ProjNet4GeoAPI-For-Unity

1.0.0

Generated by Doxygen 1.9.7

Chapter 1

Namespace Index

1.1 Package List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 GeocoordinateTransformer Namespace Reference

Classes

- class CoordinateTransformer

  Contains the coordinates UTMCoordinates which correspond to the origin of Unity's coordinate system, the point's WGS84/UTM coordinate reference system UTMCrs, and methods for the transformation of coordinates.
- class CoordinateTransformerTest

  Test class to demonstrate the functionality of CoordinateTransformer. Contains necessary test variables and methods that can be changed and called from Unity's Inspector via the context menu.
- class GeographicCoordinates

  Contains a latitude and longitude coordinate tuple and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).
- class UTMCoordinates

  Contains an easting and northing planar coordinate tuple in a WGS84/UTM coordinate reference system and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).
- class UTMCrs

  Contains all variables to specify a WGS84/UTM coordinate refernece system.

Enumerations

- enum Hemispheres { Northern , Southern }

5.1.1 Enumeration Type Documentation

5.1.1.1 Hemispheres

```
enum GeocoordinateTransformer.Hemispheres
```

Enumerator

Northern	
Southern	

Definition at line 71 of file UTMCrs.cs.

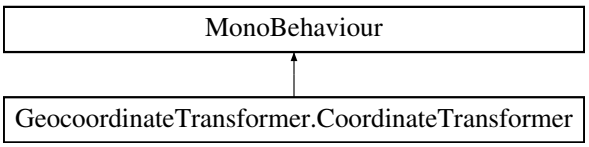
Chapter 6

Class Documentation

6.1 GeocoordinateTransformer.CoordinateTransformer Class Reference

Contains the coordinates UTMCoordinates which correspond to the origin of Unity's coordinate system, the point's WGS84/UTM coordinate reference system UTMCrs, and methods for the transformation of coordinates.

Inheritance diagram for GeocoordinateTransformer.CoordinateTransformer:



Public Member Functions

- override bool Equals (object other)
- override int GetHashCode ()
- override string ToString ()
- GeographicCoordinates GetGeographicCoordinates (UTMCoordinates utmCoordinates)

  Converts GeocoordinateTransformer.UTMCoordinates into GeocoordinateTransformer.GeographicCoordinates.
- GeographicCoordinates GetGeographicCoordinates (Vector3 unityCoordinates)

  Converts Unity Vector3 coordinates into GeocoordinateTransformer.GeographicCoordinates.
- UTMCoordinates GetUTMCoordinates (GeographicCoordinates geographicCoordinates)

  Converts GeocoordinateTransformer.GeographicCoordinates into GeocoordinateTransformer.UTMCoordinates.
- Vector3 GetUnityCoordinates (GeographicCoordinates geographicCoordinates)

  Converts GeocoordinateTransformer.GeographicCoordinates into Unity Vector3 coordinates.
- Vector3 GetUnityCoordinates (UTMCoordinates utmCoordinates)

  Converts GeocoordinateTransformer.UTMCoordinates into Unity Vector3 coordinates.
- UTMCoordinates GetUTMCoordinates (Vector3 unityCoordinates)

  Converts Unity Vector3 coordinates into GeocoordinateTransformer.UTMCoordinates.

Public Attributes

- UTMCrs UTMCrs = new(utmZone: 32, hemispheres: Hemispheres.Northern)

  Contains the WGS84/UTM coordinate reference system of the UTMCoordinates.
- UTMCoordinates UTMCoordinates = new(east: 566600, north: 5933000, altitude: 0)

  Contains the WGS84/UTM coordinates that correspond to the origin of Unity's coordinate system.

Static Public Attributes

- static CoordinateTransformer Instance

  Unity CoordinateTransformer Singleton instance.

6.1.1 Detailed Description

Contains the coordinates UTMCoordinates which correspond to the origin of Unity's coordinate system, the point's WGS84/UTM coordinate reference system UTMCrs, and methods for the transformation of coordinates.

Definition at line 13 of file CoordinateTransformer.cs.

6.1.2 Member Function Documentation

6.1.2.1 Equals()

```
override bool GeocoordinateTransformer.CoordinateTransformer.Equals (
            object other )
```

Definition at line 28 of file CoordinateTransformer.cs.

6.1.2.2 GetGeographicCoordinates() [1/2]

```
GeographicCoordinates GeocoordinateTransformer.CoordinateTransformer.GetGeographicCoordinates
(
            UTMCoordinates utmCoordinates )
```

Converts GeocoordinateTransformer.UTMCoordinates into GeocoordinateTransformer.GeographicCoordinates.

Parameters

utmCoordinates	Coordinates to be transformed.

