uNature

2.1

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Packages

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:

Template Parameters

T	Type 1

Template Parameters

T	Type 1
T1	Type 2

Template Parameters

T	Type 1
T1	Type 2
T2	Type 3

Template Parameters

T	Type 1
T1	Type 2
T2	Type 3
T3	Type 4

Template Parameters

T	The network connection type which the networking library uses.

Template Parameters

T1	the targeted networking connection
T2	the type of the data

Chapter 4

Namespace Documentation

4.1 uNature Namespace Reference

Namespaces

4.2 uNature.Core Namespace Reference

Namespaces

Classes

- interface IUTCPhysicsIgnored

  Ignore all physics on this script.
- class **UNMath**

  An custom math class.
- class **UNPhysics**

  This class handles all custom physics.
- struct UNPhysicsHit_Grass

  A class that holds the data for the hit data
- class UNPhysicsHitsArray

  An custom array that holds all ray results in an array
- struct UNPhysicsObject

  This is a base class for a UCPhysicsObject. Every class that inherites this class will be counted in the physics system.
- struct UNPhysicsTemplate
- struct Vector2i

4.3 uNature.Core.ClassExtensions Namespace Reference

Classes

- class **ClassExtensions**

  Some extensions that helps achieving things that aren't built in with unity.

4.4 uNature.Core.Collections Namespace Reference

Classes

- class UNDimensionalList

    A 2 dimensional list which is used by certain mechanics in uNature.

- class UNList

    A custom list which is used on some important interfaces in UN.

4.5 uNature.Core.Editor Namespace Reference

Namespaces

4.6 uNature.Core.Editor.Helpers Namespace Reference

Classes

- class UNEditorHelpers

4.7 uNature.Core.Extensions Namespace Reference

Classes

- class MethodHelperAttribute
- class UN_ForgeNetworking
- class UN_GAIA
- class UN_GENA
- class UN_MapMagic
- class UN_PhotonBolt
- class UN_PhotonCloud
- class UN_TerrainComposer
- class UN_UFPS
- class UN_UNet
- class UN_WorldStreamer
- class UNExtension

    A uConstruct extension that will allow other 3d party systems to work with uConstruct.

- class UNExtensionsEditor

4.8 uNature.Core.FoliageClasses Namespace Reference

Classes

- class BaseInteraction
- class FoliageChunk
- class FoliageCore_Chunk
- class FoliageCore_MainManager
- class FoliageCore_Sector
- class FoliageDB

  The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.
- class FoliageDynamicSurface
- struct FoliageLODLevel
- class FoliageManagerInstance
- class FoliageMesh
- class FoliageMeshInstance
- class FoliageMeshInstancesGroup
- class FoliageMeshManager
- class FoliagePrototype
- class FoliageReceiver
- class FoliageSector

  An sector class dedicated only to Foliage.
- class GPUMesh

  A class used to hold the gpu meshes
- class GPUMeshLOD

  GPU Mesh Lods.
- class PaintBrush
- class **RenderingPipielineUtility**

  The rendering pipe line utility for uNature. Internal Only.
- class RenderingQueue

  An rendering queue.
- class RenderingQueueInstance

  An rendering queue instance.
- struct RenderingQueueMeshInstanceSimulator

  An imposter struct to simulate an instance.
- class RenderingQueueReceiver

  An object that can handle an object queue.
- class TouchBending
- class TouchBendingEditor
- class UNFoliageEditor
- class UNFoliageManagerEditor
- class UNMeshData
- class WindSettings
- class WindZones

Enumerations

- enum **FoliageType** { **Prefab**, **Texture** }
- enum **ShaderType** { **NaN** = 0, **Basic** = 1, **Advanced** = 2, **Custom** = 3 }
- enum **FoliageResolutions** {
  _128 = 128, **_256** = 256, **_512** = 512, **_1024** = 1024,
  _2048 = 2048 }
- enum **FoliageGenerationRadius** { **_1x1** = 1, **_3x3** = 3, **_5x5** = 5 }
- enum **CurrentPaintMethod** { **Normal_Paint**, **Spline_Paint** }

Functions

- delegate void **OnFoliageEnableChanged** (FoliagePrototype changedPrototype, bool value)
- delegate void **OnFoliageManagerAssigned** (FoliageCore_MainManager instance)

4.9 uNature.Core.Math Namespace Reference

Classes

- class **UNMath**

4.10 uNature.Core.Networking Namespace Reference

Classes

- class BaseUNNetworkData
- class UNNetworkData

  A class which can be used for an abstract networking data.

Template Parameters

T	*The network connection type which the networking library uses.*

- class UNNetworkManager

  A template for networking, which can be used by networking extensions to easily get the networking actions done.

Template Parameters

T1	*the targeted networking connection*
T2	*the type of the data*

- class UNNetworkPlayerController

Enumerations

- enum **PacketType** { **HealthUpdate** }

4.11 uNature.Core.Pooling Namespace Reference

Classes

- class HarvestableTIPoolItem

  A Pool item for terrain where the tree instances should be harvestable. (Tree cutting for instance) Inherite from this class to create your own harvestable type.
- interface IHarvestableItem
- interface IPoolComponent

- class Pool

  A class that manages the Pooling of the system, Which allows huge runtime performance increase.
- class PoolItem

  An abstract class that handles the Pool items.
- class TerrainPoolItem

  A Pool item for terrain. (Tree instances)

Functions

- delegate void **OnHealthChanged** (int value)
- delegate void **OnItemStateChanged** (HarvestableTIPoolItem item)
- delegate void **OnItemDamaged** (HarvestableTIPoolItem item, int damage)
- delegate void **OnTreeInstanceStateChanged** (Terrain terrain, int instanceID)

4.12 uNature.Core.Sectors Namespace Reference

Classes

- class Chunk

  part of the sector which contains information.
- class ChunkObject
- class GrassLODLevel

  A class that holds a level which all assigned on different frames.
- class Sector

  A sector which is used to divide the UNTerrain objects in the world to increase performance (can handle more than 200k trees!!)
- class TIChunk
- struct TreeFetchingTask_MultiThreaded
- class UNTerrainSector

  An sector class dedicated only for terrains.

Functions

- delegate void **SectorRecalculated** (List< Chunk > newChunks, Vector2 newChunkSize)

4.13 uNature.Core.Seekers Namespace Reference

Classes

- class UNSeeker

  Seekers are basically GameObjects in the scene which should interact with the objects in the game.
- class UNSeekerEditor

4.14 uNature.Core.Settings Namespace Reference

Classes

- class UNSetting

  A class which should be used on custom classes that needs to be shown and serialized as a setting.
- class UNSettingAttribute

  The attribute of each setting which handles the drawing of the setting (generically).
- class UNSettingCategory

  The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.
- class UNSettings

  A class which handles certain settings of aspects in UN
- class UNSettingsEditor

Enumerations

- enum UNSettingCategories {
  Terrain, **General**, **Networking**, **Interaction**,
  Threading, **Grass** }

  The categories of the settings which will be used on the editor.

4.14.1 Enumeration Type Documentation

4.14.1.1 enum uNature.Core.Settings.UNSettingCategories [strong]

The categories of the settings which will be used on the editor.

4.15 uNature.Core.Targets Namespace Reference

Classes

- class UNTarget

  A target is what will be taken into account with the system. For example terrains.
- class UNTargetEditor

4.16 uNature.Core.Terrains Namespace Reference

Classes

- class UNTerrain

  A class that needs to be on each terrain that needs to be taken into account when managing the system.
- class UNTerrainData

  The terrain data class which is used by uNature. Can be accesed by "UNTerrain.terrainData".
- class UNTerrainEditor
- class UNTreePrototype

  A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.

