnumerator **Start** (uint maxWaitTime=10000, uint delay=0)
- void **ForceLocationUpdate** ()
- void **Pause** ()
- void **Resume** ()
- void **Update** ()
- void **Restart** ()
- void **OnEnabled** (LocationEnabledDelegate del)
- void **OnFail** (LocationFailedDelegate del)
- void **SetCompassLowPassFactor** (double factor)
- string **GetInfoString** ()
- string **GetStatusString** ()

Properties

- string **Name** [get]
- LocationProviderOptions **Options** [get, set]
- LocationReading **CurrentLocation** [get]
- LocationReading **CurrentLocationRaw** [get]
- LocationReading **LastLocation** [get]
- LocationReading **LastLocationRaw** [get]
- LocationReading **FirstLocation** [get]
- HeadingReading **CurrentHeading** [get]
- HeadingReading **LastHeading** [get]
- float **StartTime** [get]
- bool **IsCompassEnabled** [get]
- double **DistanceFromStartPoint** [get]
- bool **IsEnabled** [get]
- bool **Paused** [get]
- int **LocationUpdateCount** [get]
- bool **HasStarted** [get]
- bool **ApplyCompassTiltCompensationOnAndroid** [get, set]

Events

- LocationUpdatedDelegate **LocationUpdated**
- LocationUpdatedDelegate **LocationUpdatedRaw**
- CompassUpdateDelegate **CompassUpdated**
- LocationEnabledDelegate **LocationEnabled**
- LocationFailedDelegate **LocationFailed**

6.36.1 Detailed Description

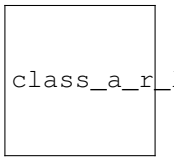
Definition at line 51 of file ILocationProvider.cs.

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

- Assets/ARLocation/Scripts/Location/ILocationProvider.cs

6.37 ARLocation.Line Class Reference

Inheritance diagram for ARLocation.Line:



Public Member Functions

- **Line** (Vector3 p0, Vector3 p1)
- void **Calculate** ()
- override float **EstimateLength** (int n=100)
- override float **GetParameterForLength** (float s)
- override Vector3 **GetPoint** (float u)
- Vector3 **GetTangent** (float u)
- override CurvePointData **GetPointAndTangent** (float u)
- override CurvePointData **GetPointAndTangentAtLength** (float s)
- override Vector3 **GetPointAtLength** (float s)
- override Vector3 [ ] **Sample** (int n)

Properties

- Vector3 **P0** `[get, set]`
- Vector3 **P1** `[get, set]`

6.37.1 Detailed Description

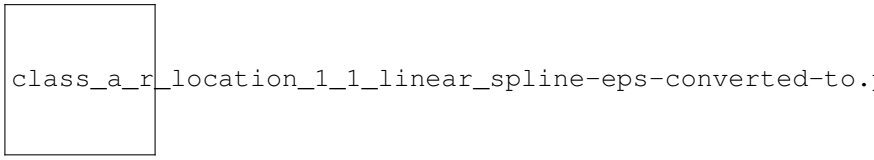
Definition at line 5 of file Line.cs.

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

- Assets/ARLocation/Scripts/Math/Line.cs

6.38 ARLocation.LinearSpline Class Reference

Inheritance diagram for ARLocation.LinearSpline:



Public Member Functions

- **LinearSpline** (Vector3[ ] points)
- override void CalculateSegments (int n)

  Calculate the catmull-rom segments. Also estimates the curve's length.

Additional Inherited Members

6.38.1 Detailed Description

Definition at line 5 of file LinearSpline.cs.

6.38.2 Member Function Documentation

6.38.2.1 CalculateSegments()

```
override void ARLocation.LinearSpline.CalculateSegments (
            int n )  [virtual]
```

Calculate the catmull-rom segments. Also estimates the curve's length.

Parameters

n	The number sample points used to estimate each segment's length.

Implements ARLocation.Spline.

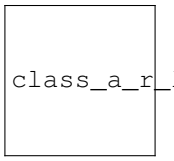
Definition at line 14 of file LinearSpline.cs.

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

- Assets/ARLocation/Scripts/Math/LinearSpline.cs

6.39 ARLocation.UI.LoadingBar Class Reference

Inheritance diagram for ARLocation.UI.LoadingBar:



Public Attributes

- float **FillPercentage** = 0.4f
- Color **StartColor** = Color.green
- Color **MiddleColor** = Color.yellow
- Color **EndColor** = Color.red
- Color **TextColor** = Color.blue
- bool **UsePercentageText**
- string **Text** = "100"

6.39.1 Detailed Description

Definition at line 7 of file LoadingBar.cs.

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

- Assets/ARLocation/Scripts/Components/UI/LoadingBar.cs

6.40 ARLocation.Location Class Reference

Represents a geographical location.

Public Member Functions

- **Location** (double latitude=0.0, double longitude=0.0, double altitude=0.0)
- Location Clone ()

  Clones this instance.
- override string **ToString** ()
- DVector3 **ToDVector3** ()
- Vector3 **ToVector3** ()

Static Public Member Functions

- static double HorizontalDistance (Location l1, Location l2)

  Calculates the horizontal distance according to the current function set in the configuration.
- static double PlaneSphericalDistance (Location l1, Location l2)

  Horizontal distance using spherical projection on a plane. https://en.wikipedia.org/wiki/↩Geographical_distance
- static double PlaneEllipsoidalFccDistance (Location l1, Location l2)

  Horizontal distance using ellipsoidal projection on a plane. https://en.wikipedia.org/wiki/↩Geographical_distance
- static double HaversineDistance (Location l1, Location l2)

  Horizontal distance, using the Haversine formula. https://stackoverflow.com/questions/41621957/a-more-effici
- static double DistanceWithAltitude (Location l1, Location l2)

  Calculates the full distance between locations, taking altitude into account.
- static DVector2 HorizontalVectorFromTo (Location l1, Location l2)

  Calculates the horizontal vector pointing from l1 to l2, in meters.
- static DVector3 VectorFromTo (Location l1, Location l2, bool ignoreHeight=false)

  Calculates the vector from l1 to l2, in meters, taking altitude into account.
- static Vector3 GetGameObjectPositionForLocation (Transform arLocationRoot, Vector3 userPosition, Location userLocation, Location objectLocation, bool heightIsRelative)

  Gets the game object world-position for location.
- static Vector3 GetGameObjectPositionForLocation (Transform arLocationRoot, Transform user, Location userLocation, Location objectLocation, bool heightIsRelative)

  Gets the game object world-position for location.
- static void PlaceGameObjectAtLocation (Transform arLocationRoot, Transform transform, Transform user, Location userLocation, Location objectLocation, bool heightIsRelative)

  Places the game object at location.
- static bool **Equal** (Location a, Location b, double eps=0.0000001)

Public Attributes

- double **Latitude**
- double **Longitude**
- double **Altitude**
- AltitudeMode **AltitudeMode** = AltitudeMode.GroundRelative
- string **Label** = ""
- bool **IgnoreAltitude** => AltitudeMode == AltitudeMode.Ignore
- DVector2 HorizontalVector => new DVector2(Latitude, Longitude)

  Gets the horizontal vector.

