AR + GPS Location

1.0.0

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

1.0.0

Initial Release

Chapter 3

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 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

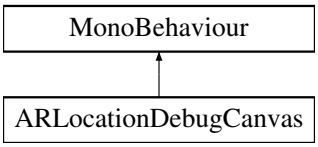
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 6

Class Documentation

6.1 ARLocationDebugCanvas Class Reference

Inheritance diagram for ARLocationDebugCanvas:



Public Member Functions

- void **SetDebugText** (string val)

6.1.1 Detailed Description

Definition at line 7 of file ARLocationDebugCanvas.cs.

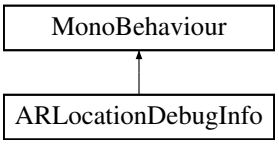
The documentation for this class was generated from the following file:

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

6.2 ARLocationDebugInfo Class Reference

Positions UI Panels with debug information on top of objects managed by a ARLocatedObjectsManager.

Inheritance diagram for ARLocationDebugInfo:



Static Public Member Functions

- static void **UpdateDebugInfoPanelScreenPositionAndText** (GameObject target, GameObject panel, string text)
- static Vector3 GetDebugInfoPanelScreenPosition (GameObject gameObject)

  Returns the screen position just above a GameObject which has a MeshRenderer component (in itself or in one of its children).

Public Attributes

- Canvas canvas

  A screen space overlay UI Canvas that will hold the panels.
- GameObject debugInfoPrefab

  A UI prefab containing a text object.

6.2.1 Detailed Description

Positions UI Panels with debug information on top of objects managed by a ARLocatedObjectsManager.

Definition at line 11 of file ARLocationDebugInfo.cs.

6.2.2 Member Function Documentation

6.2.2.1 GetDebugInfoPanelScreenPosition()

```
static Vector3 ARLocationDebugInfo.GetDebugInfoPanelScreenPosition (
            GameObject gameObject )  [static]
```

Returns the screen position just above a GameObject which has a MeshRenderer component (in itself or in one of its children).

Parameters

gameObject	

Returns

Definition at line 127 of file ARLocationDebugInfo.cs.

6.2.3 Member Data Documentation

6.2.3.1 canvas

```
Canvas ARLocationDebugInfo.canvas
```

A screen space overlay UI Canvas that will hold the panels.

Definition at line 17 of file ARLocationDebugInfo.cs.

6.2.3.2 debugInfoPrefab

```
GameObject ARLocationDebugInfo.debugInfoPrefab
```

A UI prefab containing a text object.

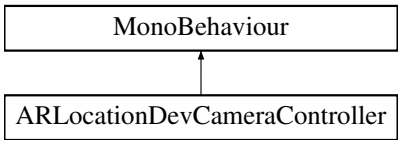
Definition at line 23 of file ARLocationDebugInfo.cs.

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

- Assets/ARLocation/Scripts/ARLocationDebugInfo.cs

6.3 ARLocationDevCameraController Class Reference

Inheritance diagram for ARLocationDevCameraController:



Public Attributes

- float MouseSensitivity = 1.0f
  The mouse look/rotation sensitivity.
- float speed = 1.0f
  The walking speed

6.3.1 Detailed Description

Definition at line 6 of file ARLocationDevCameraController.cs.

6.3.2 Member Data Documentation

6.3.2.1 MouseSensitivity

```
float ARLocationDevCameraController.MouseSensitivity = 1.0f
```

The mouse look/rotation sensitivity.

Definition at line 11 of file ARLocationDevCameraController.cs.

6.3.2.2 speed

```
float ARLocationDevCameraController.speed = 1.0f
```

The walking speed

Definition at line 16 of file ARLocationDevCameraController.cs.

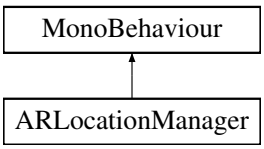
The documentation for this class was generated from the following file:

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

6.4 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 ARLocationManager:



Public Member Functions

- delegate void OnObjectAddedDelegate (ARLocationManagerEntry entry)
  A delegate that is called when a new object is addded to the manager.
- delegate void OnRestartDelegate ()
  A delegate that is called when the manager is restarted.
- delegate void OnStartDelegate ()
  Called when the manager has started and objects can be added.
- ARLocationManagerEntry GetEntry (int id)
  Fetches the entry for a given instance id.
- void OnObjectAdded (OnObjectAddedDelegate del)
  Ons the object added.
- void OnRestart (OnRestartDelegate del)
  Adds a delegate to be called when the ARLocationManager session is restarted.
- void OnStart (OnStartDelegate del)
  Adds a listener for the OnStart event.
- int Add (ARLocationManagerEntry entry)
  Registers a new entry in the ARLocationManager.
- void UpdateObjectPosition (int id)
  Updates the object position.
- void UpdateObjectPosition (ARLocationManagerEntry entry, Location deviceLocation, Vector3 delta, bool forceDisableSmooth=false)
  Updates the object position.

Public Attributes

- ARLocationManagerEntry [ ] objects

  An array describing a set of objects to be placed on the scene via GPS/geolocation coordinates.
- float stabilizationWaitTime = 6.0f

  The stabilization wait time. This is the time, during startup, where all is smoothing is disabled so that the objects can position themselves quickly in their correct places.
- float arSessionResetDistance = 20.0f

  The ar session reset distance.
- GameObject resetWaitScreen

  A 2D screen-space canvas to fill the screen while the AR Session is being reset (when arSessionResetDistance > 0)

6.4.1 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 91 of file ARLocationManager.cs.

6.4.2 Member Function Documentation

6.4.2.1 Add()

```
int ARLocationManager.Add (
            ARLocationManagerEntry entry )
```

Registers a new entry in the ARLocationManager.

Parameters

entry	Entry.

Definition at line 232 of file ARLocationManager.cs.

6.4.2.2 GetEntry()

```
ARLocationManagerEntry ARLocationManager.GetEntry (
            int id )
```

Fetches the entry for a given instance id.

Parameters

id	The transform instance ID

Returns

ARLocationManagerEntry

Definition at line 192 of file ARLocationManager.cs.

6.4.2.3 OnObjectAdded()

```
void ARLocationManager.OnObjectAdded (
            OnObjectAddedDelegate del )
```

Ons the object added.

Parameters

del	Del.