Enumerations

- enum **TerrainChangedFlags** {
  NoChange = 0, **Heightmap** = 1, **TreeInstances** = 2, **DelayedHeightmapUpdate** = 4,
  FlushEverythingImmediately = 8, **RemoveDirtyDetailsImmediately** = 16, **TreePrototypesChanged** = 32,
  WillBeDestroyed = 256 }
- enum **TerrainTabs** { **Grids**, **Pool**, **Vegetation**, **Trees** }

4.17 uNature.Core.Threading Namespace Reference

Classes

- interface IThreadTask

  A thread task interface. Implement on any customely created thread task.
- class ThreadItem

  This class handles assigning parameters before multi-threaded actions that can be called from outside of unity's main thread. for example : position.
- class ThreadTask

  A thread task that takes no parameters.
- class UNThreadManager

  This class handles the multi-threading mechanics.

Enumerations

- enum **uNature_Thread_Workers** {
  One_Worker = 1, **Two_Workers** = 2, **Three_Workers** = 3, **Four_Workers** = 4,
  Five_Workers = 5 }

4.18 uNature.Core.Utility Namespace Reference

Classes

- class BasePrototypeItem

  An base prototype item which is used for the uNature ui utility.
- class FoliageGrassMap

  Channels:
- class FoliageHeightMap

  Channels: R: Heights Channel #1 G: Heights Channel #2
- class FoliageNormalMap

  Channels: R: Heights Channel #1 G: Heights Channel #2
- class FoliageWorldMaps
- class UN_TerrainTexturePrototype

  The prototype of the terrain texture. [Used for listing the paintable surfaces]
- class **UNBatchUtility**

  An utility class for batching items.
- class UNBrushUtility

  Using this class you can paint an brush on the scene.
- struct UNCombineInstance

- class UNDictionary
- class UNFastList

  A list that requires you to set
- class UNMap

  The abstract Map class.
- class **UNMapGenerators**
- class **UNStandaloneUtility**
- struct Vector3_XZ_FAST

4.19 uNature.Demo Namespace Reference

Classes

- class UN_FirstPersonController
- class UN_MouseLook

4.20 uNature.Wrappers Namespace Reference

Namespaces

4.21 uNature.Wrappers.Linq Namespace Reference

Classes

- class **Check**
- interface IOrderedEnumerable
- class **LinqWrapper**
- class OrderedEnumerable
- class OrderedSequence
- class QuickSort
- class SortContext
- class SortSequenceContext

Enumerations

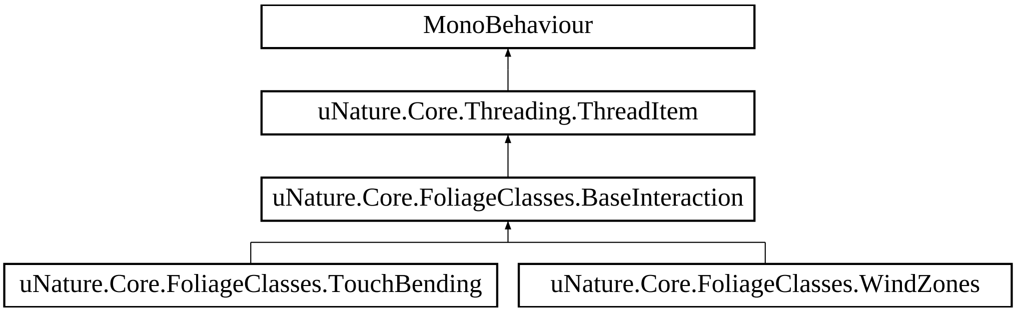
- enum **SortDirection** { **Ascending**, **Descending** }

Chapter 5

Class Documentation

5.1 uNature.Core.FoliageClasses.BaseInteraction Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.BaseInteraction:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.BaseInteraction
uNature.Core.FoliageClasses.TouchBending
uNature.Core.FoliageClasses.WindZones

Public Member Functions

- void UpdateInteraction (FoliageReceiver receiver)

  Update the interaction

Static Public Member Functions

- static List< BaseInteraction > **GetRelevantInteractions** (FoliageReceiver receiver)

Protected Member Functions

- override void **OnEnable** ()
- override void OnDisable ()

  Called when the object is disabled
- override void OnPositionChanged (Vector3 newPosition)

  Called when the item's position changed
- virtual void UpdateInteraction (FoliageReceiver receiver, Vector2 normalizedPosition)

  Please dont use map.SetPixels, it will be assigned later on automatically..
- Vector2 **GetNormalizedPosition** (FoliageReceiver receiver)
- Vector2 **GetNormalizedPosition** (Vector2 position, FoliageReceiver receiver)

Properties

- static List< BaseInteraction > **interactions** `[get]`
- virtual bool **includedInInteractionMap** `[get]`

Additional Inherited Members

5.1.1 Member Function Documentation

5.1.1.1 override void uNature.Core.FoliageClasses.BaseInteraction.OnDisable ( ) `[protected]`, `[virtual]`

Called when the object is disabled

Reimplemented from uNature.Core.Threading.ThreadItem.

Reimplemented in uNature.Core.FoliageClasses.TouchBending.

5.1.1.2 override void uNature.Core.FoliageClasses.BaseInteraction.OnPositionChanged (Vector3 *newPosition*) `[protected]`, `[virtual]`

Called when the item's position changed

Reimplemented from uNature.Core.Threading.ThreadItem.

Reimplemented in uNature.Core.FoliageClasses.TouchBending.

5.1.1.3 void uNature.Core.FoliageClasses.BaseInteraction.UpdateInteraction (**FoliageReceiver** *receiver*)

Update the interaction

Parameters

receiver	

5.1.1.4 virtual void uNature.Core.FoliageClasses.BaseInteraction.UpdateInteraction (**FoliageReceiver** *receiver,* Vector2 *normalizedPosition*) `[protected]`, `[virtual]`

Please dont use map.SetPixels, it will be assigned later on automatically..

Parameters

receiver	

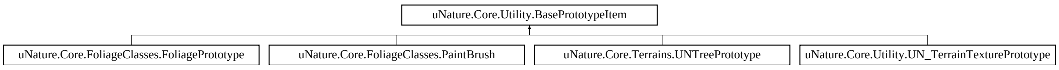
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/Interaction/BaseInteraction.cs

5.2 uNature.Core.Utility.BasePrototypeItem Class Reference

An base prototype item which is used for the uNature ui utility.

Inheritance diagram for uNature.Core.Utility.BasePrototypeItem:


uNature.Core.Utility.BasePrototypeItem
uNature.Core.FoliageClasses.FoliagePrototype
uNature.Core.FoliageClasses.PaintBrush
uNature.Core.Terrains.UNTreePrototype
uNature.Core.Utility.UN_TerrainTexturePrototype

Protected Member Functions

- virtual Texture2D **GetPreview** ()

Properties

- Texture2D **preview** `[get, set]`
- virtual bool **isEnabled** `[get]`
- virtual bool **chooseableOnDisabled** `[get]`

5.2.1 Detailed Description

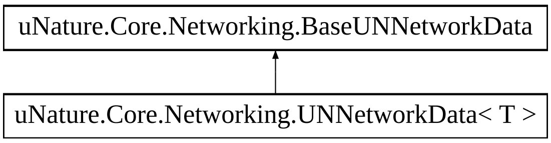
An base prototype item which is used for the uNature ui utility.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/BasePropertyItem.cs

5.3 uNature.Core.Networking.BaseUNNetworkData Class Reference

Inheritance diagram for uNature.Core.Networking.BaseUNNetworkData:


uNature.Core.Networking.BaseUNNetworkData
uNature.Core.Networking.UNNetworkData< T >

Public Member Functions

- virtual void UnPack ()

  Unpack the data

Public Attributes

- int **treeInstanceID**
- string **terrainID**
- PacketType **eventType** = PacketType.HealthUpdate

Protected Attributes

- int **_health**

Properties

- int **minHealth** `[get, set]`
- int **maxHealth** `[get, set]`
- int **health** `[get, set]`

5.3.1 Member Function Documentation

5.3.1.1 virtual void uNature.Core.Networking.BaseUNNetworkData.UnPack ( ) `[virtual]`

Unpack the data

Reimplemented in uNature.Core.Networking.UNNetworkData< T >.