6.40.1 Detailed Description

Represents a geographical location.

Definition at line 18 of file Location.cs.

6.40.2 Member Function Documentation

6.40.2.1 Clone()

```
Location ARLocation.Location.Clone ( )
```

Clones this instance.

Returns

The clone.

Definition at line 59 of file Location.cs.

6.40.2.2 DistanceWithAltitude()

```
static double ARLocation.Location.DistanceWithAltitude (
            Location l1,
            Location l2 )  [static]
```

Calculates the full distance between locations, taking altitude into account.

Returns

The with altitude.

Parameters

l1	L1.
l2	L2.

Definition at line 187 of file Location.cs.

6.40.2.3 GetGameObjectPositionForLocation() [1/2]

```
static Vector3 ARLocation.Location.GetGameObjectPositionForLocation (
            Transform arLocationRoot,
            Vector3 userPosition,
            Location userLocation,
            Location objectLocation,
            bool heightIsRelative )  [static]
```

Gets the game object world-position for location.

Parameters

arLocationRoot	
userPosition	
userLocation	
objectLocation	
heightIsRelative	

Returns

Definition at line 234 of file Location.cs.

6.40.2.4 GetGameObjectPositionForLocation() [2/2]

```
static Vector3 ARLocation.Location.GetGameObjectPositionForLocation (
            Transform arLocationRoot,
            Transform user,
            Location userLocation,
            Location objectLocation,
            bool heightIsRelative )  [static]
```

Gets the game object world-position for location.

Returns

The game object position for location.

Parameters

arLocationRoot	
user	User.
userLocation	User location.
objectLocation	Object location.
heightIsRelative	If set to `true` height is relative.

Definition at line 254 of file Location.cs.

6.40.2.5 HaversineDistance()

```
static double ARLocation.Location.HaversineDistance (
            Location l1,
            Location l2 )  [static]
```

Horizontal distance, using the Haversine formula. https://stackoverflow.com/questions/41621957/a-more-eff

Returns

The distance, in meters.

Parameters

l1	L1.
l2	L2.

Definition at line 166 of file Location.cs.

6.40.2.6 HorizontalDistance()

```
static double ARLocation.Location.HorizontalDistance (
            Location l1,
            Location l2 )  [static]
```

Calculates the horizontal distance according to the current function set in the configuration.

Returns

The distance, in meters.

Parameters

l1	L1.
l2	L2.

Definition at line 93 of file Location.cs.

6.40.2.7 HorizontalVectorFromTo()

```
static DVector2 ARLocation.Location.HorizontalVectorFromTo (
            Location l1,
            Location l2 )  [static]
```

Calculates the horizontal vector pointing from l1 to l2, in meters.

Returns

The vector from to.

Parameters

l1	L1.
l2	L2.

Definition at line 201 of file Location.cs.

6.40.2.8 PlaceGameObjectAtLocation()

```
static void ARLocation.Location.PlaceGameObjectAtLocation (
            Transform arLocationRoot,
```

```
            Transform transform,
            Transform user,
            Location userLocation,
            Location objectLocation,
            bool heightIsRelative )  [static]
```

Places the game object at location.

Parameters

arLocationRoot	
transform	The GameObject's transform.
user	The user's point of view Transform, e.g., camera.
userLocation	User Location.
objectLocation	Object Location.
heightIsRelative	

Definition at line 269 of file Location.cs.

6.40.2.9 PlaneEllipsoidalFccDistance()

```
static double ARLocation.Location.PlaneEllipsoidalFccDistance (
            Location l1,
            Location l2 )  [static]
```

Horizontal distance using ellipsoidal projection on a plane. https://en.wikipedia.org/wiki/↩Geographical_distance

Returns

The distance, in meters.

Parameters

l1	
l2	

Returns

Definition at line 143 of file Location.cs.

6.40.2.10 PlaneSphericalDistance()

```
static double ARLocation.Location.PlaneSphericalDistance (
            Location l1,
            Location l2 )  [static]
```

Horizontal distance using spherical projection on a plane. https://en.wikipedia.org/wiki/Geographical_distance

Returns

The distance, in meters.

Parameters

l1	
l2	

Returns

Definition at line 118 of file Location.cs.

6.40.2.11 VectorFromTo()

```
static DVector3 ARLocation.Location.VectorFromTo (
            Location l1,
            Location l2,
            bool ignoreHeight = false )  [static]
```

Calculates the vector from l1 to l2, in meters, taking altitude into account.

Returns

The from to.

Parameters

l1	L1.
l2	L2.
ignoreHeight	If true, y = 0 in the output vector.

Definition at line 217 of file Location.cs.

6.40.3 Member Data Documentation

6.40.3.1 HorizontalVector

```
DVector2 ARLocation.Location.HorizontalVector => new DVector2(Latitude, Longitude)
```

Gets the horizontal vector.

The horizontal vector.

Definition at line 46 of file Location.cs.

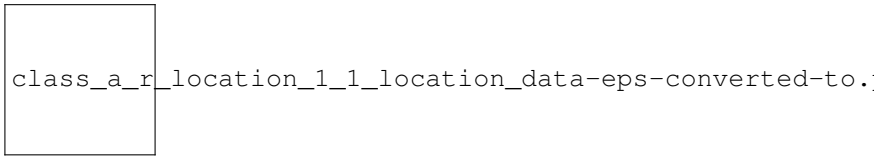
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Location/Location.cs

6.41 ARLocation.LocationData Class Reference

Data used to construct a spline passing trough a set of geographical locations.

Inheritance diagram for ARLocation.LocationData:



Public Member Functions

- override string **ToString** ()

Static Public Member Functions

- static LocationData **FromLocation** (Location location)

Public Attributes

- Location Location

  The geographical locations that the path will interpolate.

6.41.1 Detailed Description

Data used to construct a spline passing trough a set of geographical locations.

Definition at line 11 of file LocationData.cs.

6.41.2 Member Data Documentation

6.41.2.1 Location

```
Location ARLocation.LocationData.Location
```

The geographical locations that the path will interpolate.

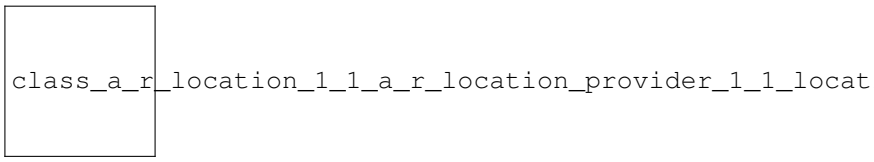
Definition at line 17 of file LocationData.cs.