Definition at line 201 of file ARLocationManager.cs.

6.4.2.4 OnObjectAddedDelegate()

```
delegate void ARLocationManager.OnObjectAddedDelegate (
            ARLocationManagerEntry entry )
```

A delegate that is called when a new object is addded to the manager.

6.4.2.5 OnRestart()

```
void ARLocationManager.OnRestart (
            OnRestartDelegate del )
```

Adds a delegate to be called when the ARLocationManager session is restarted.

Parameters

del	

Definition at line 210 of file ARLocationManager.cs.

6.4.2.6 OnRestartDelegate()

```
delegate void ARLocationManager.OnRestartDelegate ( )
```

A delegate that is called when the manager is restarted.

6.4.2.7 OnStart()

```
void ARLocationManager.OnStart (
            OnStartDelegate del )
```

Adds a listener for the OnStart event.

Parameters

del	Del.

Definition at line 219 of file ARLocationManager.cs.

6.4.2.8 OnStartDelegate()

```
delegate void ARLocationManager.OnStartDelegate ( )
```

Called when the manager has started and objects can be added.

6.4.2.9 UpdateObjectPosition() [1/2]

```
void ARLocationManager.UpdateObjectPosition (
            int id )
```

Updates the object position.

Parameters

id	The object's transform instance ID.

Definition at line 292 of file ARLocationManager.cs.

6.4.2.10 UpdateObjectPosition() [2/2]

```
void ARLocationManager.UpdateObjectPosition (
            ARLocationManagerEntry entry,
```

```
            Location deviceLocation,
            Vector3 delta,
            bool forceDisableSmooth = false )
```

Updates the object position.

Parameters

instance	Instance.
instanceLocation	Instance location.
instanceOptions	Instance options.
deviceLocation	Device location.

Definition at line 314 of file ARLocationManager.cs.

6.4.3 Member Data Documentation

6.4.3.1 arSessionResetDistance

```
float ARLocationManager.arSessionResetDistance = 20.0f
```

The ar session reset distance.

Definition at line 116 of file ARLocationManager.cs.

6.4.3.2 objects

```
ARLocationManagerEntry [] ARLocationManager.objects
```

An array describing a set of objects to be placed on the scene via GPS/geolocation coordinates.

Definition at line 98 of file ARLocationManager.cs.

6.4.3.3 resetWaitScreen

```
GameObject ARLocationManager.resetWaitScreen
```

A 2D screen-space canvas to fill the screen while the AR Session is being reset (when arSessionResetDistance $>$ 0)

Definition at line 122 of file ARLocationManager.cs.

6.4.3.4 stabilizationWaitTime

```
float ARLocationManager.stabilizationWaitTime = 6.0f
```

The stabilization wait time. This is the time, during startup, where all is smoothing is disabled so that the objects can position themselves quickly in their correct places.

Definition at line 107 of file ARLocationManager.cs.

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

- Assets/ARLocation/Scripts/ARLocationManager.cs

6.5 ARLocationManagerEntry Class Reference

This structure holds all data for a positioned GameObject in the ARLocationManager.

Public Member Functions

- void Relocate (Location newLocation)
  Changes the location of the entry.

Public Attributes

- GameObject instance
  The GameObject to be placed in the scene.
- Location location
  The GPS/geolocation coordinates.
- ARLocationObjectOptions options
  The placement options.
- bool isDirty = true
  Dirty location flag.

6.5.1 Detailed Description

This structure holds all data for a positioned GameObject in the ARLocationManager.

Definition at line 48 of file ARLocationManager.cs.

6.5.2 Member Function Documentation

6.5.2.1 Relocate()

```
void ARLocationManagerEntry.Relocate (
            Location newLocation )
```

Changes the location of the entry.

Parameters

newLocation	

Definition at line 78 of file ARLocationManager.cs.

6.5.3 Member Data Documentation

6.5.3.1 instance

```
GameObject ARLocationManagerEntry.instance
```

The GameObject to be placed in the scene.

Definition at line 54 of file ARLocationManager.cs.

6.5.3.2 isDirty

```
bool ARLocationManagerEntry.isDirty = true
```

Dirty location flag.

Definition at line 72 of file ARLocationManager.cs.

6.5.3.3 location

```
Location ARLocationManagerEntry.location
```

The GPS/geolocation coordinates.

Definition at line 60 of file ARLocationManager.cs.

6.5.3.4 options

```
ARLocationObjectOptions ARLocationManagerEntry.options
```

The placement options.

Definition at line 66 of file ARLocationManager.cs.

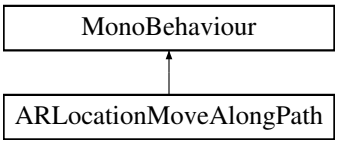
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARLocationManager.cs

6.6 ARLocationMoveAlongPath 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 ARLocationMoveAlongPath:



Public Member Functions

- 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

- 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.
- bool heightRelativeToDevice = true

  If true, all altitude data is considered relative to the current device elevation.
- int splineSampleCount = 250

  The number of points-per-segment used to calculate the spline.
- float movementSmoothingFactor = 50.0f

  The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.
- bool showDebugInfo = true

  Renders debug information. Requires an ARLocationDebugInfo at the ARLocationRoot object.
- LineRenderer lineRenderer

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

6.6.1 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 10 of file ARLocationMoveAlongPath.cs.

6.6.2 Member Function Documentation

6.6.2.1 GoTo()

```
void ARLocationMoveAlongPath.GoTo (
            float t )
```

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

Parameters

t	Between 0 and 1

Definition at line 176 of file ARLocationMoveAlongPath.cs.

6.6.2.2 Pause()

```
void ARLocationMoveAlongPath.Pause ( )
```

Pauses the movement along the path.

Definition at line 184 of file ARLocationMoveAlongPath.cs.

6.6.2.3 Play()

```
void ARLocationMoveAlongPath.Play ( )
```

Starts playing or resumes the playback.

Definition at line 166 of file ARLocationMoveAlongPath.cs.

6.6.2.4 Stop()

```
void ARLocationMoveAlongPath.Stop ( )
```

Stops the movement along the path.

Definition at line 192 of file ARLocationMoveAlongPath.cs.

6.6.3 Member Data Documentation

6.6.3.1 autoPlay

```
bool ARLocationMoveAlongPath.autoPlay = true
```

If true, start playing automatically.