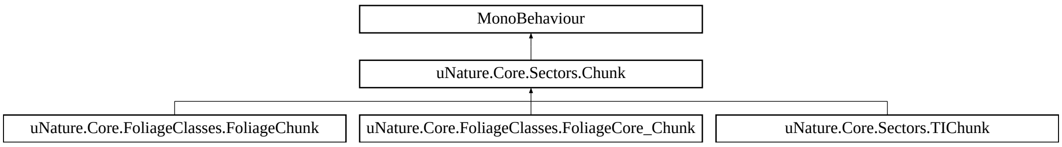
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/BaseUNNetworkData.cs

5.4 uNature.Core.Sectors.Chunk Class Reference

part of the sector which contains information.

Inheritance diagram for uNature.Core.Sectors.Chunk:


MonoBehaviour
uNature.Core.Sectors.Chunk
uNature.Core.FoliageClasses.FoliageChunk
uNature.Core.FoliageClasses.FoliageCore_Chunk
uNature.Core.Sectors.TIChunk

Public Member Functions

- virtual void Awake ()

  Called when the object is created/ initializes.

Parameters

terrain	*the terrain*

- bool Contains (Vector2 point, float offset)

  Is this point inside the chunk?
- bool Contains (Vector3 point, float offset)

  Is this point inside the chunk?
- virtual void OnCreated ()

  Called when the chunk is created.
- virtual void ResetChunk ()

  Reset the chunk's propoties.
- virtual void OnDrawGizmos ()

  Draw gizmos.

Static Public Member Functions

- static T CreateChunk< T > (Sector sector, Vector2 position, int x, int z, Vector2 scale, short chunkID)

  Create a new chunk

Public Attributes

- Vector2 **terrainRelativeSize**
- Vector2 **minPoint**
- Vector2 **maxPoint**
- short **chunkID**
- int **x**
- int **z**
- Transform **sectorOwner**

Protected Member Functions

- virtual void OnSizeChanged ()

  On size parameter changed.
- virtual void OnEnable ()

  Called on disable
- virtual void OnDisable ()

  Called on disable

Protected Attributes

- Vector3 **_position3D**

Properties

- Vector2 **position** `[get, set]`
- Vector3 **position3D** `[get]`
- Vector2 **size** `[get, set]`
- Vector2 **extents** `[get, set]`
- Vector2 **center** `[get]`
- virtual string **chunkType** `[get]`

5.4.1 Detailed Description

part of the sector which contains information.

5.4.2 Member Function Documentation

5.4.2.1 virtual void uNature.Core.Sectors.Chunk.Awake ( ) [virtual]

Called when the object is created/ initializes.

Parameters

terrain	the terrain

Reimplemented in uNature.Core.FoliageClasses.FoliageCore_Chunk, and uNature.Core.Sectors.TIChunk.

5.4.2.2 bool uNature.Core.Sectors.Chunk.Contains (Vector2 *point,* float *offset*)

Is this point inside the chunk?

Parameters

point	the point

Returns

5.4.2.3 bool uNature.Core.Sectors.Chunk.Contains (Vector3 *point,* float *offset*)

Is this point inside the chunk?

Parameters

point	the point

Returns

5.4.2.4 static T uNature.Core.Sectors.Chunk.CreateChunk< T > (Sector *sector,* Vector2 *position,* int *x,* int *z,* Vector2 *scale,* short *chunkID*) [static]

Create a new chunk

Template Parameters

T	

Parameters

sector	
position	
scale	
unTerrain	

Returns

Type Constraints

T* : *Chunk

5.4.2.5 virtual void uNature.Core.Sectors.Chunk.OnCreated ( ) [virtual]

Called when the chunk is created.

Reimplemented in uNature.Core.Sectors.TIChunk, uNature.Core.FoliageClasses.FoliageCore_Chunk, and uNature.Core.FoliageClasses.FoliageChunk.

5.4.2.6 virtual void uNature.Core.Sectors.Chunk.OnDisable ( ) [protected], [virtual]

Called on disable

Reimplemented in uNature.Core.FoliageClasses.FoliageChunk.

5.4.2.7 virtual void uNature.Core.Sectors.Chunk.OnDrawGizmos ( ) [virtual]

Draw gizmos.

Reimplemented in uNature.Core.Sectors.TIChunk, and uNature.Core.FoliageClasses.FoliageCore_Chunk.

5.4.2.8 virtual void uNature.Core.Sectors.Chunk.OnEnable ( ) [protected], [virtual]

Called on disable

5.4.2.9 virtual void uNature.Core.Sectors.Chunk.OnSizeChanged ( ) [protected], [virtual]

On size parameter changed.

Reimplemented in uNature.Core.FoliageClasses.FoliageCore_Chunk, uNature.Core.Sectors.TIChunk, and uNature.Core.FoliageClasses.FoliageChunk.

5.4.2.10 virtual void uNature.Core.Sectors.Chunk.ResetChunk ( ) [virtual]

Reset the chunk's propoties.

Reimplemented in uNature.Core.Sectors.TIChunk.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Chunk.cs

5.5 uNature.Core.Sectors.ChunkObject Class Reference

Public Member Functions

- **ChunkObject** (int _instanceID, TreeInstance treeInstance, Vector3 terrainSize, TerrainData tData, Vector3 terrainPosition)
- void **Remove** ()

Public Attributes

- TreeInstance **treeInstance**
- int **prototypeID**
- int **instanceID**
- Vector3 **worldPosition**
- Vector2 **depthPosition**
- System.DateTime **removedTime**
- float **originalHeight**
- HarvestableTIPoolItem **prefabHarvestableComponent**
- HarvestableTIPoolItem **harvestableComponent**

Properties

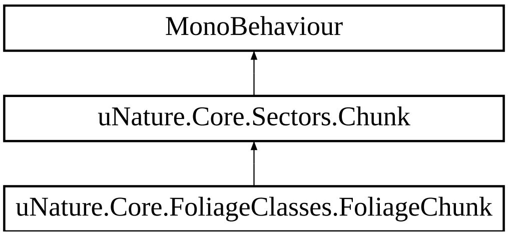
- bool **isRemoved** [get]

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Chunk.cs

5.6 uNature.Core.FoliageClasses.FoliageChunk Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageChunk:


MonoBehaviour
uNature.Core.Sectors.Chunk
uNature.Core.FoliageClasses.FoliageChunk

Public Member Functions

- override void OnCreated ()

  Create colliders

Protected Member Functions

- override void OnDisable ()

  Called on disable
- override void OnSizeChanged ()

  Called when the size of the chunk is changed.

Additional Inherited Members

5.6.1 Member Function Documentation

5.6.1.1 override void uNature.Core.FoliageClasses.FoliageChunk.OnCreated ( ) [virtual]

Create colliders

Reimplemented from uNature.Core.Sectors.Chunk.

5.6.1.2 override void uNature.Core.FoliageClasses.FoliageChunk.OnDisable ( ) [protected], [virtual]

Called on disable

Reimplemented from uNature.Core.Sectors.Chunk.

5.6.1.3 override void uNature.Core.FoliageClasses.FoliageChunk.OnSizeChanged ( ) [protected], [virtual]

Called when the size of the chunk is changed.

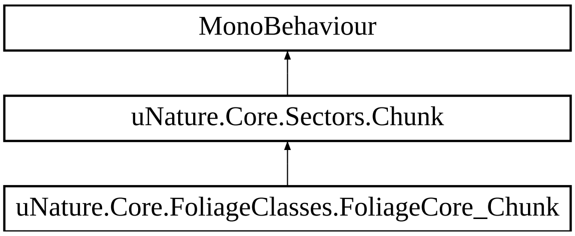
Reimplemented from uNature.Core.Sectors.Chunk.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageChunk.cs

5.7 uNature.Core.FoliageClasses.FoliageCore_Chunk Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageCore_Chunk:


MonoBehaviour
uNature.Core.Sectors.Chunk
uNature.Core.FoliageClasses.FoliageCore_Chunk

Public Member Functions

- override void OnDrawGizmos ()

  On Draw Gizmos
- override void OnCreated ()

  Called when the chunk is created.
- override void Awake ()

  Called when the object is created/ initializes.