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

- Assets/ARLocation/Scripts/ScriptableObjects/LocationData.cs

6.42 ARLocation.ARLocationProvider.LocationEnabledUnityEvent Class Reference

Inheritance diagram for ARLocation.ARLocationProvider.LocationEnabledUnityEvent:



6.42.1 Detailed Description

Definition at line 18 of file ARLocationProvider.cs.

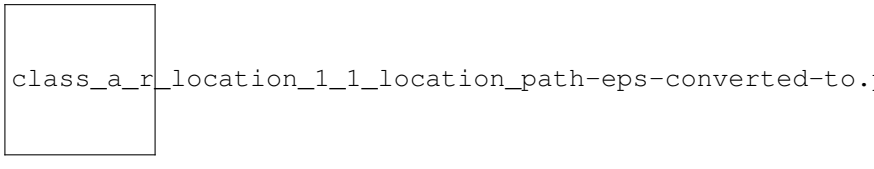
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Components/ARLocationProvider.cs

6.43 ARLocation.LocationPath Class Reference

Data used to construct a spline passing trough a set of geographical locations.

Inheritance diagram for ARLocation.LocationPath:



Public Attributes

- Location [ ] Locations

  The geographical locations that the path will interpolate.
- SplineType **SplineType** = SplineType.CatmullromSpline
- float Alpha = 0.5f

  The path's alpha/tension factor.
- float SceneViewScale = 1.0f

  The scale used in the editor scene viewer for drawing the path.

6.43.1 Detailed Description

Data used to construct a spline passing trough a set of geographical locations.

Definition at line 11 of file LocationPath.cs.

6.43.2 Member Data Documentation

6.43.2.1 Alpha

```
float ARLocation.LocationPath.Alpha = 0.5f
```

The path's alpha/tension factor.

Definition at line 26 of file LocationPath.cs.

6.43.2.2 Locations

```
Location [] ARLocation.LocationPath.Locations
```

The geographical locations that the path will interpolate.

Definition at line 17 of file LocationPath.cs.

6.43.2.3 SceneViewScale

```
float ARLocation.LocationPath.SceneViewScale = 1.0f
```

The scale used in the editor scene viewer for drawing the path.

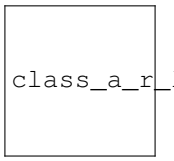
Definition at line 31 of file LocationPath.cs.

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

- Assets/ARLocation/Scripts/ScriptableObjects/LocationPath.cs

6.44 ARLocation.LocationPathInspector Class Reference

Inheritance diagram for ARLocation.LocationPathInspector:



Public Member Functions

- override void **OnInspectorGUI** ()

6.44.1 Detailed Description

Definition at line 10 of file LocationPathInspector.cs.

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

- Assets/ARLocation/Editor/LocationPathInspector.cs

6.45 ARLocation.LocationPropertyData Class Reference

Public Types

- enum **LocationPropertyType** { **Location**, **LocationData** }

Public Attributes

- LocationPropertyType **LocationInputType** = LocationPropertyType.Location
- LocationData **LocationData**
- Location **Location** = new Location()
- OverrideAltitudeData **OverrideAltitudeData** = new OverrideAltitudeData()

6.45.1 Detailed Description

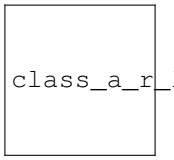
Definition at line 23 of file PlaceAtLocation.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.46 ARLocation.LocationPropertyDataDrawer Class Reference

Inheritance diagram for ARLocation.LocationPropertyDataDrawer:



Public Member Functions

- void **FindSerializedProperties** (SerializedProperty property)
- override float **GetPropertyHeight** (SerializedProperty property, GUIContent label)
- override void **OnGUI** (Rect position, SerializedProperty property, GUIContent label)

6.46.1 Detailed Description

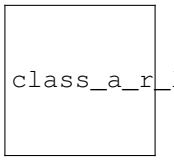
Definition at line 7 of file LocationPropertyDataDrawer.cs.

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

- Assets/ARLocation/Editor/LocationPropertyDataDrawer.cs

6.47 ARLocation.UI.LocationProviderInfo Class Reference

Inheritance diagram for ARLocation.UI.LocationProviderInfo:



6.47.1 Detailed Description

Definition at line 9 of file LocationProviderInfo.cs.

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

- Assets/ARLocation/Scripts/Components/UI/LocationProviderInfo.cs

6.48 ARLocation.LocationProviderOptions Class Reference

Public Attributes

- float TimeBetweenUpdates = 2.0f

  The minimum desired update time, in seconds.
- double MinDistanceBetweenUpdates

  The minimum distance between consecutive location updates, in meters.
- double AccuracyRadius = 25.0f

  The minimum accuracy of accepted location measurements, in meters.
- uint **MaxNumberOfUpdates**

6.48.1 Detailed Description

Definition at line 9 of file ILocationProvider.cs.

6.48.2 Member Data Documentation

6.48.2.1 AccuracyRadius

```
double ARLocation.LocationProviderOptions.AccuracyRadius = 25.0f
```

The minimum accuracy of accepted location measurements, in meters.

Definition at line 30 of file ILocationProvider.cs.

6.48.2.2 MinDistanceBetweenUpdates

```
double ARLocation.LocationProviderOptions.MinDistanceBetweenUpdates
```

The minimum distance between consecutive location updates, in meters.

Definition at line 21 of file ILocationProvider.cs.

6.48.2.3 TimeBetweenUpdates

```
float ARLocation.LocationProviderOptions.TimeBetweenUpdates = 2.0f
```

The minimum desired update time, in seconds.

Definition at line 15 of file ILocationProvider.cs.

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

- Assets/ARLocation/Scripts/Location/ILocationProvider.cs

6.49 ARLocation.LocationReading Struct Reference

Public Member Functions

- Location **ToLocation** ()
- override string **ToString** ()

Static Public Member Functions

- static double **HorizontalDistance** (LocationReading a, LocationReading b)

Public Attributes

- double **latitude**
- double **longitude**
- double **altitude**
- double **accuracy**
- int **floor**
- long timestamp

  Epoch time in ms

6.49.1 Detailed Description

Definition at line 4 of file LocationReading.cs.

6.49.2 Member Data Documentation

6.49.2.1 timestamp

```
long ARLocation.LocationReading.timestamp
```

Epoch time in ms

Definition at line 15 of file LocationReading.cs.

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

- Assets/ARLocation/Scripts/Location/LocationReading.cs

6.50 ARLocation.PlaceAtLocation.LocationSettingsData Class Reference

Public Member Functions

- Location **GetLocation** ()

Public Attributes

- LocationPropertyData **LocationInput** = new LocationPropertyData()

6.50.1 Detailed Description

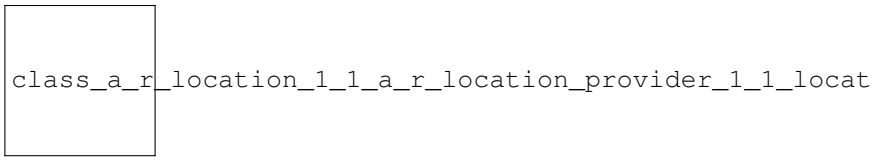
Definition at line 80 of file PlaceAtLocation.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.51 ARLocation.ARLocationProvider.LocationUpdatedUnityEvent Class Reference

Inheritance diagram for ARLocation.ARLocationProvider.LocationUpdatedUnityEvent:



6.51.1 Detailed Description

Definition at line 20 of file ARLocationProvider.cs.