Definition at line 40 of file ARLocationMoveAlongPath.cs.

6.6.3.2 heightRelativeToDevice

```
bool ARLocationMoveAlongPath.heightRelativeToDevice = true
```

If true, all altitude data is considered relative to the current device elevation.

Definition at line 46 of file ARLocationMoveAlongPath.cs.

6.6.3.3 lineRenderer

```
LineRenderer ARLocationMoveAlongPath.lineRenderer
```

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

Definition at line 70 of file ARLocationMoveAlongPath.cs.

6.6.3.4 locationPath

```
LocationPath ARLocationMoveAlongPath.locationPath
```

The LocationPath describing the path to be traversed.

Definition at line 16 of file ARLocationMoveAlongPath.cs.

6.6.3.5 loop

```
bool ARLocationMoveAlongPath.loop = true
```

If true, play the path traversal in a loop.

Definition at line 34 of file ARLocationMoveAlongPath.cs.

6.6.3.6 movementSmoothingFactor

```
float ARLocationMoveAlongPath.movementSmoothingFactor = 50.0f
```

The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

Definition at line 58 of file ARLocationMoveAlongPath.cs.

6.6.3.7 showDebugInfo

```
bool ARLocationMoveAlongPath.showDebugInfo = true
```

Renders debug information. Requires an ARLocationDebugInfo at the ARLocationRoot object.

Definition at line 64 of file ARLocationMoveAlongPath.cs.

6.6.3.8 speed

```
float ARLocationMoveAlongPath.speed = 1.0f
```

The speed along the path.

Definition at line 22 of file ARLocationMoveAlongPath.cs.

6.6.3.9 splineSampleCount

```
int ARLocationMoveAlongPath.splineSampleCount = 250
```

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

Definition at line 52 of file ARLocationMoveAlongPath.cs.

6.6.3.10 up

```
Vector3 ARLocationMoveAlongPath.up = Vector3.up
```

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

Definition at line 28 of file ARLocationMoveAlongPath.cs.

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

- Assets/ARLocation/Scripts/ARLocationMoveAlongPath.cs

6.7 ARLocationObjectOptions Class Reference

The options passed to the ARLocationManager when adding a new positioned GameObject.

Public Attributes

- bool isHeightRelative

  If true, the altitude will be computed as relative to the device level.
- bool showDebugInfoPanel

  If true, will display a UI panel with debug information above the object.
- bool createInstance

  If true, will clone the object when placing it on the scene.
- float movementSmoothingFactor

  The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

6.7.1 Detailed Description

The options passed to the ARLocationManager when adding a new positioned GameObject.

Definition at line 17 of file ARLocationManager.cs.

6.7.2 Member Data Documentation

6.7.2.1 createInstance

```
bool ARLocationObjectOptions.createInstance
```

If true, will clone the object when placing it on the scene.

Definition at line 35 of file ARLocationManager.cs.

6.7.2.2 isHeightRelative

```
bool ARLocationObjectOptions.isHeightRelative
```

If true, the altitude will be computed as relative to the device level.

Definition at line 23 of file ARLocationManager.cs.

6.7.2.3 movementSmoothingFactor

```
float ARLocationObjectOptions.movementSmoothingFactor
```

The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

Definition at line 41 of file ARLocationManager.cs.

6.7.2.4 showDebugInfoPanel

```
bool ARLocationObjectOptions.showDebugInfoPanel
```

If true, will display a UI panel with debug information above the object.

Definition at line 29 of file ARLocationManager.cs.

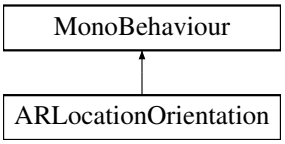
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARLocationManager.cs

6.8 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 ARLocationOrientation:



Public Member Functions

- void Restart ()

  Restarts the orientation tracking.

Public Attributes

- int averageCount = 250

  Only update after measuring the heading N times, and take the average.
- bool measureOnlyAtStartup = false

  If true, only measure during the session's start-up. If false, update the heading every time we calculate an average value. Setting to false will give more accurate results but may cause some jittering in the scene object's.
- bool useRawUntilFirstAverage = true

  If set to true, use raw heading values until measuring the first average.
- bool noSmoothingBeforeFirstAverage = true

  If set to true, disable movement smoothing until we get a first averaged value.
- float smoothing = 50.0f

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

6.8.1 Detailed Description

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.

Definition at line 9 of file ARLocationOrientation.cs.

6.8.2 Member Function Documentation

6.8.2.1 Restart()

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

Restarts the orientation tracking.

Definition at line 54 of file ARLocationOrientation.cs.

6.8.3 Member Data Documentation

6.8.3.1 averageCount

```
int ARLocationOrientation.averageCount = 250
```

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

Definition at line 15 of file ARLocationOrientation.cs.

6.8.3.2 measureOnlyAtStartup

```
bool ARLocationOrientation.measureOnlyAtStartup = false
```

If true, only measure during the session's start-up. If false, update the heading every time we calculate an average value. Setting to false will give more accurate results but may cause some jittering in the scene object's.

Definition at line 23 of file ARLocationOrientation.cs.

6.8.3.3 noSmoothingBeforeFirstAverage

```
bool ARLocationOrientation.noSmoothingBeforeFirstAverage = true
```

If set to true, disable movement smoothing until we get a first averaged value.

Definition at line 35 of file ARLocationOrientation.cs.

6.8.3.4 smoothing

```
float ARLocationOrientation.smoothing = 50.0f
```

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

Definition at line 41 of file ARLocationOrientation.cs.

6.8.3.5 useRawUntilFirstAverage

```
bool ARLocationOrientation.useRawUntilFirstAverage = true
```

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

Definition at line 29 of file ARLocationOrientation.cs.

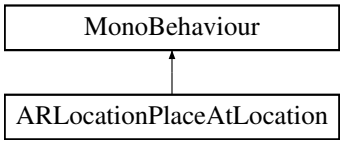
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARLocationOrientation.cs

6.9 ARLocationPlaceAtLocation Class Reference

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

Inheritance diagram for ARLocationPlaceAtLocation:



Public Member Functions

- void SetLocation (Location newLocation)

  Sets the GameObject's location to a new one.