Parameters

terrain	*the terrain*

- FoliageManagerInstance GetOrCreateFoliageManagerInstance ()

  Get the attached Foliage Manager Instance and if not availabe, create one.
- bool InBounds (Vector3 normalizedPosition, float distance)

  Check for in bounds

Protected Member Functions

- override void OnSizeChanged ()

  On size parameter changed.

Properties

- bool isFoliageInstanceAttached `[get]`

  Checks if an foliage instance exist on this chunk.

Additional Inherited Members

5.7.1 Member Function Documentation

5.7.1.1 override void uNature.Core.FoliageClasses.FoliageCore_Chunk.Awake ( ) `[virtual]`

Called when the object is created/ initializes.

Parameters

terrain	the terrain

Reimplemented from uNature.Core.Sectors.Chunk.

5.7.1.2 **FoliageManagerInstance** uNature.Core.FoliageClasses.FoliageCore_Chunk.GetOrCreateFoliageManagerInstance ( )

Get the attached Foliage Manager Instance and if not availabe, create one.

Returns

5.7.1.3 bool uNature.Core.FoliageClasses.FoliageCore_Chunk.InBounds (Vector3 *normalizedPosition,* float *distance*)

Check for in bounds

Parameters

normalizedPosition	
distance	

Returns

5.7.1.4 override void uNature.Core.FoliageClasses.FoliageCore_Chunk.OnCreated ( ) [virtual]

Called when the chunk is created.

Reimplemented from uNature.Core.Sectors.Chunk.

5.7.1.5 override void uNature.Core.FoliageClasses.FoliageCore_Chunk.OnDrawGizmos ( ) [virtual]

On Draw Gizmos

Reimplemented from uNature.Core.Sectors.Chunk.

5.7.1.6 override void uNature.Core.FoliageClasses.FoliageCore_Chunk.OnSizeChanged ( ) [protected], [virtual]

On size parameter changed.

Reimplemented from uNature.Core.Sectors.Chunk.

5.7.2 Property Documentation

5.7.2.1 bool uNature.Core.FoliageClasses.FoliageCore_Chunk.isFoliageInstanceAttached [get]

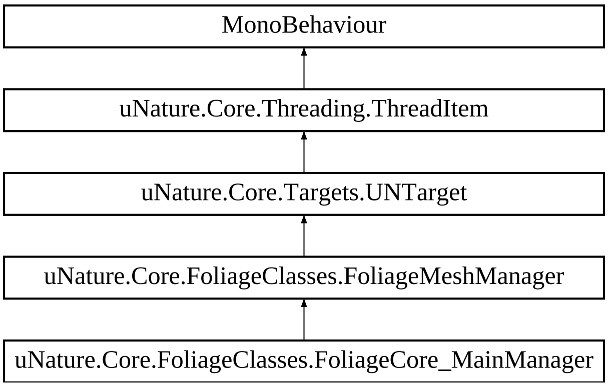
Checks if an foliage instance exist on this chunk.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageCore_Chunk.cs

5.8 uNature.Core.FoliageClasses.FoliageCore_MainManager Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageCore_MainManager:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Targets.UNTarget
uNature.Core.FoliageClasses.FoliageMeshManager
uNature.Core.FoliageClasses.FoliageCore_MainManager

Public Member Functions

- int GetChunkID (float x, float z)

    Get chunk from bounds.
- bool CheckChunkInBounds (int chunkID)

    Check if the chunk id is in range
- void InsertFoliageFromTerrain (Terrain terrain, bool saveImmediately)

    Copy the terrain's details and use it with the custom Foliage system.
- void UpdateHeightsOnTerrain (Terrain terrain)

    Update the heights on a terrain
- void UpdateHeights (int x, int z, int scaleX, int scaleZ)

    Update Heights On Cords
- byte[,] GetDetailLayer (int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex)

    Set detail layer in world cords
- void SetDetailLayer (int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex, byte[,] densities)

    Set detail layer in world cords

Static Public Member Functions

- static void SaveDelayedMaps ()

    Save maps that have been marked as delayed (waiting for update)
- static void RemoveGrassMap (FoliagePrototype prototype)

    Remove Grass Map Globally
- static void WarmUpGrassMaps ()

    Update the existing grass maps
- static void WarmUpGrassMaps (FoliageCore_Chunk[ ] specificChunks)

    Update the existing grass maps
- static void ResetGrassMap (List< FoliagePrototype > prototypes)

    Reset the existing grass maps
- static void InitializeAndCreateIfNotFound ()

    Create an instance if not created
- static void DestroyManager ()

    Destroy this manager instance and clean up the data.

Public Attributes

- const int **FOLIAGE_MAIN_AREA_RADIUS** = 10240
- const int **FOLIAGE_INSTANCE_AREA_SIZE** = (FOLIAGE_MAIN_AREA_RADIUS ∗ 2) / FOLIAGE_MAIN↩ _AREA_RESOLUTION
- bool **useQualitySettingsShadowDistance** = false
- float **foliageShadowDistance** = 100

Protected Member Functions

- override void Awake ()

  Initiate awake settings.
- override void **OnEnable** ()
- override void OnDisable ()

  Called when the object is disabled
- override void Update ()

  Update...

Properties

- static FoliageCore_MainManager **instance** `[get]`
- string **guid** `[get]`
- new bool **enabled** `[get, set]`
- float **density** `[get, set]`
- int **FoliageGenerationLayerMask** `[get, set]`
- FoliageCore_Sector **sector** `[get]`

Events

- static OnFoliageManagerAssigned **OnFoliageManagerAssignedEvent**

Additional Inherited Members

5.8.1 Member Function Documentation

5.8.1.1 override void uNature.Core.FoliageClasses.FoliageCore_MainManager.Awake ( ) `[protected], [virtual]`

Initiate awake settings.

Reimplemented from uNature.Core.Targets.UNTarget.

5.8.1.2 bool uNature.Core.FoliageClasses.FoliageCore_MainManager.CheckChunkInBounds (int *chunkID*)

Check if the chunk id is in range

Parameters

chunkID	

Returns

5.8.1.3 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.DestroyManager ( ) [static]

Destroy this manager instance and clean up the data.

5.8.1.4 int uNature.Core.FoliageClasses.FoliageCore_MainManager.GetChunkID (float *x*, float *z*)

Get chunk from bounds.

[REMOVE MAIN MANAGER POSITION FROM CORDS!!] for example: cordX = transform.position.x - FoliageCore_MainManager.instance.transform.position.x.

Parameters

x	
z	

Returns

5.8.1.5 byte [,] uNature.Core.FoliageClasses.FoliageCore_MainManager.GetDetailLayer (int *worldX*, int *worldZ*, int *sizeX*, int *sizeZ*, int *prototypeIndex*)

Set detail layer in world cords

Parameters

worldX	WORLD CORDS!!
worldZ	WORLD CORDS!!
sizeX	WORLD CORDS!!
sizeZ	WORLD CORDS!!
prototypeIndex	prototype.id

5.8.1.6 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.InitializeAndCreateIfNotFound ( ) [static]

Create an instance if not created

5.8.1.7 void uNature.Core.FoliageClasses.FoliageCore_MainManager.InsertFoliageFromTerrain (Terrain *terrain,* bool *saveImmediately*)

Copy the terrain's details and use it with the custom Foliage system.

Parameters

terrain	

5.8.1.8 override void uNature.Core.FoliageClasses.FoliageCore_MainManager.OnDisable ( ) [protected], [virtual]

Called when the object is disabled

Reimplemented from uNature.Core.Threading.ThreadItem.