Returns

GeocoordinateTransformer.GeographicCoordinates tranformation result.

Definition at line 121 of file CoordinateTransformer.cs.

6.1.2.3 GetGeographicCoordinates() [2/2]

```
GeographicCoordinates GeocoordinateTransformer.CoordinateTransformer.GetGeographicCoordinates
(
            Vector3 unityCoordinates )
```

Converts Unity Vector3 coordinates into GeocoordinateTransformer.GeographicCoordinates.

Parameters

unityCoordinates	Coordinates to be transformed.

Returns

GeocoordinateTransformer.GeographicCoordinates tranformation result.

Definition at line 133 of file CoordinateTransformer.cs.

6.1.2.4 GetHashCode()

```
override int GeocoordinateTransformer.CoordinateTransformer.GetHashCode ( )
```

Definition at line 40 of file CoordinateTransformer.cs.

6.1.2.5 GetUnityCoordinates() [1/2]

```
Vector3 GeocoordinateTransformer.CoordinateTransformer.GetUnityCoordinates (
            GeographicCoordinates geographicCoordinates )
```

Converts GeocoordinateTransformer.GeographicCoordinates into Unity Vector3 coordinates.

Parameters

geographicCoordinates	Coordinates to be transformed.

Returns

Unity Vector3 tranformation result.

Definition at line 157 of file CoordinateTransformer.cs.

6.1.2.6 GetUnityCoordinates() [2/2]

```
Vector3 GeocoordinateTransformer.CoordinateTransformer.GetUnityCoordinates (
            UTMCoordinates utmCoordinates )
```

Converts GeocoordinateTransformer.UTMCoordinates into Unity Vector3 coordinates.

Parameters

utmCoordinates	Coordinates to be transformed.

Returns

Unity Vector3 tranformation result.

Exceptions

System.ArgumentException	Thrown when a coordinate component exceeds a distance of 100 km on one of the axes. The latter exceeds the capacity of a float.

Calculate the possition relative to the

See also

SceneOriginUTMCoordinates

and check if the calculated coordinate component value can be processed by Unity.

Definition at line 171 of file CoordinateTransformer.cs.

6.1.2.7 GetUTMCoordinates() [1/2]

```
UTMCoordinates GeocoordinateTransformer.CoordinateTransformer.GetUTMCoordinates (
            GeographicCoordinates geographicCoordinates )
```

Converts GeocoordinateTransformer.GeographicCoordinates into GeocoordinateTransformer.UTMCoordinates.

Parameters

geographicCoordinates	Coordinates to be transformed.

Returns

GeocoordinateTransformer.UTMCoordinates tranformation result.

Definition at line 145 of file CoordinateTransformer.cs.

6.1.2.8 GetUTMCoordinates() [2/2]

```
UTMCoordinates GeocoordinateTransformer.CoordinateTransformer.GetUTMCoordinates (
            Vector3 unityCoordinates )
```

Converts Unity Vector3 coordinates into GeocoordinateTransformer.UTMCoordinates.

Parameters

unityCoordinates	Coordinates to be transformed.

Returns

GeocoordinateTransformer.UTMCoordinates tranformation result.

Definition at line 201 of file CoordinateTransformer.cs.

6.1.2.9 ToString()

```
override string GeocoordinateTransformer.CoordinateTransformer.ToString ( )
```

Definition at line 45 of file CoordinateTransformer.cs.

6.1.3 Member Data Documentation

6.1.3.1 Instance

```
CoordinateTransformer GeocoordinateTransformer.CoordinateTransformer.Instance  [static]
```

Unity CoordinateTransformer Singleton instance.

Definition at line 227 of file CoordinateTransformer.cs.

6.1.3.2 UTMCoordinates

```
UTMCoordinates GeocoordinateTransformer.CoordinateTransformer.UTMCoordinates = new(east↩
:  566600, north:  5933000, altitude:  0)
```

Contains the WGS84/UTM coordinates that correspond to the origin of Unity's coordinate system.

Make sure that the provided point is within a distance smaller than 100 km of the spatial data you want to use in the scene.

Definition at line 25 of file CoordinateTransformer.cs.

6.1.3.3 UTMCrs

```
UTMCrs GeocoordinateTransformer.CoordinateTransformer.UTMCrs = new(utmZone:  32, hemispheres:
Hemispheres.Northern)
```

Contains the WGS84/UTM coordinate reference system of the UTMCoordinates.

Definition at line 18 of file CoordinateTransformer.cs.