Public Attributes

- Location location

  The location to place the GameObject at.
- bool isHeightRelative = true

  If true, the altitude will be computed as relative to the device level.
- bool showDebugInfoPanel = false

  If true, will display a UI panel with debug information above the object.
- float movementSmoothingFactor = 250.0f

  The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.
- ARLocationManager **manager**

6.9.1 Detailed Description

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

Definition at line 8 of file ARLocationPlaceAtLocation.cs.

6.9.2 Member Function Documentation

6.9.2.1 SetLocation()

```
void ARLocationPlaceAtLocation.SetLocation (
            Location newLocation )
```

Sets the GameObject's location to a new one.

Parameters

newLocation	

Definition at line 63 of file ARLocationPlaceAtLocation.cs.

6.9.3 Member Data Documentation

6.9.3.1 isHeightRelative

```
bool ARLocationPlaceAtLocation.isHeightRelative = true
```

If true, the altitude will be computed as relative to the device level.

Definition at line 19 of file ARLocationPlaceAtLocation.cs.

6.9.3.2 location

```
Location ARLocationPlaceAtLocation.location
```

The location to place the GameObject at.

Definition at line 13 of file ARLocationPlaceAtLocation.cs.

6.9.3.3 movementSmoothingFactor

```
float ARLocationPlaceAtLocation.movementSmoothingFactor = 250.0f
```

The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

Definition at line 31 of file ARLocationPlaceAtLocation.cs.

6.9.3.4 showDebugInfoPanel

```
bool ARLocationPlaceAtLocation.showDebugInfoPanel = false
```

If true, will display a UI panel with debug information above the object.

Definition at line 25 of file ARLocationPlaceAtLocation.cs.

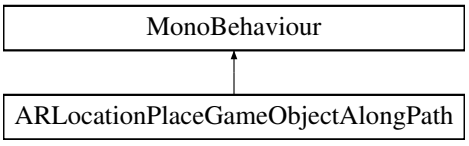
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARLocationPlaceAtLocation.cs

6.10 ARLocationPlaceGameObjectAlongPath 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 ARLocationPlaceGameObjectAlongPath:



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
- bool isHeightRelative = true

  If true, the altitude will be computed as relative to the device level.
- bool showDebugInfoPanel = false

  If true, will display a UI panel with debug information above the object.
- float movementSmoothingFactor = 0

  The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.
- int splineSampleSize = 200

  The size of the sample used to calculate the spline.

6.10.1 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 10 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2 Member Data Documentation

6.10.2.1 isHeightRelative

```
bool ARLocationPlaceGameObjectAlongPath.isHeightRelative = true
```

If true, the altitude will be computed as relative to the device level.

Definition at line 35 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2.2 movementSmoothingFactor

```
float ARLocationPlaceGameObjectAlongPath.movementSmoothingFactor = 0
```

The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

Definition at line 47 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2.3 objectCount

```
int ARLocationPlaceGameObjectAlongPath.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 29 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2.4 path

```
LocationPath ARLocationPlaceGameObjectAlongPath.path
```

The path to place the prefab instances on.

Definition at line 16 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2.5 prefab

```
GameObject ARLocationPlaceGameObjectAlongPath.prefab
```

The prefab/GameObject to be palced along the path.

Definition at line 22 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2.6 showDebugInfoPanel

```
bool ARLocationPlaceGameObjectAlongPath.showDebugInfoPanel = false
```

If true, will display a UI panel with debug information above the object.

Definition at line 41 of file ARLocationPlaceGameObjectAlongPath.cs.

6.10.2.7 splineSampleSize

```
int ARLocationPlaceGameObjectAlongPath.splineSampleSize = 200
```

The size of the sample used to calculate the spline.

Definition at line 53 of file ARLocationPlaceGameObjectAlongPath.cs.

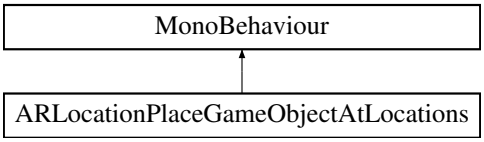
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARLocationPlaceGameObjectAlongPath.cs

6.11 ARLocationPlaceGameObjectAtLocations 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 ARLocationPlaceGameObjectAtLocations:



Public Member Functions

- void AddLocation (Location location)

  Adds a location to the locations list.

Public Attributes

- List< Location > locations

  The locations where the objects will be instantiated.
- GameObject prefab

  The game object that will be instantiated.
- bool isHeightRelative = true

  If true, all altitude will be considered relative to the device.
- bool showDebugInfoPanel = true

  If true, when a `ARLocatedObjectsDebugInfo` is present, a UI Panel with debug information will appear on the top of the object.
- float movementSmoothingFactor = 120f

  The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

6.11.1 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 12 of file ARLocationPlaceGameObjectAtLocations.cs.

6.11.2 Member Function Documentation

6.11.2.1 AddLocation()

```
void ARLocationPlaceGameObjectAtLocations.AddLocation (
            Location location )
```

Adds a location to the locations list.

Parameters

location	

Definition at line 63 of file ARLocationPlaceGameObjectAtLocations.cs.

6.11.3 Member Data Documentation

6.11.3.1 isHeightRelative

```
bool ARLocationPlaceGameObjectAtLocations.isHeightRelative = true
```

If true, all altitude will be considered relative to the device.

Definition at line 30 of file ARLocationPlaceGameObjectAtLocations.cs.

6.11.3.2 locations

```
List<Location> ARLocationPlaceGameObjectAtLocations.locations
```

The locations where the objects will be instantiated.

Definition at line 18 of file ARLocationPlaceGameObjectAtLocations.cs.

6.11.3.3 movementSmoothingFactor

```
float ARLocationPlaceGameObjectAtLocations.movementSmoothingFactor = 120f
```

The smoothing factor for movement due to GPS location adjustments; if set to zero it is disabled.

Definition at line 43 of file ARLocationPlaceGameObjectAtLocations.cs.

6.11.3.4 prefab

```
GameObject ARLocationPlaceGameObjectAtLocations.prefab
```

The game object that will be instantiated.

Definition at line 24 of file ARLocationPlaceGameObjectAtLocations.cs.

6.11.3.5 showDebugInfoPanel