5.8.1.9 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.RemoveGrassMap (FoliagePrototype *prototype*) [static]

Remove Grass Map Globally

Parameters

id	

5.8.1.10 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.ResetGrassMap (List< FoliagePrototype > *prototypes*) [static]

Reset the existing grass maps

Parameters

prototype	

5.8.1.11 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.SaveDelayedMaps ( ) [static]

Save maps that have been marked as delayed (waiting for update)

5.8.1.12 void uNature.Core.FoliageClasses.FoliageCore_MainManager.SetDetailLayer (int *worldX,* int *worldZ,* int *sizeX,* int *sizeZ,* int *prototypeIndex,* byte *densities[,]*)

Set detail layer in world cords

Parameters

worldX	WORLD CORDS!!
worldZ	WORLD CORDS!!
sizeX	WORLD CORDS!!
sizeZ	WORLD CORDS!!
prototypeIndex	prototype.id
densities	the density in bytes from 0 -> 15

5.8.1.13 override void uNature.Core.FoliageClasses.FoliageCore_MainManager.Update ( ) `[protected]`, `[virtual]`

Update...

Reimplemented from uNature.Core.Threading.ThreadItem.

5.8.1.14 void uNature.Core.FoliageClasses.FoliageCore_MainManager.UpdateHeights (int *x*, int *z*, int *scaleX*, int *scaleZ*)

Update Heights On Cords

Parameters

x	
z	
scaleX	
scaleZ	

5.8.1.15 void uNature.Core.FoliageClasses.FoliageCore_MainManager.UpdateHeightsOnTerrain (Terrain *terrain*)

Update the heights on a terrain

Parameters

terrain	

5.8.1.16 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.WarmUpGrassMaps ( ) `[static]`

Update the existing grass maps

Parameters

prototype	

5.8.1.17 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.WarmUpGrassMaps (FoliageCore_Chunk[ ] *specificChunks*) [static]

Update the existing grass maps

Parameters

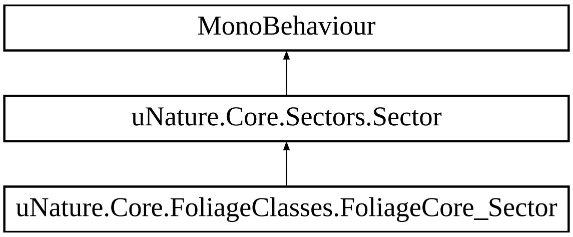
prototype	

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageCore_MainManager.cs

5.9 uNature.Core.FoliageClasses.FoliageCore_Sector Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageCore_Sector:


MonoBehaviour
uNature.Core.Sectors.Sector
uNature.Core.FoliageClasses.FoliageCore_Sector

Public Attributes

- List< FoliageCore_Chunk > **foliageChunks** = new List<FoliageCore_Chunk>()

Protected Member Functions

- override void OnChunkCreated (Chunk chunk)

  Called when a chunk is created to allow custom logic on the inherited sectors.
- override void OnStartCreatingChunks ()

  Called right before starting to create the chunks.
- override void OnResolutionChanged ()

  Called when the resolution has been updated.

Additional Inherited Members

5.9.1 Member Function Documentation

5.9.1.1 override void uNature.Core.FoliageClasses.FoliageCore_Sector.OnChunkCreated (Chunk *chunk*) [protected], [virtual]

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

chunk	

Reimplemented from uNature.Core.Sectors.Sector.

5.9.1.2 override void uNature.Core.FoliageClasses.FoliageCore_Sector.OnResolutionChanged ( ) [protected], [virtual]

Called when the resolution has been updated.

Reimplemented from uNature.Core.Sectors.Sector.

5.9.1.3 override void uNature.Core.FoliageClasses.FoliageCore_Sector.OnStartCreatingChunks ( ) [protected], [virtual]

Called right before starting to create the chunks.

Reimplemented from uNature.Core.Sectors.Sector.

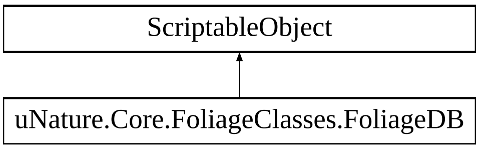
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageCore_Sector.cs

5.10 uNature.Core.FoliageClasses.FoliageDB Class Reference

The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.

Inheritance diagram for uNature.Core.FoliageClasses.FoliageDB:



Public Member Functions

- FoliagePrototype AddPrototype (Texture2D texture, GameObject prefab, float minWidth, float minHeight, float maxWidth, float maxHeight, float spread, int layer, Color healthyColor, Color dryColor)

    Add a new Foliage prototype.
- FoliagePrototype AddPrototype (DetailPrototype detailPrototype)

    Add a new Foliage prototype.
- FoliagePrototype AddPrototype (Texture2D texture)

    Add a new Foliage prototype.
- FoliagePrototype AddPrototype (GameObject prefab)

    Add a new Foliage prototype.
- void RemovePrototype (FoliagePrototype prototype)

    Remove an existing Foliage prototype.
- void UpdateShaderWindSettings ()

    Update wind settings globally
- void UpdateShaderGeneralSettings ()

    This will update the general settigns of the shader such as density, min width, max width etc

Public Attributes

- WindSettings **globalWindSettings** = new WindSettings()

Properties

- static FoliageDB instance `[get]`

  Get the instance, if not found, it will automatically create one.
- static List< FoliagePrototype > **unSortedPrototypes** `[get]`
- static Dictionary< int, FoliagePrototype > **sortedPrototypes** `[get]`
- List< PaintBrush > **brushes** `[get, set]`

5.10.1 Detailed Description

The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.

5.10.2 Member Function Documentation

5.10.2.1 FoliagePrototype uNature.Core.FoliageClasses.FoliageDB.AddPrototype (Texture2D *texture,* GameObject *prefab,* float *minWidth,* float *minHeight,* float *maxWidth,* float *maxHeight,* float *spread,* int *layer,* Color *healthyColor,* Color *dryColor*)

Add a new Foliage prototype.

5.10.2.2 FoliagePrototype uNature.Core.FoliageClasses.FoliageDB.AddPrototype (DetailPrototype *detailPrototype*)

Add a new Foliage prototype.

5.10.2.3 FoliagePrototype uNature.Core.FoliageClasses.FoliageDB.AddPrototype (Texture2D *texture*)

Add a new Foliage prototype.

5.10.2.4 FoliagePrototype uNature.Core.FoliageClasses.FoliageDB.AddPrototype (GameObject *prefab*)

Add a new Foliage prototype.

5.10.2.5 void uNature.Core.FoliageClasses.FoliageDB.RemovePrototype (FoliagePrototype *prototype*)

Remove an existing Foliage prototype.

5.10.2.6 void uNature.Core.FoliageClasses.FoliageDB.UpdateShaderGeneralSettings ( )

This will update the general settigns of the shader such as density, min width, max width etc

5.10.2.7 void uNature.Core.FoliageClasses.FoliageDB.UpdateShaderWindSettings ( )

Update wind settings globally

5.10.3 Property Documentation

5.10.3.1 FoliageDB uNature.Core.FoliageClasses.FoliageDB.instance [static], [get]

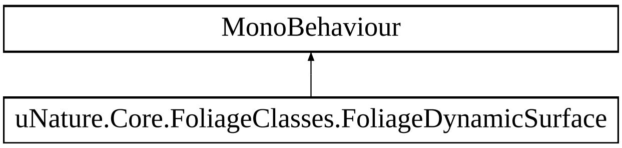
Get the instance, if not found, it will automatically create one.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageDB.cs

5.11 uNature.Core.FoliageClasses.FoliageDynamicSurface Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageDynamicSurface:



Public Attributes

- float **updateDistanceDifference** = 0.05f

Protected Member Functions

- virtual void **OnEnable** ()
- virtual void **OnDisable** ()
- virtual void **Update** ()
- virtual void **ApplyPositionChange** ()
- virtual void **ApplyScaleChange** ()