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

- Assets/ARLocation/Scripts/Components/ARLocationProvider.cs

6.52 ARLocation.LowPassFilter Class Reference

Public Member Functions

- **LowPassFilter** (double smoothFactor=0.5f)
- double **Apply** (double value)

Public Attributes

- double **smoothFactor**

6.52.1 Detailed Description

Definition at line 3 of file LowPassFilter.cs.

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

- Assets/ARLocation/Scripts/Math/LowPassFilter.cs

6.53 ARLocation.Utils.Misc Class Reference

Static Public Member Functions

- static bool **IsARDevice** ()
- static float **FloatListAverage** (List< float > list)
- static float **GetNormalizedDegrees** (float value)
- static T **FindAndGetComponent**< **T** > (string name)
- static T **FindAndGetComponentAndLogError**< **T** > (string name, string message)
- static GameObject **FindAndLogError** (string name, string message)
- static Spline **BuildSpline** (SplineType type, Vector3[ ] points, int n, float alpha)
- static void **SetActiveOnAllChildren** (GameObject go, bool value)
- static void **SetGameObjectVisible** (GameObject go, bool value)
- static void **HideGameObject** (GameObject go)
- static void **ShowGameObject** (GameObject go)

6.53.1 Detailed Description

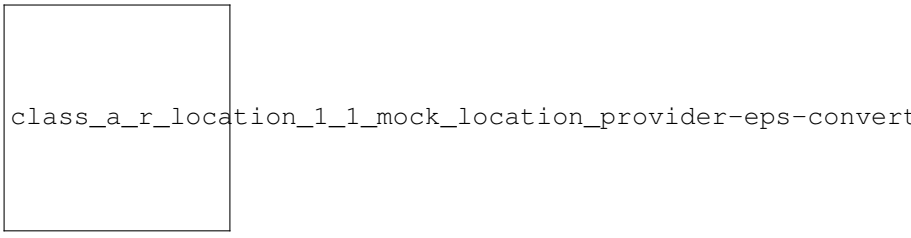
Definition at line 6 of file Misc.cs.

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

- Assets/ARLocation/Scripts/Utils/Misc.cs

6.54 ARLocation.MockLocationProvider Class Reference

Inheritance diagram for ARLocation.MockLocationProvider:



Public Attributes

- override string **Name** => "MockLocationProvider"
- override bool **IsCompassEnabled** => true
- Location **mockLocation** = new Location()

Protected Member Functions

- override HeadingReading ReadHeading ()

  Reads the heading from the device; should be implemented by each provider.
- override LocationReading ReadLocation ()

  Reads the location from the device; should be implemented by each provider.
- override void RequestLocationAndCompassUpdates ()

  Requests the location and compass updates from the device; should be implemented by each provider.
- override void UpdateLocationRequestStatus ()

  Updates the location service status from the device; should be implemented by each provider.

Additional Inherited Members

6.54.1 Detailed Description

Definition at line 6 of file MockLocationProvider.cs.

6.54.2 Member Function Documentation

6.54.2.1 ReadHeading()

```
override HeadingReading ARLocation.MockLocationProvider.ReadHeading ( )  [protected], [virtual]
```

Reads the heading from the device; should be implemented by each provider.

Returns

The heading.

Implements ARLocation.AbstractLocationProvider.

Definition at line 14 of file MockLocationProvider.cs.

6.54.2.2 ReadLocation()

```
override LocationReading ARLocation.MockLocationProvider.ReadLocation ( )  [protected], [virtual]
```

Reads the location from the device; should be implemented by each provider.

Returns

The location.

Implements ARLocation.AbstractLocationProvider.

Definition at line 31 of file MockLocationProvider.cs.

6.54.2.3 RequestLocationAndCompassUpdates()

```
override void ARLocation.MockLocationProvider.RequestLocationAndCompassUpdates ( )  [protected],
[virtual]
```

Requests the location and compass updates from the device; should be implemented by each provider.

Implements ARLocation.AbstractLocationProvider.

Definition at line 46 of file MockLocationProvider.cs.

6.54.2.4 UpdateLocationRequestStatus()

```
override void ARLocation.MockLocationProvider.UpdateLocationRequestStatus ( )  [protected],
[virtual]
```

Updates the location service status from the device; should be implemented by each provider.

Implements ARLocation.AbstractLocationProvider.

Definition at line 51 of file MockLocationProvider.cs.

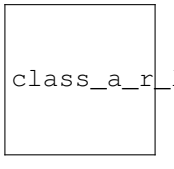
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Location/MockLocationProvider.cs

6.55 ARLocation.MoveAlongPath Class Reference

This component, when attached to a GameObject, makes it traverse a path that interpolates a given set of geographical locations.

Inheritance diagram for ARLocation.MoveAlongPath:



Classes

- class PathSettingsData
- class PlacementSettingsData
- class StateData

Public Member Functions

- void **SetLocationPath** (LocationPath path)
- void **Restart** ()
- void Play ()

  Starts playing or resumes the playback.
- void GoTo (float t)

  Moves the object to the spline point corresponding to the given parameter.
- void Pause ()

  Pauses the movement along the path.
- void Stop ()

  Stops the movement along the path.

Public Attributes

- PathSettingsData **PathSettings** = new PathSettingsData()
- PlacementSettingsData **PlacementSettings** = new PlacementSettingsData()
- bool **DebugMode**

6.55.1 Detailed Description

This component, when attached to a GameObject, makes it traverse a path that interpolates a given set of geographical locations.

Definition at line 16 of file MoveAlongPath.cs.

6.55.2 Member Function Documentation

6.55.2.1 GoTo()

```
void ARLocation.MoveAlongPath.GoTo (
            float t )
```

Moves the object to the spline point corresponding to the given parameter.

Parameters

t	Between 0 and 1

Definition at line 188 of file MoveAlongPath.cs.

6.55.2.2 Pause()

```
void ARLocation.MoveAlongPath.Pause ( )
```

Pauses the movement along the path.

Definition at line 196 of file MoveAlongPath.cs.

6.55.2.3 Play()

```
void ARLocation.MoveAlongPath.Play ( )
```

Starts playing or resumes the playback.

Definition at line 178 of file MoveAlongPath.cs.

6.55.2.4 Stop()

```
void ARLocation.MoveAlongPath.Stop ( )
```

Stops the movement along the path.

Definition at line 204 of file MoveAlongPath.cs.