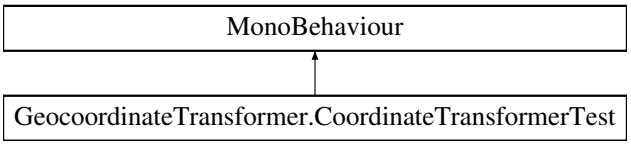
The documentation for this class was generated from the following file:

- CoordinateTransformer.cs

6.2 GeocoordinateTransformer.CoordinateTransformerTest Class Reference

Test class to demonstrate the functionality of CoordinateTransformer. Contains necessary test variables and methods that can be changed and called from Unity's Inspector via the context menu.

Inheritance diagram for GeocoordinateTransformer.CoordinateTransformerTest:



Properties

- CoordinateTransformer CoordinateTransformer `[get, set]`

  Contains the GeocoordinateTransformer.CoordinateTransformer that was created in the Unity Scene.

6.2.1 Detailed Description

Test class to demonstrate the functionality of CoordinateTransformer. Contains necessary test variables and methods that can be changed and called from Unity's Inspector via the context menu.

Definition at line 9 of file CoordinateTransformerTest.cs.

6.2.2 Property Documentation

6.2.2.1 CoordinateTransformer

```
CoordinateTransformer GeocoordinateTransformer.CoordinateTransformerTest.CoordinateTransformer
[get], [set]
```

Contains the GeocoordinateTransformer.CoordinateTransformer that was created in the Unity Scene.

Exceptions

System.ApplicationException	Thrown when there was no GeocoordinateTransformer.CoordinateTransformer could be found in the Unity Scene.

Definition at line 21 of file CoordinateTransformerTest.cs.

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

- CoordinateTransformerTest.cs

6.3 GeocoordinateTransformer.GeographicCoordinates Class Reference

Contains a latitude and longitude coordinate tuple and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Public Member Functions

- GeographicCoordinates (double latitude, double longitude, double altitude)
- override bool Equals (object other)
- override int GetHashCode ()
- override string ToString ()

Public Attributes

- double latitude

  Latitude value of the coordinate tuple.
- double longitude

  Longitude value of the coordinate tuple.
- double altitude

  Altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

6.3.1 Detailed Description

Contains a latitude and longitude coordinate tuple and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Definition at line 10 of file GeographicCoordinates.cs.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 GeographicCoordinates()

```
GeocoordinateTransformer.GeographicCoordinates.GeographicCoordinates (
            double latitude,
            double longitude,
            double altitude )
```

Definition at line 28 of file GeographicCoordinates.cs.

6.3.3 Member Function Documentation

6.3.3.1 Equals()

```
override bool GeocoordinateTransformer.GeographicCoordinates.Equals (
            object other )
```

Definition at line 36 of file GeographicCoordinates.cs.

6.3.3.2 GetHashCode()

```
override int GeocoordinateTransformer.GeographicCoordinates.GetHashCode ( )
```

Definition at line 49 of file GeographicCoordinates.cs.

6.3.3.3 ToString()

```
override string GeocoordinateTransformer.GeographicCoordinates.ToString ( )
```

Definition at line 54 of file GeographicCoordinates.cs.

6.3.4 Member Data Documentation

6.3.4.1 altitude

```
double GeocoordinateTransformer.GeographicCoordinates.altitude
```

Altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Note that the altitude value is not associated with the WGS84/UTM coordinate reference system.

Definition at line 26 of file GeographicCoordinates.cs.

6.3.4.2 latitude

```
double GeocoordinateTransformer.GeographicCoordinates.latitude
```

Latitude value of the coordinate tuple.

Definition at line 15 of file GeographicCoordinates.cs.

6.3.4.3 longitude

```
double GeocoordinateTransformer.GeographicCoordinates.longitude
```

Longitude value of the coordinate tuple.

Definition at line 20 of file GeographicCoordinates.cs.

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

- GeographicCoordinates.cs

6.4 GeocoordinateTransformer.UTMCoordinates Class Reference

Contains an easting and northing planar coordinate tuple in a WGS84/UTM coordinate reference system and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Public Member Functions

- UTMCoordinates (double east, double north, double altitude)
- override bool Equals (object other)
- override int GetHashCode ()
- override string ToString ()

Public Attributes

- double east

  Easting value of the coordinate tuple.
- double north

  Northing value of the coordinate tuple.
- double altitude

  Altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

6.4.1 Detailed Description

Contains an easting and northing planar coordinate tuple in a WGS84/UTM coordinate reference system and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Definition at line 9 of file UTMCoordinates.cs.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 UTMCoordinates()

```
GeocoordinateTransformer.UTMCoordinates.UTMCoordinates (
            double east,
            double north,
            double altitude )
```

Definition at line 28 of file UTMCoordinates.cs.