```
bool ARLocationPlaceGameObjectAtLocations.showDebugInfoPanel = true
```

If true, when a `ARLocatedObjectsDebugInfo` is present, a UI Panel with debug information will appear on the top of the object.

Definition at line 37 of file ARLocationPlaceGameObjectAtLocations.cs.

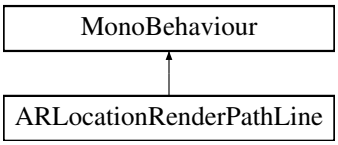
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/ARLocationPlaceGameObjectAtLocations.cs

6.12 ARLocationRenderPathLine Class Reference

This component renders a LocationPath using a given LineRenderer.

Inheritance diagram for ARLocationRenderPathLine:



Public Attributes

- LocationPath locationPath

  The LocationPath describing the path to be traversed.
- bool heightRelativeToDevice = true

  If true, all altitude data is considered relative to the current device elevation.
- 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.
- bool showDebugInfo = true

  Renders debug information. Requires an ARLocationDebugInfo at the ARLocationRoot object.
- LineRenderer lineRenderer

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

6.12.1 Detailed Description

This component renders a LocationPath using a given LineRenderer.

Definition at line 8 of file ARLocationRenderPathLine.cs.

6.12.2 Member Data Documentation

6.12.2.1 heightRelativeToDevice

```
bool ARLocationRenderPathLine.heightRelativeToDevice = true
```

If true, all altitude data is considered relative to the current device elevation.

Definition at line 19 of file ARLocationRenderPathLine.cs.

6.12.2.2 lineRenderer

```
LineRenderer ARLocationRenderPathLine.lineRenderer
```

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

Definition at line 43 of file ARLocationRenderPathLine.cs.

6.12.2.3 lineRenderSampleCount

```
int ARLocationRenderPathLine.lineRenderSampleCount = 250
```

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

Definition at line 31 of file ARLocationRenderPathLine.cs.

6.12.2.4 locationPath

```
LocationPath ARLocationRenderPathLine.locationPath
```

The LocationPath describing the path to be traversed.

Definition at line 13 of file ARLocationRenderPathLine.cs.

6.12.2.5 showDebugInfo

```
bool ARLocationRenderPathLine.showDebugInfo = true
```

Renders debug information. Requires an ARLocationDebugInfo at the ARLocationRoot object.

Definition at line 37 of file ARLocationRenderPathLine.cs.

6.12.2.6 splineSampleCount

```
int ARLocationRenderPathLine.splineSampleCount = 250
```

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

Definition at line 25 of file ARLocationRenderPathLine.cs.

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

- Assets/ARLocation/Scripts/ARLocationRenderPathLine.cs

6.13 CatmullRomCurve Class Reference

A catmull-rom curve.

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.
- Vector3 GetPoint (float u=0)

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

  Calculates the point and the tangent of the curve.
- Vector3 [ ] Sample (int N=100)

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

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

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

  Gets the curve point at a given length.
- 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]`

6.13.1 Detailed Description

A catmull-rom curve.

Definition at line 18 of file CatmullRomCurve.cs.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 CatmullRomCurve()

```
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 159 of file CatmullRomCurve.cs.

6.13.3 Member Function Documentation

6.13.3.1 EstimateLength()

```
float CatmullRomCurve.EstimateLength (
            int N = 100 )
```

Returns the estimated length.

Returns

The length.

Parameters

N	N.

Definition at line 286 of file CatmullRomCurve.cs.

6.13.3.2 GetParameterForLength()

```
float CatmullRomCurve.GetParameterForLength (
            float s )
```

Gets the curve parameter for a given length.

Returns

The parameter for length.

Parameters

s	S.

Definition at line 301 of file CatmullRomCurve.cs.

6.13.3.3 GetPoint()

```
Vector3 CatmullRomCurve.GetPoint (
            float u = 0 )
```

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

Definition at line 187 of file CatmullRomCurve.cs.

6.13.3.4 GetPointAndTangent()

```
CurvePointData CatmullRomCurve.GetPointAndTangent (
            float u = 0 )
```

Calculates the point and the tangent of the curve.

Parameters

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

Returns

Definition at line 209 of file CatmullRomCurve.cs.

6.13.3.5 GetPointAndTangentAtLength()

```
CurvePointData CatmullRomCurve.GetPointAndTangentAtLength (
            float s )
```

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

Parameters

s	

Returns

Definition at line 336 of file CatmullRomCurve.cs.

6.13.3.6 GetPointAtLength()

```
Vector3 CatmullRomCurve.GetPointAtLength (
            float s )
```

Gets the curve point at a given length.

Returns

The point at length.

Parameters

s	S.

Definition at line 325 of file CatmullRomCurve.cs.

6.13.3.7 Sample()

```
Vector3 [] CatmullRomCurve.Sample (
            int N = 100 )
```

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.

Definition at line 245 of file CatmullRomCurve.cs.

6.13.4 Property Documentation

6.13.4.1 Alpha

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

Gets or sets the alpha.

The alpha.

Definition at line 76 of file CatmullRomCurve.cs.

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

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

6.14 CatmullRomSpline Class Reference

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

Public Member Functions

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

  Creates a new Catmull-rom spline.
- void CalculateSegments (int N=250)

  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.

Properties

- Vector3 [ ] points `[get]`

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

  The full (estimated) length of the spline.
- float Alpha `[get, set]`

  The alpha/tension parameter of the spline.

6.14.1 Detailed Description

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

Definition at line 8 of file CatmullRomSpline.cs.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 CatmullRomSpline()

```
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.

Definition at line 69 of file CatmullRomSpline.cs.

6.14.3 Member Function Documentation

6.14.3.1 CalculateSegments()

```
void CatmullRomSpline.CalculateSegments (
            int N = 250 )
```

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

Parameters

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

Definition at line 81 of file CatmullRomSpline.cs.

6.14.3.2 DrawCurveWithLineRenderer()

```
void CatmullRomSpline.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	

Definition at line 156 of file CatmullRomSpline.cs.

6.14.3.3 DrawGizmos()

```
void CatmullRomSpline.DrawGizmos ( )
```

Draw the curve and sample point using Gizmos.

Definition at line 207 of file CatmullRomSpline.cs.

6.14.3.4 GetPointAndTangentAtArcLength()

```
CurvePointData CatmullRomSpline.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 135 of file CatmullRomSpline.cs.

6.14.3.5 GetPointAtArcLength()

```
Vector3 CatmullRomSpline.GetPointAtArcLength (
            float s )
```

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

Parameters

s	The arc-length.

Returns

Definition at line 114 of file CatmullRomSpline.cs.

6.14.3.6 SamplePoints() [1/2]

```
Vector3 [] CatmullRomSpline.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 179 of file CatmullRomSpline.cs.