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

- Assets/ARLocation/Scripts/Components/MoveAlongPath.cs

6.56 ARLocation.Utils.MovingAveragePosition Class Reference

Public Member Functions

- DVector3 **CalculateAveragePosition** ()
- void **AddEntry** (DVector3 position, double accuracy)
- void **Rest** ()

Public Attributes

- double **aMin** = 2.0
- double **aMax** = 10.0
- double **cutoff** = 0.01
- double **alpha** = 0.25

6.56.1 Detailed Description

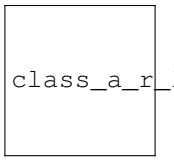
Definition at line 4 of file MovingAveragePosition.cs.

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

- Assets/ARLocation/Scripts/Utils/MovingAveragePosition.cs

6.57 ARLocation.PlaceAtLocation.ObjectUpdatedEvent Class Reference

Inheritance diagram for ARLocation.PlaceAtLocation.ObjectUpdatedEvent:



6.57.1 Detailed Description

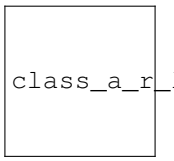
Definition at line 54 of file PlaceAtLocation.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.58 ARLocation.ARLocationOrientation.OnBeforeOrientationUpdatedEvent Class Reference

Inheritance diagram for ARLocation.ARLocationOrientation.OnBeforeOrientationUpdatedEvent:



6.58.1 Detailed Description

Definition at line 20 of file ARLocationOrientation.cs.

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

- Assets/ARLocation/Scripts/Components/ARLocationOrientation.cs

6.59 ARLocation.Hotspot.OnHotspotActivatedUnityEvent Class Reference

Inheritance diagram for ARLocation.Hotspot.OnHotspotActivatedUnityEvent:

6.59.1 Detailed Description

Definition at line 13 of file Hotspot.cs.

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

- Assets/ARLocation/Scripts/Components/Hotspot.cs

6.60 ARLocation.Utils.OpenStreetMapOptions Class Reference

Public Attributes

- TextAsset **OsmXmlFile**
- bool **FetchFromOverpassApi**
- OverpassRequestData **overPassRequestData**

6.60.1 Detailed Description

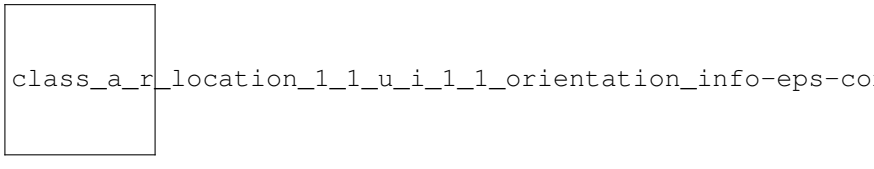
Definition at line 28 of file CreatePointOfInterestTextMeshes.cs.

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

- Assets/ARLocation/Scripts/Utils/CreatePointOfInterestTextMeshes.cs

6.61 ARLocation.UI.OrientationInfo Class Reference

Inheritance diagram for ARLocation.UI.OrientationInfo:



6.61.1 Detailed Description

Definition at line 6 of file OrientationInfo.cs.

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

- Assets/ARLocation/Scripts/Components/UI/OrientationInfo.cs

6.62 ARLocation.Utils.OverpassRequestData Class Reference

Public Attributes

- Location **SouthWest**
- Location **NorthEast**

6.62.1 Detailed Description

Definition at line 18 of file CreatePointOfInterestTextMeshes.cs.

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

- Assets/ARLocation/Scripts/Utils/CreatePointOfInterestTextMeshes.cs

6.63 ARLocation.OverrideAltitudeData Class Reference

Public Attributes

- bool **OverrideAltitude**
- double **Altitude**
- AltitudeMode **AltitudeMode** = AltitudeMode.GroundRelative

6.63.1 Detailed Description

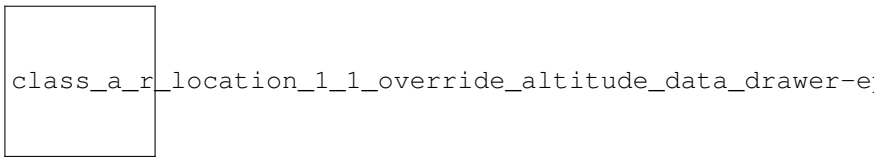
Definition at line 10 of file PlaceAtLocation.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.64 ARLocation.OverrideAltitudeDataDrawer Class Reference

Inheritance diagram for ARLocation.OverrideAltitudeDataDrawer:



Public Member Functions

- override void **OnGUI** (Rect position, SerializedProperty property, GUIContent label)
- override float **GetPropertyHeight** (SerializedProperty property, GUIContent label)

6.64.1 Detailed Description

Definition at line 7 of file OverrideAltitudeDataDrawer.cs.

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

- Assets/ARLocation/Editor/OverrideAltitudeDataDrawer.cs

6.65 ARLocation.MoveAlongPath.PathSettingsData Class Reference

Public Attributes

- LocationPath LocationPath

  The LocationPath describing the path to be traversed.
- float Speed = 1.0f

  The speed along the path.
- Vector3 Up = Vector3.up

  The up direction to be used for orientation along the path.
- bool Loop = true

  If true, play the path traversal in a loop.
- bool AutoPlay = true

  If true, start playing automatically.
- int SplineSampleCount = 250

  The number of points-per-segment used to calculate the spline.
- LineRenderer LineRenderer

  If present, renders the spline in the scene using the given line renderer.
- float **Offset**

6.65.1 Detailed Description

Definition at line 19 of file MoveAlongPath.cs.

6.65.2 Member Data Documentation

6.65.2.1 AutoPlay

```
bool ARLocation.MoveAlongPath.PathSettingsData.AutoPlay = true
```

If true, start playing automatically.

Definition at line 49 of file MoveAlongPath.cs.

6.65.2.2 LineRenderer

```
LineRenderer ARLocation.MoveAlongPath.PathSettingsData.LineRenderer
```

If present, renders the spline in the scene using the given line renderer.

Definition at line 61 of file MoveAlongPath.cs.

6.65.2.3 LocationPath

```
LocationPath ARLocation.MoveAlongPath.PathSettingsData.LocationPath
```

The LocationPath describing the path to be traversed.

Definition at line 25 of file MoveAlongPath.cs.

6.65.2.4 Loop

```
bool ARLocation.MoveAlongPath.PathSettingsData.Loop = true
```

If true, play the path traversal in a loop.

Definition at line 43 of file MoveAlongPath.cs.

6.65.2.5 Speed

```
float ARLocation.MoveAlongPath.PathSettingsData.Speed = 1.0f
```

The speed along the path.

Definition at line 31 of file MoveAlongPath.cs.

6.65.2.6 SplineSampleCount

```
int ARLocation.MoveAlongPath.PathSettingsData.SplineSampleCount = 250
```

The number of points-per-segment used to calculate the spline.

Definition at line 55 of file MoveAlongPath.cs.