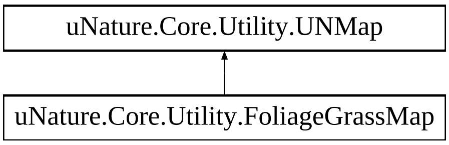
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageDynamicSurface.cs

5.12 uNature.Core.Utility.FoliageGrassMap Class Reference

Channels:

Inheritance diagram for uNature.Core.Utility.FoliageGrassMap:


uNature.Core.Utility.UNMap
uNature.Core.Utility.FoliageGrassMap

Public Member Functions

- **FoliageGrassMap** (Texture2D texture, FoliagePrototype prototype, FoliageManagerInstance mInstance)
- **FoliageGrassMap** (Texture2D texture, Color32[ ] pixels, FoliagePrototype prototype, FoliageManagerInstance mInstance)
- void **UpdateMap** ()
- void **ResetDensity** ()
- Texture2D SafeGetMap (FoliagePrototype prototype)

  This method will make sure that the map exists and if it doesnt it will create it.
- void Dispose ()

  Destroy this current grass map.
- byte GetDensity (int x, int z)

  Get density at normalized x & z
- void SetDensity (int x, int z, byte density)

  Set density at normalized x & z
- void SetDensityFast (int index, byte density)
- void MarkDensitiesDirty ()

  Mark the densities as dirty.
- void **Save** ()

Static Public Member Functions

- static void **SaveAllMaps** ()
- static void **ApplyAreaSizeChange** (FoliageManagerInstance mInstance)
- static void UpdateGrassMaps (FoliageManagerInstance mInstance)

  Update all of the availble grass maps (pixels)

Properties

- float **perlinScale** `[get, set]`
- int **prototypeID** `[get]`
- static bool **globalDirty** `[get]`

Additional Inherited Members

5.12.1 Detailed Description

Channels:

R: Free G: Free B: Density A: Perlin Noise

5.12.2 Member Function Documentation

5.12.2.1 void uNature.Core.Utility.FoliageGrassMap.Dispose ( )

Destroy this current grass map.

5.12.2.2 byte uNature.Core.Utility.FoliageGrassMap.GetDensity (int *x*, int *z*)

Get density at normalized x & z

Parameters

x	
z	

Returns

5.12.2.3 void uNature.Core.Utility.FoliageGrassMap.MarkDensitiesDirty ( )

Mark the densities as dirty.

5.12.2.4 Texture2D uNature.Core.Utility.FoliageGrassMap.SafeGetMap (FoliagePrototype *prototype*)

This method will make sure that the map exists and if it doesnt it will create it.

Used mainly for the rendering so if the user accidently/ purposely removed the grass map it will automatically generate a new one so it wont affect the system.

Parameters

manager	

Returns

5.12.2.5 void uNature.Core.Utility.FoliageGrassMap.SetDensity (int *x,* int *z,* byte *density*)

Set density at normalized x & z

Parameters

x	
z	

Returns

5.12.2.6 void uNature.Core.Utility.FoliageGrassMap.SetDensityFast (int *index,* byte *density*)

Parameters

x	
z	

Returns

5.12.2.7 static void uNature.Core.Utility.FoliageGrassMap.UpdateGrassMaps (FoliageManagerInstance *mInstance*) [static]

Update all of the availble grass maps (pixels)

Parameters

size	

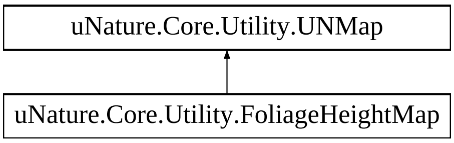
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.13 uNature.Core.Utility.FoliageHeightMap Class Reference

Channels: R: Heights Channel #1 G: Heights Channel #2

Inheritance diagram for uNature.Core.Utility.FoliageHeightMap:


uNature.Core.Utility.UNMap
uNature.Core.Utility.FoliageHeightMap

Public Member Functions

- **FoliageHeightMap** (Texture2D texture, FoliageManagerInstance mInstance)
- void **Save** ()

Additional Inherited Members

5.13.1 Detailed Description

Channels: R: Heights Channel #1 G: Heights Channel #2

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.14 uNature.Core.FoliageClasses.FoliageLODLevel Struct Reference

Public Member Functions

- **FoliageLODLevel** (float lodDistance, float lodValue)

Public Attributes

- const int **LOD_MAX_DISTANCE** = 500
- Vector2 **_vectorRepresentation**
- bool **_isDirty**

Properties

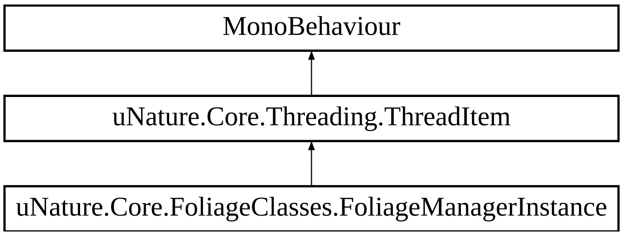
- Vector2 **vectorRepresentation** `[get]`
- float **lodDistance** `[get, set]`
- float **lodValue** `[get, set]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

5.15 uNature.Core.FoliageClasses.FoliageManagerInstance Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageManagerInstance:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.FoliageManagerInstance

Public Member Functions

- void ForceMapsRestore ()

  Restores the changes on the maps as long as saving changes on runtime isnt checked on the settings.
- void **UpdateMaterialBlock** (MaterialPropertyBlock mBlock)
- void **RemoveGrassMap** (FoliagePrototype prototype)
- int TransformCord (float x, float removeOffset)

  Transform 1 cord
- float TransformCordFloat (float x, float removeOffset)

  Transform 1 cord
- int TransformCordCustom (float x, float removeOffset, float multiplier)

  Transform 1 cord
- float TransformCordCustomFloat (float x, float removeOffset, float multiplier)

  Transform 1 cord
- int InverseCord (float x, float addOffset)

  Transform 1 cord
- float InverseCordFloat (float x, float addOffset)

  Transform 1 cord
- int InverseCordCustom (float x, float addOffset, float multiplier)

  Transform 1 cord
- float InverseCordCustomFloat (float x, float addOffset, float multiplier)

  Transform 1 cord

Static Public Member Functions

- static void **CleanUp** (FoliageManagerInstance mInstance)

Protected Member Functions

- override void **OnEnable** ()
- override void OnDisable ()

  Called when the object is disabled
- virtual void **UpdateResolutionChange** ()

Properties

- static List< FoliageManagerInstance > **instances** `[get]`
- string **guid** `[get]`
- Dictionary< int, GPUMesh > **meshInstances** `[get]`
- Vector3 **pos** `[get]`
- FoliageResolutions **foliageAreaResolution** `[get, set]`
- int **foliageAreaResolutionIntegral** `[get]`
- float **transformCordsMultiplier** `[get, set]`
- int **FoliageGenerationLayerMask** `[get, set]`
- Texture2D **colorMap** `[get, set]`
- FoliageWorldMaps **worldMaps** `[get, set]`
- Dictionary< int, FoliageGrassMap > **grassMaps** `[get]`
- Dictionary< int, FoliageGrassMap > **unSafeGrassMaps** `[get]`
- FoliageCore_Chunk **attachedTo** `[get]`

Additional Inherited Members

5.15.1 Member Function Documentation

5.15.1.1 void uNature.Core.FoliageClasses.FoliageManagerInstance.ForceMapsRestore ( )

Restores the changes on the maps as long as saving changes on runtime isnt checked on the settings.

5.15.1.2 int uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCord (float *x*, float *addOffset*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.3 int uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCordCustom (float *x*, float *addOffset*, float *multiplier*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.4 float uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCordCustomFloat (float *x*, float *addOffset*, float *multiplier*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.5 float uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCordFloat (float *x*, float *addOffset*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.6 override void uNature.Core.FoliageClasses.FoliageManagerInstance.OnDisable ( ) [protected], [virtual]

Called when the object is disabled

Reimplemented from uNature.Core.Threading.ThreadItem.