6.14.3.7 SamplePoints() [2/2]

```
Vector3 [] CatmullRomSpline.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 199 of file CatmullRomSpline.cs.

6.14.4 Property Documentation

6.14.4.1 Alpha

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

The alpha/tension parameter of the spline.

Definition at line 49 of file CatmullRomSpline.cs.

6.14.4.2 Length

```
float CatmullRomSpline.Length  [get]
```

The full (estimated) length of the spline.

Definition at line 43 of file CatmullRomSpline.cs.

6.14.4.3 points

```
Vector3 [] CatmullRomSpline.points  [get]
```

The points interpolated of the spline.

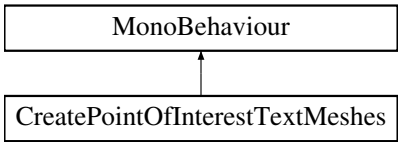
Definition at line 13 of file CatmullRomSpline.cs.

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

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

6.15 CreatePointOfInterestTextMeshes Class Reference

Inheritance diagram for 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**

6.15.1 Detailed Description

Definition at line 39 of file CreatePointOfInterestTextMeshes.cs.

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

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

6.16 CurvePointData Struct Reference

A struct holding a pair of point/tangent values.

Public Attributes

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

6.16.1 Detailed Description

A struct holding a pair of point/tangent values.

Definition at line 8 of file CatmullRomCurve.cs.

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

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

6.17 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 x=0.0, double y=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.

6.17.1 Detailed Description

Definition at line 7 of file DVector2.cs.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 DVector2()

```
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 51 of file DVector2.cs.

6.17.3 Member Function Documentation

6.17.3.1 Distance()

```
static double 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 124 of file DVector2.cs.

6.17.3.2 Dot()

```
static double 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 113 of file DVector2.cs.

6.17.3.3 Equals()

```
bool 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 72 of file DVector2.cs.

6.17.3.4 Lerp()

```
static DVector2 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 136 of file DVector2.cs.

6.17.3.5 Normalize()

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

Normalize this instance.

Definition at line 80 of file DVector2.cs.

6.17.3.6 operator∗()

```
static DVector2 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 148 of file DVector2.cs.

6.17.3.7 operator+()

```
static DVector2 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 176 of file DVector2.cs.

6.17.3.8 operator-()

```
static DVector2 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 190 of file DVector2.cs.

6.17.3.9 operator/()

```
static DVector2 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 162 of file DVector2.cs.

6.17.3.10 Set()

```
void DVector2.Set (
            double x = 0.0,
            double y = 0.0 )
```

Set the specified x and y.

Parameters

x	The x coordinate.
y	The y coordinate.

Definition at line 92 of file DVector2.cs.

6.17.3.11 ToString()

```
override string 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 102 of file DVector2.cs.

6.17.3.12 toVector2()

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

Converts to a Vector2.

Returns

The vector2.

Definition at line 61 of file DVector2.cs.

6.17.4 Property Documentation

6.17.4.1 magnitude

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

Gets the magnitude of the vector.

The magnitude.

Definition at line 17 of file DVector2.cs.

6.17.4.2 normalized

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

Gets the normalized version of this vector.

The normalized.

Definition at line 29 of file DVector2.cs.

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

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

6.18 DVector3 Struct Reference

Public Member Functions

- DVector3 (double x=0.0, double y=0.0, double z=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 x=0.0, double y=0.0, double z=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/ (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.

6.18.1 Detailed Description

Definition at line 7 of file DVector3.cs.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 DVector3()

```
DVector3.DVector3 (
            double x = 0.0,
            double y = 0.0,
            double z = 0.0 )
```

Initializes a new instance of the T:DVector3 struct.

Parameters

x	The x coordinate.
y	The y coordinate.

Definition at line 49 of file DVector3.cs.

6.18.3 Member Function Documentation

6.18.3.1 Distance()

```
static double 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 125 of file DVector3.cs.

6.18.3.2 Dot()

```
static double 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 114 of file DVector3.cs.

6.18.3.3 Equals()

```
bool 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 71 of file DVector3.cs.

6.18.3.4 Lerp()

```
static DVector3 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 137 of file DVector3.cs.

6.18.3.5 Normalize()

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

Normalize this instance.

Definition at line 79 of file DVector3.cs.

6.18.3.6 operator∗()

```
static DVector3 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 149 of file DVector3.cs.

6.18.3.7 operator+()

```
static DVector3 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 179 of file DVector3.cs.

6.18.3.8 operator-()

```
static DVector3 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 194 of file DVector3.cs.

6.18.3.9 operator/()

```
static DVector3 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 164 of file DVector3.cs.

6.18.3.10 Set()

```
void DVector3.Set (
            double x = 0.0,
            double y = 0.0,
            double z = 0.0 )
```

Set the specified x and y.

Parameters

x	The x coordinate.
y	The y coordinate.

Definition at line 92 of file DVector3.cs.

6.18.3.11 ToString()

```
override string 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 103 of file DVector3.cs.

6.18.3.12 toVector3()

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

Converts to a Vector3.

Returns

The vector2.

Definition at line 60 of file DVector3.cs.

6.18.4 Property Documentation

6.18.4.1 magnitude

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

Gets the magnitude of the vector.

The magnitude.

Definition at line 18 of file DVector3.cs.

6.18.4.2 normalized

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

Gets the normalized version of this vector.

The normalized.

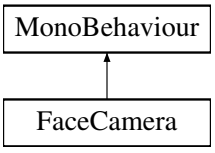
Definition at line 30 of file DVector3.cs.

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

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

6.19 FaceCamera Class Reference

Inheritance diagram for FaceCamera:



6.19.1 Detailed Description

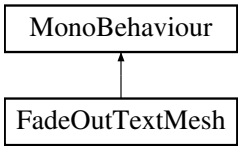
Definition at line 5 of file FaceCamera.cs.