6.65.2.7 Up

```
Vector3 ARLocation.MoveAlongPath.PathSettingsData.Up = Vector3.up
```

The up direction to be used for orientation along the path.

Definition at line 37 of file MoveAlongPath.cs.

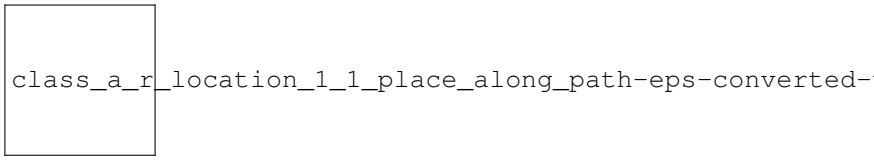
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Components/MoveAlongPath.cs

6.66 ARLocation.PlaceAlongPath Class Reference

This component places instances of a given prefab/GameObject along equally spaced positions in a LocationPath. Should be placed in the ARLocationRoot GameObject.

Inheritance diagram for ARLocation.PlaceAlongPath:



Public Attributes

- LocationPath Path

  The path to place the prefab instances on.
- GameObject Prefab

  The prefab/GameObject to be palced along the path.
- int ObjectCount = 10

  The number of object instances to be placed, excluding the endpoints. That is, the total number of instances is equal to objectCount + 2
- int SplineSampleSize = 200

  The size of the sample used to calculate the spline.
- PlaceAtLocation.PlaceAtOptions **PlacementSettings**
- bool **DebugMode**

6.66.1 Detailed Description

This component places instances of a given prefab/GameObject along equally spaced positions in a LocationPath. Should be placed in the ARLocationRoot GameObject.

Definition at line 14 of file PlaceAlongPath.cs.

6.66.2 Member Data Documentation

6.66.2.1 ObjectCount

```
int ARLocation.PlaceAlongPath.ObjectCount = 10
```

The number of object instances to be placed, excluding the endpoints. That is, the total number of instances is equal to objectCount + 2

Definition at line 35 of file PlaceAlongPath.cs.

6.66.2.2 Path

```
LocationPath ARLocation.PlaceAlongPath.Path
```

The path to place the prefab instances on.

Definition at line 22 of file PlaceAlongPath.cs.

6.66.2.3 Prefab

```
GameObject ARLocation.PlaceAlongPath.Prefab
```

The prefab/GameObject to be palced along the path.

Definition at line 28 of file PlaceAlongPath.cs.

6.66.2.4 SplineSampleSize

```
int ARLocation.PlaceAlongPath.SplineSampleSize = 200
```

The size of the sample used to calculate the spline.

Definition at line 41 of file PlaceAlongPath.cs.

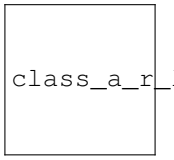
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Components/PlaceAlongPath.cs

6.67 ARLocation.PlaceAtLocation Class Reference

Apply to a GameObject to place it at a specified geographic location.

Inheritance diagram for ARLocation.PlaceAtLocation:



Classes

- class LocationSettingsData
- class ObjectUpdatedEvent
- class PlaceAtOptions
- class StateData

Public Member Functions

- void **Restart** ()
- void **UpdatePosition** (Location deviceLocation)

Static Public Member Functions

- static GameObject **CreatePlacedInstance** (GameObject go, Location location, PlaceAtOptions options, bool useDebugMode=false)
- static PlaceAtLocation **AddPlaceAtComponent** (GameObject go, Location location, PlaceAtOptions options, bool useDebugMode=false)

Public Attributes

- LocationSettingsData **LocationOptions** = new LocationSettingsData()
- PlaceAtOptions **PlacementOptions** = new PlaceAtOptions()
- bool **DebugMode**
- ObjectUpdatedEvent **ObjectLocationUpdated**
- ObjectUpdatedEvent **ObjectPositionUpdated**
- double **RawGpsDistance**
- bool **UseGroundHeight** => state.Location.AltitudeMode == AltitudeMode.GroundRelative

Properties

- Location **Location** `[get, set]`
- float **SceneDistance** `[get]`
- bool **Paused** `[get, set]`

6.67.1 Detailed Description

Apply to a GameObject to place it at a specified geographic location.

Definition at line 51 of file PlaceAtLocation.cs.

6.67.2 Member Data Documentation

6.67.2.1 RawGpsDistance

```
double ARLocation.PlaceAtLocation.RawGpsDistance
```

Initial value:

```
=>
            Location.HorizontalDistance(locationProvider.
      Provider.CurrentLocationRaw.ToLocation(),
                state.Location)
```

Definition at line 175 of file PlaceAtLocation.cs.

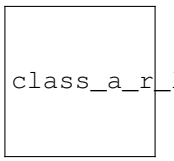
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.68 ARLocation.PlaceAtLocations Class Reference

This class instantiates a prefab at the given GPS locations. Must be in the `ARLocationRoot` GameObject with a `ARLocatedObjectsManager` Component.

Inheritance diagram for ARLocation.PlaceAtLocations:



Classes

- class Entry

Public Member Functions

- void **AddLocation** (Location location)

Public Attributes

- List< PlaceAtLocation.LocationSettingsData > **Locations**
- PlaceAtLocation.PlaceAtOptions **PlacementOptions**
- GameObject Prefab

  The game object that will be instantiated.
- bool **DebugMode**

6.68.1 Detailed Description

This class instantiates a prefab at the given GPS locations. Must be in the `ARLocationRoot` GameObject with a `ARLocatedObjectsManager` Component.

Definition at line 16 of file PlaceAtLocations.cs.

6.68.2 Member Data Documentation

6.68.2.1 Prefab

`GameObject ARLocation.PlaceAtLocations.Prefab`

The game object that will be instantiated.

Definition at line 34 of file PlaceAtLocations.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocations.cs

6.69 ARLocation.PlaceAtLocation.PlaceAtOptions Class Reference

Public Attributes

- float **MovementSmoothing** = 0.1f
- int **MaxNumberOfLocationUpdates**
- bool **UseMovingAverage**
- bool **HideObjectUntilItIsPlaced** = true

6.69.1 Detailed Description

Definition at line 59 of file PlaceAtLocation.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.70 ARLocation.MoveAlongPath.PlacementSettingsData Class Reference

Public Attributes

- AltitudeMode **AltitudeMode**
- uint **MaxNumberOfLocationUpdates**

6.70.1 Detailed Description

Definition at line 68 of file MoveAlongPath.cs.

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

- Assets/ARLocation/Scripts/Components/MoveAlongPath.cs

6.71 ARLocation.Utils.POIData Class Reference

Public Attributes

- Location **location**
- string **name**

6.71.1 Detailed Description

Definition at line 11 of file CreatePointOfInterestTextMeshes.cs.

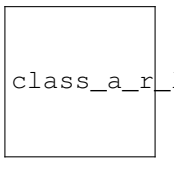
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Utils/CreatePointOfInterestTextMeshes.cs

6.72 ARLocation.RenderPathLine Class Reference

This component renders a LocationPath using a given LineRenderer.