6.4.3 Member Function Documentation

6.4.3.1 Equals()

```
override bool GeocoordinateTransformer.UTMCoordinates.Equals (
            object other )
```

Definition at line 36 of file UTMCoordinates.cs.

6.4.3.2 GetHashCode()

```
override int GeocoordinateTransformer.UTMCoordinates.GetHashCode ( )
```

Definition at line 49 of file UTMCoordinates.cs.

6.4.3.3 ToString()

```
override string GeocoordinateTransformer.UTMCoordinates.ToString ( )
```

Definition at line 54 of file UTMCoordinates.cs.

6.4.4 Member Data Documentation

6.4.4.1 altitude

```
double GeocoordinateTransformer.UTMCoordinates.altitude
```

Altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Note that the altitude value is not associated with the WGS84/UTM coordinate reference system.

Definition at line 25 of file UTMCoordinates.cs.

6.4.4.2 east

```
double GeocoordinateTransformer.UTMCoordinates.east
```

Easting value of the coordinate tuple.

Definition at line 14 of file UTMCoordinates.cs.

6.4.4.3 north

```
double GeocoordinateTransformer.UTMCoordinates.north
```

Northing value of the coordinate tuple.

Definition at line 19 of file UTMCoordinates.cs.

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

- UTMCoordinates.cs

6.5 GeocoordinateTransformer.UTMCrs Class Reference

Contains all variables to specify a WGS84/UTM coordinate refernece system.

Public Member Functions

- UTMCrs (int utmZone, Hemispheres hemispheres)
- bool IsNorthernHemisphere ()

  Returns true if the zone is on the northern hemisphere. Otherwise it returns false.
- override bool Equals (object other)
- override int GetHashCode ()
- override string ToString ()

Public Attributes

- int UTMZone = 32

  Specifies the zone of the WGS84/UTM coordinate system (between 1 and 60).
- Hemispheres Hemisphere = Hemispheres.Northern

  Hemisphere of the region. Choose between the northern or southern hemisphere.

6.5.1 Detailed Description

Contains all variables to specify a WGS84/UTM coordinate refernece system.

Definition at line 10 of file UTMCrs.cs.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 UTMCrs()

```
GeocoordinateTransformer.UTMCrs.UTMCrs (
            int utmZone,
            Hemispheres hemispheres )
```

Definition at line 26 of file UTMCrs.cs.

6.5.3 Member Function Documentation

6.5.3.1 Equals()

```
override bool GeocoordinateTransformer.UTMCrs.Equals (
            object other )
```

Definition at line 48 of file UTMCrs.cs.

6.5.3.2 GetHashCode()

```
override int GeocoordinateTransformer.UTMCrs.GetHashCode ( )
```

Definition at line 60 of file UTMCrs.cs.

6.5.3.3 IsNorthernHemisphere()

```
bool GeocoordinateTransformer.UTMCrs.IsNorthernHemisphere ( )
```

Returns true if the zone is on the northern hemisphere. Otherwise it returns false.

Returns

The default value is true.

Definition at line 38 of file UTMCrs.cs.

6.5.3.4 ToString()

```
override string GeocoordinateTransformer.UTMCrs.ToString ( )
```

Definition at line 65 of file UTMCrs.cs.

6.5.4 Member Data Documentation

6.5.4.1 Hemisphere

```
Hemispheres GeocoordinateTransformer.UTMCrs.Hemisphere = Hemispheres.Northern
```

Hemisphere of the region. Choose between the northern or southern hemisphere.

The default hemisphere setting is Northern as it is suitable for Hamburg in Germany.

Definition at line 24 of file UTMCrs.cs.

6.5.4.2 UTMZone

```
int GeocoordinateTransformer.UTMCrs.UTMZone = 32
```

Specifies the zone of the WGS84/UTM coordinate system (between 1 and 60).

The default WGS84/UTM coordinate system is 32 which is most suitable for Hamburg in Germany.

Definition at line 17 of file UTMCrs.cs.

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

- UTMCrs.cs

Chapter 7

File Documentation

7.1 CoordinateTransformer.cs File Reference

Classes

- class GeocoordinateTransformer.CoordinateTransformer

  Contains the coordinates UTMCoordinates which correspond to the origin of Unity's coordinate system, the point's WGS84/UTM coordinate reference system UTMCrs, and methods for the transformation of coordinates.