5.15.1.7 int uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCord (float *x*, float *removeOffset*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.8 int uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCordCustom (float *x*, float *removeOffset*, float *multiplier*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.9 float uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCordCustomFloat (float *x,* float *removeOffset,* float *multiplier*)

Transform 1 cord

Parameters

x	

Returns

5.15.1.10 float uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCordFloat (float *x,* float *removeOffset*)

Transform 1 cord

Parameters

x	

Returns

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageManagerInstance.cs

5.16 uNature.Core.FoliageClasses.FoliageMesh Class Reference

Public Member Functions

- **FoliageMesh** (GameObject go, int layer, string name, Vector3 offset)

Public Attributes

- const int **OPTIMIZATION_MESH_INSTANCES_DENSITIES_LIMITER** = 12
- Mesh **mesh**
- Material **mat**
- Vector3 **eulerAngles**
- Vector3 **scale** = Vector3.one
- Vector3 **offset**
- int **vertexCount**

Properties

- Vector3 **rendererScale** `[get]`
- Vector3 **worldScale** `[get]`
- UNMeshData **meshData** `[get]`
- ShaderType **shaderType** `[get]`
- int **MeshInstancesLimiter_Optimization_Clamp** `[get]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

5.17 uNature.Core.FoliageClasses.FoliageMeshInstance Class Reference

Public Member Functions

- Vector3 **GetPosition** (Vector3 pos)
- byte GetDensity (float positionX, float positionZ, byte maxDensity, Color32[ ] mapPixels, int mapWidth)

  get density of the mesh instance.

Static Public Member Functions

- static Mesh[ ][ ] CreateFoliageInstances (int prototypeIndex, List< byte > densities, out FoliageMesh←InstancesGroup meshGroup, FoliageResolutions resolution)

  Generates the mesh instances.
- static void MarkDensitiesDirty ()

  Mark all of the densities as dirty

Public Attributes

- FoliagePrototype **prototype**
- Vector3 **position**
- Mesh **mesh**
- Bounds **cullBounds**
- int **boundsSizeX**
- int **boundsSizeZ**
- Vector3 **boundsExtents**
- int **meshIndex**

Properties

- FoliageChunk **currentChunk** `[get, set]`

5.17.1 Member Function Documentation

5.17.1.1 static Mesh [ ][ ] uNature.Core.FoliageClasses.FoliageMeshInstance.CreateFoliageInstances (int *prototypeIndex*, List< byte > *densities*, out **FoliageMeshInstancesGroup** *meshGroup*, FoliageResolutions *resolution*) [static]

Generates the mesh instances.

Returns

5.17.1.2 byte uNature.Core.FoliageClasses.FoliageMeshInstance.GetDensity (float *positionX*, float *positionZ*, byte *maxDensity*, Color32[ ] *mapPixels*, int *mapWidth*)

get density of the mesh instance.

Parameters

pos	

Returns

5.17.1.3 static void uNature.Core.FoliageClasses.FoliageMeshInstance.MarkDensitiesDirty ( ) [static]

Mark all of the densities as dirty

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageMeshInstance.cs

5.18 uNature.Core.FoliageClasses.FoliageMeshInstancesGroup Class Reference

Public Member Functions

- void **Add** (FoliageMeshInstance instance)
- void **Finish** ()
- void **Destroy** ()

Public Attributes

- FoliageMeshInstance[ ] **meshInstances** = null

Properties

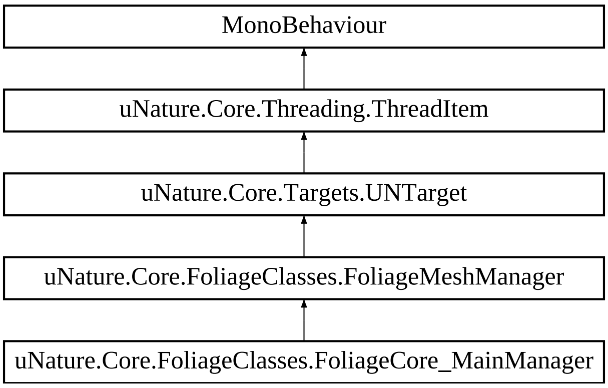
- int **Count** `[get]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageMeshManager.cs

5.19 uNature.Core.FoliageClasses.FoliageMeshManager Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageMeshManager:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Targets.UNTarget
uNature.Core.FoliageClasses.FoliageMeshManager
uNature.Core.FoliageClasses.FoliageCore_MainManager

Public Member Functions

- void UpdateMeshBounds (Vector3 centerPos)

  Update the mesh instances bounds
- void **OnGlobalPostRender** (Camera camera)
- void **DEBUG_DrawUI** ()

Static Public Member Functions

- static void GenerateFoliageMeshInstances ()

  Generate new mesh instances
- static void GenerateFoliageMeshInstances (FoliageResolutions resolution)

  Generate new mesh instances
- static void GenerateFoliageMeshInstances (int prototypeID)

  Generate new mesh instances
- static void GenerateFoliageMeshInstanceForIndex (int prototypeIndex, FoliageResolutions resolution)

  Create Foliage mesh instances for a certain index and foliage size.
- static void DestroyMeshInstance (int prototypeID)

  Destroy a mesh instance
- static void RegenerateQueueInstances ()

  Restart all of the queue instances.
- static Mesh CreateNewMesh ()

  Create a new mesh instace

Public Attributes

- bool **DEBUG_Window_Open** = false
- bool **DEBUG_Window_Minimized** = false

Static Public Attributes

- static List< Mesh > **globalMeshesThreshold** = new List<Mesh>()

Protected Member Functions

- override void **OnEnable** ()
- override void OnDisable ()

  Called when the object is disabled
- override void Update ()

  Update...

Static Protected Member Functions

- static void DestroyMeshInstances ()

  Destroy the current mesh instances.

Protected Attributes

- int **_lastRenderedVertices**
- int **_lastRenderedDrawCalls**
- int **_lastRenderedPrototypes**

Properties

- static int **PROPERTY_ID_WORLDPOSITION** `[get]`
- static int **PROPERTY_ID_GRASSMAP** `[get]`
- static int **PROPERTY_ID_FOLIAGE_INTERACTION_TOUCH_BENDING_OBJECTS** `[get]`
- static Dictionary< FoliageResolutions, Dictionary< int, GPUMesh > > **prototypeMeshInstances** `[get]`
- MaterialPropertyBlock **propertyBlock** `[get]`
- int **lastRenderedVertices** `[get]`
- int **lastRenderedDrawCalls** `[get]`
- int **lastRenderedPrototypes** `[get]`

5.19.1 Member Function Documentation

5.19.1.1 static Mesh uNature.Core.FoliageClasses.FoliageMeshManager.CreateNewMesh ( ) `[static]`

Create a new mesh instace

Returns

5.19.1.2 static void uNature.Core.FoliageClasses.FoliageMeshManager.DestroyMeshInstance (int *prototypeID*) `[static]`

Destroy a mesh instance

Parameters

prototypeID	

5.19.1.3 static void uNature.Core.FoliageClasses.FoliageMeshManager.DestroyMeshInstances ( ) [static], [protected]

Destroy the current mesh instances.

5.19.1.4 static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstanceForIndex (int *prototypeIndex*, FoliageResolutions *resolution*) [static]

Create Foliage mesh instances for a certain index and foliage size.

Parameters

meshInstances	
prototypeIndex	

5.19.1.5 static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstances ( ) [static]

Generate new mesh instances

Parameters

areaSize	

5.19.1.6 static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstances (FoliageResolutions *resolution*) [static]

Generate new mesh instances

Parameters

areaSize	

5.19.1.7 static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstances (int *prototypeID*) [static]

Generate new mesh instances

Parameters

areaSize	

5.19.1.8 override void uNature.Core.FoliageClasses.FoliageMeshManager.OnDisable ( ) [protected], [virtual]

Called when the object is disabled

Reimplemented from uNature.Core.Threading.ThreadItem.