Inheritance diagram for ARLocation.RenderPathLine:



Public Attributes

- LocationPath LocationPath

  The LocationPath describing the path to be traversed.
- AltitudeMode **AltitudeMode**
- int SplineSampleCount = 250

  The number of points-per-segment used to calculate the spline.
- int LineRenderSampleCount = 250

  The number of points-per-segment used to calculate the spline.
- LineRenderer LineRenderer

  If present, renders the spline in the scene using the given line renderer.

6.72.1 Detailed Description

This component renders a LocationPath using a given LineRenderer.

Definition at line 12 of file RenderPathLine.cs.

6.72.2 Member Data Documentation

6.72.2.1 LineRenderer

```
LineRenderer ARLocation.RenderPathLine.LineRenderer
```

If present, renders the spline in the scene using the given line renderer.

Definition at line 41 of file RenderPathLine.cs.

6.72.2.2 LineRenderSampleCount

```
int ARLocation.RenderPathLine.LineRenderSampleCount = 250
```

The number of points-per-segment used to calculate the spline.

Definition at line 35 of file RenderPathLine.cs.

6.72.2.3 LocationPath

```
LocationPath ARLocation.RenderPathLine.LocationPath
```

The LocationPath describing the path to be traversed.

Definition at line 18 of file RenderPathLine.cs.

6.72.2.4 SplineSampleCount

```
int ARLocation.RenderPathLine.SplineSampleCount = 250
```

The number of points-per-segment used to calculate the spline.

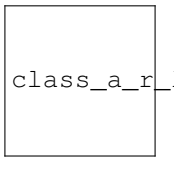
Definition at line 29 of file RenderPathLine.cs.

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

- Assets/ARLocation/Scripts/Components/RenderPathLine.cs

6.73 ARLocation.Utils.RotateObject Class Reference

Inheritance diagram for ARLocation.Utils.RotateObject:



Public Attributes

- float **Speed** = 10.0f
- Vector3 **Axis** = Vector3.up

6.73.1 Detailed Description

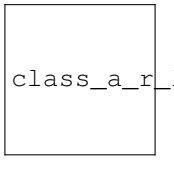
Definition at line 5 of file RotateObject.cs.

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

- Assets/ARLocation/Scripts/Utils/RotateObject.cs

6.74 ARLocation.Utils.SelectScene Class Reference

Inheritance diagram for ARLocation.Utils.SelectScene:



Public Member Functions

- void **LoadScene** (string sceneName)

6.74.1 Detailed Description

Definition at line 6 of file SelectScene.cs.

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

- Assets/ARLocation/Scripts/Utils/SelectScene.cs

6.75 ARLocation.GroundHeight.SettingsData Class Reference

Public Attributes

- float **InitialGroundHeightGuess** = 1.4f
- float **MinGroundHeight** = 0.4f
- float **MaxGroundHeight** = 3.0f
- bool **UseArLocationConfigSettings** = true

6.75.1 Detailed Description

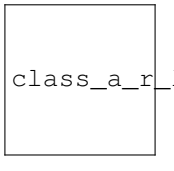
Definition at line 22 of file GroundHeight.cs.

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

- Assets/ARLocation/Scripts/Components/GroundHeight.cs

6.76 ARLocation.Utils.ShowHideSelfOnPointerClick Class Reference

Inheritance diagram for ARLocation.Utils.ShowHideSelfOnPointerClick:



6.76.1 Detailed Description

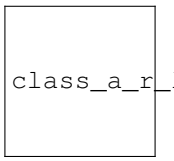
Definition at line 6 of file ShowHideSelfOnPointerClick.cs.

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

- Assets/ARLocation/Scripts/Utils/ShowHideSelfOnPointerClick.cs

6.77 ARLocation.Utils.Singleton< T > Class Template Reference

Inheritance diagram for ARLocation.Utils.Singleton< T >:



Public Member Functions

- virtual void **Awake** ()

Properties

- static T Instance [get]

  Access singleton instance through this propriety.

6.77.1 Detailed Description

Type Constraints

T* : *MonoBehaviour

Definition at line 7 of file Singleton.cs.

6.77.2 Property Documentation

6.77.2.1 Instance

```
T ARLocation.Utils.Singleton< T >.Instance [static], [get]
```

Access singleton instance through this propriety.

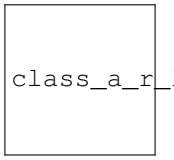
Definition at line 18 of file Singleton.cs.

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

- Assets/ARLocation/Scripts/Utils/Singleton.cs

6.78 ARLocation.Utils.SmoothMove Class Reference

Inheritance diagram for ARLocation.Utils.SmoothMove:



Public Member Functions

- void **Move** (Vector3 to, Action callback=null)

Static Public Member Functions

- static SmoothMove **AddSmoothMove** (GameObject go, float epsilon)

Public Attributes

- float **Epsilon** = 0.5f

Properties

- Vector3 **Target** `[get, set]`

6.78.1 Detailed Description

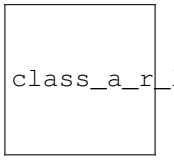
Definition at line 7 of file SmoothMove.cs.

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

- Assets/ARLocation/Scripts/Utils/SmoothMove.cs

6.79 ARLocation.Spline Class Reference

Inheritance diagram for ARLocation.Spline:



Public Member Functions

- abstract void CalculateSegments (int n)

  Calculate the catmull-rom segments. Also estimates the curve's length.
- Vector3 GetPointAtArcLength (float s)

  Returns the point of the spline at a given arc-length.
- CurvePointData GetPointAndTangentAtArcLength (float s)

  Returns a CurvePointData whith the point and tangent of the spline at a given arc-length.
- void DrawCurveWithLineRenderer (LineRenderer renderer, System.Func< Vector3, Vector3 > func, int n=100)

  Draws the curve using a given LineRenderer, with points being processed by a given function beforehand.
- Vector3 [ ] SamplePoints (int n, System.Func< Vector3, Vector3 > func)

  Calculates a sample of (N+2) equidistant points along the spline.
- Vector3 [ ] SamplePoints (int n)

  Calculates a sample of (N+2) equidistant points along the spline.
- void DrawGizmos ()

  Draw the curve and sample point using Gizmos.

Protected Attributes

- Curve [ ] segments

  The CatmullRom curve-segments of the spline.
- int segmentCount = 0

  The number of segments that make up the spline.
- float [ ] **lengths**

Properties

- Vector3 [ ] Points `[get, protected set]`

  The points interpolated of the spline.
- float Length `[get, protected set]`

  The full (estimated) length of the spline.

6.79.1 Detailed Description

Definition at line 12 of file Spline.cs.

6.79.2 Member Function Documentation

6.79.2.1 CalculateSegments()

```
abstract void ARLocation.Spline.CalculateSegments (
            int n )  [pure virtual]
```

Calculate the catmull-rom segments. Also estimates the curve's length.