Namespaces

- namespace GeocoordinateTransformer

7.2 CoordinateTransformer.cs

Go to the documentation of this file.

```
00001 using ProjNet;
00002 using ProjNet.CoordinateSystems;
00003 using ProjNet.CoordinateSystems.Transformations;
00004 using System;
00005 using UnityEngine;
00006
00007 namespace GeocoordinateTransformer
00008 {
00009
00013     public class CoordinateTransformer : MonoBehaviour
00014     {
00018         public UTMCrs UTMCrs = new(utmZone: 32, hemispheres: Hemispheres.Northern);
00019
00024         [Tooltip("WGS84/UTM coordinates, e. g. X: 566600 (E), Y: 5933000 (N), Z: 0 (Altitude)")]
00025         public UTMCoordinates UTMCoordinates = new(east: 566600, north: 5933000, altitude: 0);
00026
00027
00028         public override bool Equals(object other)
00029         {
00030             if (other == null || GetType() != other.GetType()) { return false; }
00031
00032             CoordinateTransformer otherSceneOriginUtmCoordinates = other as CoordinateTransformer;
00033
00034             return (
00035                 this.UTMCrs == otherSceneOriginUtmCoordinates.UTMCrs &&
00036                 this.UTMCoordinates == otherSceneOriginUtmCoordinates.UTMCoordinates
00037                 );
00038         }
00039
00040         public override int GetHashCode()
00041         {
```

```
00042             return base.GetHashCode();
00043         }
00044
00045         public override string ToString()
00046         {
00047             Debug.Log("Here");
00048             return String.Format("SceneOriginUtmCoordinates (with UtmCRS: {0}, UTMCoordinates: {1})",
      UTMCrs, UTMCoordinates);
00049         }
00050
00051
00052
00056         private CoordinateSystemServices _CoordinateSystemServices;
00057
00061         private CoordinateSystemServices CoordinateSystemServices
00062         {
00063             get
00064             {
00065                 if (_CoordinateSystemServices == null)
00066                 {
00067                     _CoordinateSystemServices = new CoordinateSystemServices();
00068                 }
00069                 return _CoordinateSystemServices;
00070             }
00071         }
00072
00073
00077         private ICoordinateTransformation _UTMToGeographicCoordinates;
00078
00083         private ICoordinateTransformation UTMToGeographicCoordinates
00084         {
00085             get
00086             {
00087                 if (_UTMToGeographicCoordinates == null)
00088                 {
00089                     _UTMToGeographicCoordinates =
      CoordinateSystemServices.CreateTransformation(ProjectedCoordinateSystem.WGS84_UTM(UTMCrs.UTMZone,
      UTMCrs.IsNorthernHemisphere()), GeographicCoordinateSystem.WGS84);
00090                 }
00091                 return _UTMToGeographicCoordinates;
00092             }
00093         }
00094
00098         private ICoordinateTransformation _GeographicToUTMCoordinates;
00099
00104         private ICoordinateTransformation GeographicToUTMCoordinates
00105         {
00106             get
00107             {
00108                 if (_GeographicToUTMCoordinates == null)
00109                 {
00110                     _GeographicToUTMCoordinates =
      CoordinateSystemServices.CreateTransformation(GeographicCoordinateSystem.WGS84,
      ProjectedCoordinateSystem.WGS84_UTM(UTMCrs.UTMZone, UTMCrs.IsNorthernHemisphere()));
00111                 }
00112                 return _GeographicToUTMCoordinates;
00113             }
00114         }
00115
00121         public GeographicCoordinates GetGeographicCoordinates(UTMCoordinates utmCoordinates)
00122         {
00123             double[] coords = UTMToGeographicCoordinates.MathTransform.Transform(new double[] {
      utmCoordinates.east, utmCoordinates.north });
00124             return new GeographicCoordinates(latitude: coords[1], longitude: coords[0], altitude:
      utmCoordinates.altitude);
00125         }
00126
00127
00133         public GeographicCoordinates GetGeographicCoordinates(Vector3 unityCoordinates)
00134         {
00135             UTMCoordinates utmCoordinates = GetUTMCoordinates(unityCoordinates);