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

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

6.20 FadeOutTextMesh Class Reference

Inheritance diagram for FadeOutTextMesh:



Public Attributes

- float **duration** = 2.0f

6.20.1 Detailed Description

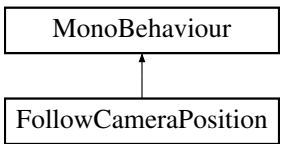
Definition at line 5 of file FadeOutTextMesh.cs.

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

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

6.21 FollowCameraPosition Class Reference

Inheritance diagram for FollowCameraPosition:



6.21.1 Detailed Description

Definition at line 5 of file FollowCameraPosition.cs.

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

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

6.22 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)

  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 accound.
- static Vector3 GetGameObjectPositionForLocation (Vector3 userPosition, Location userLocation, Location objectLocation, bool heightIsRelative)

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

  Gets the game object position for location.
- static void PlaceGameObjectAtLocation (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

  The latitude, in degrees.
- double longitude

  The longitude, in degrees.
- double altitude

  The altitude, in meters.
- bool ignoreAltitude = false

  If true, the altitude will be ignored when placing an object, and the object's will be placed at the same elevation as the device.
- string label = ""

  An optional label to the location

Properties

- DVector2 horizontalVector [get]

  Gets the horizontal vector.

6.22.1 Detailed Description

Represents a geographical location.

Definition at line 10 of file Location.cs.

6.22.2 Member Function Documentation

6.22.2.1 Clone()

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

Clones this instance.

Returns

The clone.

Definition at line 66 of file Location.cs.

6.22.2.2 DistanceWithAltitude()

```
static double 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 111 of file Location.cs.

6.22.2.3 GetGameObjectPositionForLocation() [1/2]

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

Gets the game object position for location.

Parameters

userPosition	
userLocation	
objectLocation	
heightIsRelative	

Returns

Definition at line 157 of file Location.cs.

6.22.2.4 GetGameObjectPositionForLocation() [2/2]

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

Gets the game object position for location.

Returns

The game object position for location.

Parameters

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

Definition at line 172 of file Location.cs.

6.22.2.5 HorizontalDistance()

```
static double Location.HorizontalDistance (
            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 91 of file Location.cs.

6.22.2.6 HorizontalVectorFromTo()

```
static DVector2 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 125 of file Location.cs.

6.22.2.7 PlaceGameObjectAtLocation()

```
static void Location.PlaceGameObjectAtLocation (
            Transform transform,
            Transform user,
            Location userLocation,
            Location objectLocation,
            bool heightIsRelative )  [static]
```

Places the game object at location.

Parameters

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

Definition at line 185 of file Location.cs.

6.22.2.8 VectorFromTo()

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

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

Returns

The from to.

Parameters

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

Definition at line 141 of file Location.cs.

6.22.3 Member Data Documentation

6.22.3.1 altitude

```
double Location.altitude
```

The altitude, in meters.

Definition at line 28 of file Location.cs.

6.22.3.2 ignoreAltitude

```
bool Location.ignoreAltitude = false
```

If true, the altitude will be ignored when placing an object, and the object's will be placed at the same elevation as the device.

Definition at line 35 of file Location.cs.

6.22.3.3 label

```
string Location.label = ""
```

An optional label to the location

Definition at line 41 of file Location.cs.

6.22.3.4 latitude

```
double Location.latitude
```

The latitude, in degrees.

Definition at line 16 of file Location.cs.

6.22.3.5 longitude

```
double Location.longitude
```

The longitude, in degrees.

Definition at line 22 of file Location.cs.

6.22.4 Property Documentation

6.22.4.1 horizontalVector

```
DVector2 Location.horizontalVector  [get]
```

Gets the horizontal vector.

The horizontal vector.

Definition at line 48 of file Location.cs.

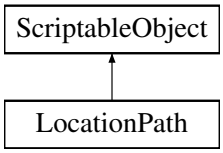
The documentation for this class was generated from the following file:

- Assets/ARLocation/Scripts/Location.cs

6.23 LocationPath Class Reference

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

Inheritance diagram for LocationPath:



Public Attributes

- Location [ ] locations

  The geographical locations that the path will interpolate.
- 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.

6.23.1 Detailed Description

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

Definition at line 10 of file LocationPath.cs.

6.23.2 Member Data Documentation

6.23.2.1 alpha

```
float LocationPath.alpha = 0.5f
```

The path's alpha/tension factor.

Definition at line 22 of file LocationPath.cs.

6.23.2.2 locations

```
Location [] LocationPath.locations
```

The geographical locations that the path will interpolate.

Definition at line 16 of file LocationPath.cs.

6.23.2.3 sceneViewScale

```
float LocationPath.sceneViewScale = 1.0f
```

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

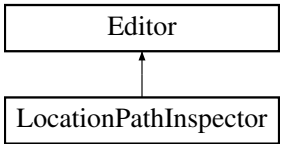
Definition at line 27 of file LocationPath.cs.

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

- Assets/ARLocation/Scripts/LocationPath.cs

6.24 LocationPathInspector Class Reference

Inheritance diagram for LocationPathInspector:



Public Member Functions

- override void **OnInspectorGUI** ()

6.24.1 Detailed Description

Definition at line 7 of file LocationPathInspector.cs.

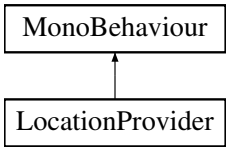
The documentation for this class was generated from the following file:

- Assets/ARLocation/Editor/LocationPathInspector.cs

6.25 LocationProvider Class Reference

This Component manages the initialization and management of location and heading data.

Inheritance diagram for LocationProvider:



Public Member Functions

- delegate void LocationUpdatedDelegate (Location currentLocation, Location lastLocation, Vector3 position↩ Delta, float accuracy)

  Delegate to be called when the device's location has been updated.
- delegate void CompassUpdateDelegate (double trueHeading)

  Delegate to be called when the device's heading has been updated.
- void UpdateMockLocation (Location loc)

  Set the mock location. For in-editor simulator usage.
- void onLocationUpdated (LocationUpdatedDelegate del)

  Register a delegate to be called when the device's location is updated.
- void onCompassUpdated (CompassUpdateDelegate del)

  Register a delegate to be called when the device's heading is updated.

Public Attributes

- int maxStartWaitTime = 20

  The maximum time to wait for location services to initialize."
- Location mockLocation

  A mock location for use in the editor simulator.
- float desiredAccuracyInMeters = 0.1f

  The desired location accuracy, in meters.
- float updateDistanceInMeters = 0.1f

  Hint how often location updates are provided by givin an update distance, in meters.

Properties

- bool isEnabled [get]

  If true, the location services have been started and the LocationProvider is fully enabled.
- float TimeSinceStart [get]

  How much time has passed since the locations services have been enabled, in seconds.
- Location currentLocation [get]

  The last measured device location.
- Location lastLocation [get]

  The previous device location.
- Vector3 currentPositionDelta [get]

  The world space vector representing the device's last translation (that is, from the lastLocation to the currentLocation), in meters.
- float currentHorizontalAccuracy [get]

  Returns the horizontal accuracy of the current location measurement.

6.25.1 Detailed Description

This Component manages the initialization and management of location and heading data.

Definition at line 11 of file LocationProvider.cs.

6.25.2 Member Function Documentation

6.25.2.1 CompassUpdateDelegate()