Parameters

n	The number sample points used to estimate each segment's length.

Implemented in ARLocation.CatmullRomSpline, and ARLocation.LinearSpline.

6.79.2.2 DrawCurveWithLineRenderer()

```
void ARLocation.Spline.DrawCurveWithLineRenderer (
            LineRenderer renderer,
            System.Func< Vector3, Vector3 > func,
            int n = 100 )
```

Draws the curve using a given LineRenderer, with points being processed by a given function beforehand.

Parameters

renderer	
func	
n	

Definition at line 91 of file Spline.cs.

6.79.2.3 DrawGizmos()

```
void ARLocation.Spline.DrawGizmos ( )
```

Draw the curve and sample point using Gizmos.

Definition at line 142 of file Spline.cs.

6.79.2.4 GetPointAndTangentAtArcLength()

```
CurvePointData ARLocation.Spline.GetPointAndTangentAtArcLength (
            float s )
```

Returns a CurvePointData whith the point and tangent of the spline at a given arc-length.

Parameters

s	The arc-length.

Returns

Definition at line 69 of file Spline.cs.

6.79.2.5 GetPointAtArcLength()

```
Vector3 ARLocation.Spline.GetPointAtArcLength (
            float s )
```

Returns the point of the spline at a given arc-length.

Parameters

s	The arc-length.

Returns

Definition at line 48 of file Spline.cs.

6.79.2.6 SamplePoints() [1/2]

```
Vector3 [] ARLocation.Spline.SamplePoints (
            int n,
            System.Func< Vector3, Vector3 > func )
```

Calculates a sample of (N+2) equidistant points along the spline.

Parameters

n	The number of points in the sample will be (N+2).
func	A function that can be used to transform the sampled poins.

Returns

Definition at line 114 of file Spline.cs.

6.79.2.7 SamplePoints() [2/2]

```
Vector3 [] ARLocation.Spline.SamplePoints (
            int n )
```

Calculates a sample of (N+2) equidistant points along the spline.

Parameters

n	The number of points in the sample will be (N+2).

Returns

Definition at line 134 of file Spline.cs.

6.79.3 Member Data Documentation

6.79.3.1 segmentCount

```
int ARLocation.Spline.segmentCount = 0  [protected]
```

The number of segments that make up the spline.

Definition at line 28 of file Spline.cs.

6.79.3.2 segments

```
Curve [] ARLocation.Spline.segments  [protected]
```

The CatmullRom curve-segments of the spline.

Definition at line 23 of file Spline.cs.

6.79.4 Property Documentation

6.79.4.1 Length

```
float ARLocation.Spline.Length  [get], [protected set]
```

The full (estimated) length of the spline.

Definition at line 33 of file Spline.cs.

6.79.4.2 Points

```
Vector3 [] ARLocation.Spline.Points  [get], [protected set]
```

The points interpolated of the spline.

Definition at line 18 of file Spline.cs.

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

- Assets/ARLocation/Scripts/Math/Spline.cs

6.80 ARLocation.MoveAlongPath.StateData Class Reference

Public Attributes

- uint **UpdateCount**
- Vector3 [ ] **Points**
- int **PointCount**
- bool **Playing**
- Spline **Spline**
- Vector3 **Translation**

6.80.1 Detailed Description

Definition at line 79 of file MoveAlongPath.cs.

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

- Assets/ARLocation/Scripts/Components/MoveAlongPath.cs

6.81 ARLocation.PlaceAtLocation.StateData Class Reference

Public Attributes

- Location **Location**
- uint **LocationUpdatedCount**
- uint **PositionUpdatedCount**
- bool **Paused**

6.81.1 Detailed Description

Definition at line 120 of file PlaceAtLocation.cs.

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

- Assets/ARLocation/Scripts/Components/PlaceAtLocation.cs

6.82 ARLocation.Hotspot.StateData Class Reference

Public Attributes

- bool **Activated**
- GameObject **Instance**
- Location **Location**

6.82.1 Detailed Description

Definition at line 41 of file Hotspot.cs.

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

- Assets/ARLocation/Scripts/Components/Hotspot.cs

6.83 ARLocation.GroundHeight.StateData Class Reference

Public Attributes

- float **CurrentGroundY**
- float **CurrentPlaneDistance** = -1.0f
- Vector3 **CurrentPlaneCenter**
- bool **NeedsUpdate** = true

6.83.1 Detailed Description

Definition at line 38 of file GroundHeight.cs.

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

- Assets/ARLocation/Scripts/Components/GroundHeight.cs

6.84 ARLocation.UnityLocationProvider Class Reference

Inheritance diagram for ARLocation.UnityLocationProvider:

Public Attributes

- override string **Name** => "UnityLocationProvider"
- override bool **IsCompassEnabled** => Input.compass.enabled

Protected Member Functions

- override void RequestLocationAndCompassUpdates ()

  Requests the location and compass updates from the device; should be implemented by each provider.
- override void **InnerOnEnabled** ()
- override void UpdateLocationRequestStatus ()

  Updates the location service status from the device; should be implemented by each provider.
- override LocationReading ReadLocation ()

  Reads the location from the device; should be implemented by each provider.
- override HeadingReading ReadHeading ()

  Reads the heading from the device; should be implemented by each provider.

Additional Inherited Members

6.84.1 Detailed Description

Definition at line 7 of file UnityLocationProvider.cs.

6.84.2 Member Function Documentation

6.84.2.1 ReadHeading()

```
override HeadingReading ARLocation.UnityLocationProvider.ReadHeading ( )  [protected], [virtual]
```

Reads the heading from the device; should be implemented by each provider.

Returns

The heading.

Implements ARLocation.AbstractLocationProvider.

Definition at line 73 of file UnityLocationProvider.cs.

6.84.2.2 ReadLocation()

```
override LocationReading ARLocation.UnityLocationProvider.ReadLocation ( )  [protected], [virtual]
```

Reads the location from the device; should be implemented by each provider.

Returns

The location.

Implements ARLocation.AbstractLocationProvider.

Definition at line 53 of file UnityLocationProvider.cs.

6.84.2.3 RequestLocationAndCompassUpdates()

```
override void ARLocation.UnityLocationProvider.RequestLocationAndCompassUpdates ( )  [protected],
[virtual]
```

Requests the location and compass updates from the device; should be implemented by each provider.

Implements ARLocation.AbstractLocationProvider.

Definition at line 14 of file UnityLocationProvider.cs.

6.84.2.4 UpdateLocationRequestStatus()

```
override void ARLocation.UnityLocationProvider.UpdateLocationRequestStatus ( )  [protected],
[virtual]
```

Updates the location service status from the device; should be implemented by each provider.

Implements ARLocation.AbstractLocationProvider.

Definition at line 31 of file UnityLocationProvider.cs.

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

- Assets/ARLocation/Scripts/Location/UnityLocationProvider.cs