00136             return GetGeographicCoordinates(utmCoordinates);
00137         }
00138
00139
00145         public UTMCoordinates GetUTMCoordinates(GeographicCoordinates geographicCoordinates)
00146         {
00147             double[] coords = GeographicToUTMCoordinates.MathTransform.Transform(new double[] {
      geographicCoordinates.longitude, geographicCoordinates.latitude });
00148             return new UTMCoordinates(east: coords[0], north: coords[1], altitude:
      geographicCoordinates.altitude);
00149         }
00150
00151
00157         public Vector3 GetUnityCoordinates(GeographicCoordinates geographicCoordinates)
00158         {
00159             UTMCoordinates utmCoordinates = GetUTMCoordinates(geographicCoordinates);
```

```
00160             //Debug.LogFormat("utmCoordinates: {0}", utmCoordinates.ToString());
00161             return GetUnityCoordinates(utmCoordinates);
00162         }
00163 
00164 
00171         public Vector3 GetUnityCoordinates(UTMCoordinates utmCoordinates)
00172         {
00174 
00175             float eastComponent = (float)(utmCoordinates.east - UTMCoordinates.east);
00176             float northComponent = (float)(utmCoordinates.north - UTMCoordinates.north);
00177             float altitudeComponent = (float)(utmCoordinates.altitude - UTMCoordinates.altitude);
00178 
00179             if (
00180             !(eastComponent < 100 || eastComponent * -1 < 100) ||
00181             !(northComponent < 100 || northComponent * -1 < 100) ||
00182             !(altitudeComponent < 100 || altitudeComponent * -1 < 100)
00183             )
00184             {
00185                 throw new ArgumentException("Objects with a distance of more than 100 km from the
      Unity orgigin cannot be processed. Consider to change the SceneOriginUTMCoordinates value.");
00186             }
00187 
00188             Vector3 positionRelativeToAnchorPoint = new(eastComponent, northComponent,
      altitudeComponent);
00189 
00190             return SwitchLeftHandedRightHandedCoordinates(positionRelativeToAnchorPoint);
00191         }
00192 
00193 
00194 
00195 
00201         public UTMCoordinates GetUTMCoordinates(Vector3 unityCoordinates)
00202         {
00203             Vector3 pointsSwitchedUnityCoordinate =
      SwitchLeftHandedRightHandedCoordinates(unityCoordinates);
00204 
00205             double eastComponent = pointsSwitchedUnityCoordinate.x + UTMCoordinates.east;
00206             double northComponent = pointsSwitchedUnityCoordinate.y + UTMCoordinates.north;
00207             double altitude = pointsSwitchedUnityCoordinate.z + UTMCoordinates.altitude;
00208 
00209             return new UTMCoordinates(east: eastComponent, north: northComponent, altitude: altitude);
00210         }
00211 
00212 
00218         private Vector3 SwitchLeftHandedRightHandedCoordinates(Vector3 coordinateValues)
00219         {
00220             return new Vector3(coordinateValues.x, coordinateValues.z, coordinateValues.y);
00221         }
00222 
00223 
00227         public static CoordinateTransformer Instance;
00228 
00229         private void Awake()
00230         {
00231             // If Instance is not null (any time after the first time)
00232             // AND
00233             // If Instance is not 'this' (after the first time)
00234             if (Instance != null && Instance != this)
00235             {
00236                 // ...then destroy the game object this script component is attached to.
00237                 Destroy(gameObject);
00238                 Debug.LogWarningFormat("Only one CoordinateTransformer can exist.");
00239             }
00240             else
00241             {
00242                 // Tell Unity not to destory the GameObject this
00243                 //  is attached to between scenes.
00244                 DontDestroyOnLoad(gameObject);
00245                 // Save an internal reference to the first instance of this class
00246                 Instance = this;
00247             }
00248         }
00249     }
00250 
00251 }
```