```
delegate void LocationProvider.CompassUpdateDelegate (
            double trueHeading )
```

Delegate to be called when the device's heading has been updated.

Parameters

trueHeading	

6.25.2.2 LocationUpdatedDelegate()

```
delegate void LocationProvider.LocationUpdatedDelegate (
            Location currentLocation,
            Location lastLocation,
            Vector3 positionDelta,
            float accuracy )
```

Delegate to be called when the device's location has been updated.

Parameters

currentLocation	
lastLocation	
positionDelta	
accuracy	

6.25.2.3 onCompassUpdated()

```
void LocationProvider.onCompassUpdated (
            CompassUpdateDelegate del )
```

Register a delegate to be called when the device's heading is updated.

Parameters

del	

Definition at line 299 of file LocationProvider.cs.

6.25.2.4 onLocationUpdated()

```
void LocationProvider.onLocationUpdated (
            LocationUpdatedDelegate del )
```

Register a delegate to be called when the device's location is updated.

Parameters

del	

Definition at line 290 of file LocationProvider.cs.

6.25.2.5 UpdateMockLocation()

```
void LocationProvider.UpdateMockLocation (
            Location loc )
```

Set the mock location. For in-editor simulator usage.

Parameters

loc	

Definition at line 219 of file LocationProvider.cs.

6.25.3 Member Data Documentation

6.25.3.1 desiredAccuracyInMeters

```
float LocationProvider.desiredAccuracyInMeters = 0.1f
```

The desired location accuracy, in meters.

Definition at line 30 of file LocationProvider.cs.

6.25.3.2 maxStartWaitTime

```
int LocationProvider.maxStartWaitTime = 20
```

The maximum time to wait for location services to initialize."

Definition at line 17 of file LocationProvider.cs.

6.25.3.3 mockLocation

```
Location LocationProvider.mockLocation
```

A mock location for use in the editor simulator.

Definition at line 23 of file LocationProvider.cs.

6.25.3.4 updateDistanceInMeters

```
float LocationProvider.updateDistanceInMeters = 0.1f
```

Hint how often location updates are provided by givin an update distance, in meters.

Definition at line 36 of file LocationProvider.cs.

6.25.4 Property Documentation

6.25.4.1 currentHorizontalAccuracy

```
float LocationProvider.currentHorizontalAccuracy  [get]
```

Returns the horizontal accuracy of the current location measurement.

Definition at line 118 of file LocationProvider.cs.

6.25.4.2 currentLocation

```
Location LocationProvider.currentLocation  [get]
```

The last measured device location.

Definition at line 85 of file LocationProvider.cs.

6.25.4.3 currentPositionDelta

```
Vector3 LocationProvider.currentPositionDelta  [get]
```

The world space vector representing the device's last translation (that is, from the lastLocation to the current↩ Location), in meters.

Definition at line 107 of file LocationProvider.cs.

6.25.4.4 isEnabled

```
bool LocationProvider.isEnabled  [get]
```

If true, the location services have been started and the LocationProvider is fully enabled.

Definition at line 56 of file LocationProvider.cs.

6.25.4.5 lastLocation

```
Location LocationProvider.lastLocation  [get]
```

The previous device location.

Definition at line 96 of file LocationProvider.cs.

6.25.4.6 TimeSinceStart

```
float LocationProvider.TimeSinceStart  [get]
```

How much time has passed since the locations services have been enabled, in seconds.

Definition at line 75 of file LocationProvider.cs.

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

- Assets/ARLocation/Scripts/LocationProvider.cs

6.26 OpenStreetMapOptions Class Reference

Public Attributes

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

6.26.1 Detailed Description

Definition at line 27 of file CreatePointOfInterestTextMeshes.cs.

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

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

6.27 OverpassRequestData Class Reference

Public Attributes

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

6.27.1 Detailed Description

Definition at line 17 of file CreatePointOfInterestTextMeshes.cs.

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

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

6.28 POIData Class Reference

Public Attributes

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

6.28.1 Detailed Description

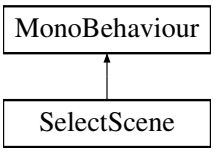
Definition at line 10 of file CreatePointOfInterestTextMeshes.cs.

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

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

6.29 SelectScene Class Reference

Inheritance diagram for SelectScene:



Public Member Functions

- void **LoadScene** (string name)

6.29.1 Detailed Description

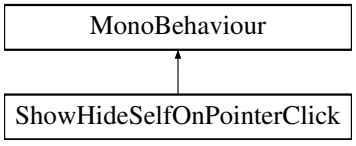
Definition at line 6 of file SelectScene.cs.

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

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

6.30 ShowHideSelfOnPointerClick Class Reference

Inheritance diagram for ShowHideSelfOnPointerClick:



6.30.1 Detailed Description

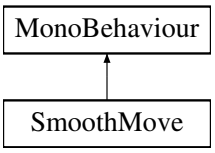
Definition at line 5 of file ShowHideSelfOnPointerClick.cs.

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

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

6.31 SmoothMove Class Reference

Inheritance diagram for SmoothMove:



Public Attributes

- float **smoothing** = 120f

Properties

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

6.31.1 Detailed Description

Definition at line 5 of file SmoothMove.cs.

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

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

6.32 Utils Class Reference

Static Public Member Functions

- static bool **IsARDevice** ()
- static float **FloatListAverage** (List< float > list)
- static float **GetNormalizedDegrees** (float value)

6.32.1 Detailed Description

Definition at line 5 of file Utils.cs.

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

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