7.3 CoordinateTransformerTest.cs File Reference

Classes

- class GeocoordinateTransformer.CoordinateTransformerTest

Test class to demonstrate the functionality of CoordinateTransformer. Contains necessary test variables and methods that can be changed and called from Unity's Inspector via the context menu.

Namespaces

- namespace GeocoordinateTransformer

7.4 CoordinateTransformerTest.cs

Go to the documentation of this file.

```
00001 using System;
00002 using UnityEngine;
00003
00004 namespace GeocoordinateTransformer
00005 {
00009     public class CoordinateTransformerTest : MonoBehaviour
00010     {
00014         [field: SerializeField]
00015         private CoordinateTransformer _coordinateTransformer;
00016
00021         public CoordinateTransformer CoordinateTransformer
00022         {
00023             get
00024             {
00025                 if (!_coordinateTransformer)
00026                 {
00027                     _coordinateTransformer = GameObject.FindAnyObjectByType<CoordinateTransformer>();
00028                     if (!_coordinateTransformer)
00029                     {
00030                         throw new ApplicationException("CoordinateTransformer could not be found in
      scene. Please add one CoordinateTransformer to each szene and provide UTM coordinates for the
      representing Unity's origin.");
00031                     }
00032                 }
00033
00034                 return _coordinateTransformer;
00035             }
00036             set => _coordinateTransformer = value;
00037         }
00038
00039         // Awake is called by Unity
00040         private void Awake()
00041         {
00042             CoordinateTransformer = GameObject.FindAnyObjectByType<CoordinateTransformer>();
00043             if (!CoordinateTransformer)
00044             {
00045                 Debug.LogError("CoordinateTransformer could not be found in scene. Please add one
      CoordinateTransformer to each szene and provide UTM coordinates for the representing Unity's
      origin.");
00046             }
00047         }
00048
00052         [SerializeField]
00053         private GeographicCoordinates geographicTestCoordinates = new(latitude: 53.5417104602435,
      longitude: 10.0051097859429, altitude: 4.25);
00054
00058         [SerializeField]
00059         private UTMCoordinates utmTestCoordinates = new(east: 566605, north: 5933004, altitude: 3);
00060
00061
00065         [ContextMenu("PlaceAccordingToGeographicCoordinates")]
00066         private void PlaceAccordingToGeographicCoordinates()
00067         {
00068             Debug.LogFormat("Given Geographic coordinates -- latitude: {0}, longitude: {1} and given
      altitude: {2}", geographicTestCoordinates.latitude, geographicTestCoordinates.longitude,
      geographicTestCoordinates.altitude);
00069
00070             Vector3 unityCoordinates =
      CoordinateTransformer.GetUnityCoordinates(geographicTestCoordinates);
00071             this.gameObject.transform.position = unityCoordinates;
00072             Debug.LogFormat("Derived Unity coordinates -- x: {0}, y: {1}, z:{2}\nObject placed
      accordingly.", unityCoordinates.x, unityCoordinates.y, unityCoordinates.z);
00073         }
00074
00075
00079         [ContextMenu("PlaceAccordingToUTMCoordinates")]
00080         private void PlaceAccordingToUTMCoordinates()
```

```
00081         {
00082             Debug.LogFormat("Given UTM coordinates -- E: {0}, N: {1} and given altitude: {2}",
      utmTestCoordinates.east, utmTestCoordinates.north, utmTestCoordinates.altitude);
00083
00084             Vector3 unityCoordinates = CoordinateTransformer.GetUnityCoordinates(utmTestCoordinates);
00085             this.gameObject.transform.position = unityCoordinates;
00086             Debug.LogFormat("Derived Unity coordinates -- x: {0}, y: {1}, z:{2}\nObject placed
      accordingly.", unityCoordinates.x, unityCoordinates.y, unityCoordinates.z);
00087         }
00088
00089
00093         [ContextMenu("GetGeographicCoordinatesOfCurrentPlacement")]
00094         private void GetGeographicCoordinatesOfCurrentPlacement()
00095         {
00096             Vector3 unityCoordinates = this.gameObject.transform.position;
00097             Debug.LogFormat("Given Unity coordinates of placed object -- x: {0}, y: {1}, z:{2}",
      unityCoordinates.x, unityCoordinates.y, unityCoordinates.z);
00098
00099             GeographicCoordinates geographicCoordinates =
      CoordinateTransformer.GetGeographicCoordinates(unityCoordinates);
00100             Debug.LogFormat("Derived GeographicCoordinates coordinates -- latitude: {0}, longitude:
      {1} and constant altitude: {2}", geographicCoordinates.latitude, geographicCoordinates.longitude,
      geographicCoordinates.altitude);
00101         }
00102
00103
00107         [ContextMenu("GetUTMCoordinatesOfCurrentPlacement")]
00108         private void GetUTMCoordinatesOfCurrentPlacement()
00109         {
00110             Vector3 unityCoordinates = this.gameObject.transform.position;
00111             Debug.LogFormat("Given Unity coordinates of placed object -- x: {0}, y: {1}, z:{2}",
      unityCoordinates.x, unityCoordinates.y, unityCoordinates.z);
00112
00113             UTMCoordinates utmCoordinates = CoordinateTransformer.GetUTMCoordinates(unityCoordinates);
00114             Debug.LogFormat("Derived UTM coordinates -- E: {0}, N: {1} and constant altitude: {2}",
      utmCoordinates.east, utmCoordinates.north, utmCoordinates.altitude);
00115         }
00116     }
00117
00118 }
```

7.5 GeographicCoordinates.cs File Reference

Classes

- class GeocoordinateTransformer.GeographicCoordinates

  Contains a latitude and longitude coordinate tuple and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Namespaces

- namespace GeocoordinateTransformer

7.6 GeographicCoordinates.cs

Go to the documentation of this file.

```
00001 using System;
00002 using UnityEditor.Experimental.GraphView;
00003
00004 namespace GeocoordinateTransformer
00005 {
00009     [System.Serializable]
00010     public class GeographicCoordinates
00011     {
00015         public double latitude;
00016
00020         public double longitude;
00021
00026         public double altitude;
```

```
00027
00028         public GeographicCoordinates(double latitude, double longitude, double altitude)
00029         {
00030             this.latitude = latitude;
00031             this.longitude = longitude;
00032             this.altitude = altitude;
00033         }
00034
00035
00036         public override bool Equals(object other)
00037         {
00038             if (other == null || GetType() != other.GetType()) { return false; }
00039
00040             GeographicCoordinates othergeographicCoordinates = other as GeographicCoordinates;
00041
00042             return (
00043                 this.latitude == othergeographicCoordinates.latitude &&
00044                 this.longitude == othergeographicCoordinates.longitude &&
00045                 this.altitude == othergeographicCoordinates.altitude
00046                 );
00047         }
00048
00049         public override int GetHashCode()
00050         {
00051             return base.GetHashCode();
00052         }
00053
00054         public override string ToString()
00055         {
00056             return String.Format("GeographicCoordinates (with latitude: {0}, longitude: {1}, altitude:
      {2}", latitude, longitude, altitude);
00057         }
00058     }
00059 }
```

7.7 UTMCoordinates.cs File Reference

Classes

- class GeocoordinateTransformer.UTMCoordinates

  Contains an easting and northing planar coordinate tuple in a WGS84/UTM coordinate reference system and an altitude value in a user chosen height reference system (e.g. DHHN2016 in Germany).

Namespaces

- namespace GeocoordinateTransformer

7.8 UTMCoordinates.cs

Go to the documentation of this file.

```
00001 using System;
00002
00003 namespace GeocoordinateTransformer
00004 {
00008     [System.Serializable]
00009     public class UTMCoordinates
00010     {
00014         public double east;
00015
00019         public double north;
00020
00025         public double altitude;
00026
00027
00028         public UTMCoordinates(double east, double north, double altitude)
00029         {
00030             this.east = east;
00031             this.north = north;
00032             this.altitude = altitude;
```

```
00033         }
00034
00035
00036         public override bool Equals(object other)
00037         {
00038             if (other == null || GetType() != other.GetType()) { return false; }
00039
00040             UTMCoordinates otherUTMCoordinates = other as UTMCoordinates;
00041
00042             return (
00043                 this.east == otherUTMCoordinates.east &&
00044                 this.north == otherUTMCoordinates.north &&
00045                 this.altitude == otherUTMCoordinates.altitude
00046                 );
00047         }
00048
00049         public override int GetHashCode()
00050         {
00051             return base.GetHashCode();
00052         }
00053
00054         public override string ToString()
00055         {
00056             return String.Format("UTMCoordinates (with east: {0}, north: {1}, altitude: {2})", east,
      north, altitude);
00057         }
00058     }
00059 }
```

7.9 UTMCrs.cs File Reference

Classes

- class GeocoordinateTransformer.UTMCrs

  Contains all variables to specify a WGS84/UTM coordinate refernece system.

Namespaces

- namespace GeocoordinateTransformer

Enumerations

- enum GeocoordinateTransformer.Hemispheres { GeocoordinateTransformer.Northern , GeocoordinateTransformer.Southern }

7.10 UTMCrs.cs

Go to the documentation of this file.

```
00001 using System;
00002 using UnityEngine;
00003
00004 namespace GeocoordinateTransformer
00005 {
00009     [System.Serializable]
00010     public class UTMCrs
00011     {
00016         [SerializeField, Range(1, 60)]
00017         public int UTMZone = 32;
00018
00023         [SerializeField]
00024         public Hemispheres Hemisphere = Hemispheres.Northern;
00025
00026         public UTMCrs(int utmZone, Hemispheres hemispheres)
00027         {
```

```
00028             if (utmZone < 0 || utmZone > 60) { throw new ArgumentException(String.Format("Invalid UTM
      zone of {0}", utmZone)); }
00029
00030             this.UTMZone = utmZone;
00031             this.Hemisphere = hemispheres;
00032         }
00033
00038         public bool IsNorthernHemisphere()
00039         {
00040             return Hemisphere switch
00041             {
00042                 Hemispheres.Northern => true,
00043                 Hemispheres.Southern => false,
00044                 _ => true,
00045             };
00046         }
00047
00048         public override bool Equals(object other)
00049         {
00050             if (other == null || GetType() != other.GetType()) { return false; }
00051
00052             UTMCrs otherUtmCRS = other as UTMCrs;
00053
00054             return (
00055                 this.UTMZone == otherUtmCRS.UTMZone &&
00056                 this.Hemisphere == otherUtmCRS.Hemisphere
00057                 );
00058         }
00059
00060         public override int GetHashCode()
00061         {
00062             return base.GetHashCode();
00063         }
00064
00065         public override string ToString()
00066         {
00067             return String.Format("UtmCRS (with UTMZone: {0}, Hemisphere: {1})", UTMZone, Hemisphere);
00068         }
00069     }
00070
00071     public enum Hemispheres
00072     {
00073         Northern,
00074         Southern
00075     }
00076 }
```

Index