5.19.1.9 static void uNature.Core.FoliageClasses.FoliageMeshManager.RegenerateQueueInstances ( ) [static]

Restart all of the queue instances.

5.19.1.10 override void uNature.Core.FoliageClasses.FoliageMeshManager.Update ( ) [protected], [virtual]

Update...

Reimplemented from uNature.Core.Threading.ThreadItem.

5.19.1.11 void uNature.Core.FoliageClasses.FoliageMeshManager.UpdateMeshBounds (Vector3 *centerPos*)

Update the mesh instances bounds

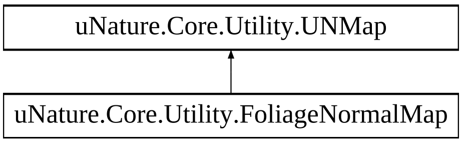
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageMeshManager.cs

5.20 uNature.Core.Utility.FoliageNormalMap Class Reference

Channels: R: Heights Channel #1 G: Heights Channel #2

Inheritance diagram for uNature.Core.Utility.FoliageNormalMap:


uNature.Core.Utility.UNMap
uNature.Core.Utility.FoliageNormalMap

Public Member Functions

- **FoliageNormalMap** (Texture2D texture, FoliageManagerInstance mInstance)
- void Save ()

  Save

Additional Inherited Members

5.20.1 Detailed Description

Channels: R: Heights Channel #1 G: Heights Channel #2

5.20.2 Member Function Documentation

5.20.2.1 void uNature.Core.Utility.FoliageNormalMap.Save ( )

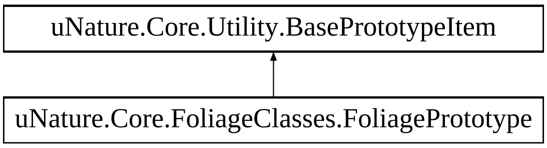
Save

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.21 uNature.Core.FoliageClasses.FoliagePrototype Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliagePrototype:


uNature.Core.Utility.BasePrototypeItem
uNature.Core.FoliageClasses.FoliagePrototype

Public Member Functions

- bool **EqualsToPrototype** (DetailPrototype detail)
- void ApplyWind ()

  Apply the wind parameters to this Foliage prototype.
- void ApplyColorMap (Texture2D map, Texture2D normalMap)

  Apply color map
- void ApplyGrassMap (Texture2D map)

  Apply color map
- void UpdateManagerInformation ()

  Update the global spread noise.

Static Public Member Functions

- static FoliagePrototype CreatePrototype (Texture2D texture, GameObject prefab, float minWidth, float min←
Height, float maxWidth, float maxHeight, float spread, int layer, int id, Color healthyColor, Color dryColor)

  Create a prototype.

Public Attributes

- const string **SHADER_BASIC_NAME** = "uNature/FoliageShader_Basic"
- const string **SHADER_ADVANCED_NAME** = "uNature/FoliageShader_Advanced"
- const float **SIZE_MIN_VALUE** = 0.1f
- const float **SIZE_MAX_VALUE** = 5.0f
- WindSettings **customWindSettings** = new WindSettings()
- Vector3 **instancedEuler**

Static Public Attributes

- static Color **DEFAULT_HEALTHY_COLOR** = new Color(33f / 255, 129f / 255, 25f / 255, 1)
- static Color **DEFAULT_DRY_COLOR** = new Color(205f / 255, 188f / 255, 26f / 255, 1)

Protected Member Functions

- override Texture2D GetPreview ()

  Get Preview

Properties

- static GameObject **FoliageTexGameObject** `[get]`
- FoliageType **FoliageType** `[get, set]`
- GameObject **FoliageMesh** `[get, set]`
- Texture2D **FoliageTexture** `[get, set]`
- int **id** `[get]`
- int **maxFoliageCapability** `[get]`
- float **spread** `[get, set]`
- float **minimumWidth** `[get, set]`
- float **maximumWidth** `[get, set]`
- float **minimumHeight** `[get, set]`
- float **maximumHeight** `[get, set]`
- bool **receiveShadows** `[get, set]`
- Color **dryColor** `[get, set]`
- Color **healthyColor** `[get, set]`
- bool **castShadows** `[get, set]`
- int **fadeDistance** `[get, set]`
- int **maxGeneratedDensity** `[get, set]`
- bool **useColorMap** `[get, set]`
- bool **rotateNormals** `[get, set]`
- string **name** `[get]`
- bool **enabled** `[get, set]`
- int **meshLodsCount** `[get]`
- int **renderingLayer** `[get, set]`
- bool **touchBendingEnabled** `[get, set]`
- float **touchBendingStrength** `[get, set]`
- float **cutOff** `[get, set]`
- override bool **isEnabled** `[get]`
- override bool **chooseableOnDisabled** `[get]`
- bool **useCustomWind** `[get, set]`
- bool **useLODs** `[get, set]`
- FoliageLODLevel[ ] **lods** `[get, set]`
- FoliageMesh **FoliageInstancedMeshData** `[get]`
- List< byte > **densitiesLODs** `[get]`

Events

- static OnFoliageEnableChanged **OnFoliageEnabledStateChangedEvent**

5.21.1 Member Function Documentation

5.21.1.1 void uNature.Core.FoliageClasses.FoliagePrototype.ApplyColorMap (Texture2D *map,* Texture2D *normalMap*)

Apply color map

Res = area size.

5.21.1.2 void uNature.Core.FoliageClasses.FoliagePrototype.ApplyGrassMap (Texture2D *map*)

Apply color map

Res = area size.

5.21.1.3 void uNature.Core.FoliageClasses.FoliagePrototype.ApplyWind ( )

Apply the wind parameters to this Foliage prototype.

5.21.1.4 static FoliagePrototype uNature.Core.FoliageClasses.FoliagePrototype.CreatePrototype (Texture2D *texture,* GameObject *prefab,* float *minWidth,* float *minHeight,* float *maxWidth,* float *maxHeight,* float *spread,* int *layer,* int *id,* Color *healthyColor,* Color *dryColor*) [static]

Create a prototype.

Parameters

texture	
prefab	
minSize	
maxSize	
spread	
id	

Returns

5.21.1.5 override Texture2D uNature.Core.FoliageClasses.FoliagePrototype.GetPreview ( ) [protected], [virtual]

Get Preview

Returns

Reimplemented from uNature.Core.Utility.BasePrototypeItem.

5.21.1.6 void uNature.Core.FoliageClasses.FoliagePrototype.UpdateManagerInformation ( )

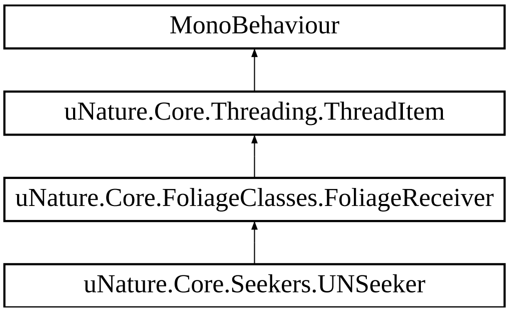
Update the global spread noise.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

5.22 uNature.Core.FoliageClasses.FoliageReceiver Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageReceiver:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.FoliageReceiver
uNature.Core.Seekers.UNSeeker

Public Attributes

- bool **isGrassReceiver** = true

Static Public Attributes

- static readonly List< FoliageReceiver > **FReceivers** = new List<FoliageReceiver>()

Protected Member Functions

- override void **OnEnable** ()
- override void OnDisable ()

  Called when the object is disabled
- override void Update ()

  Update...

Protected Attributes

- float **_grassCheckDistance** = 20f

Properties

- FoliageCore_Chunk[ ] **neighbors** [get]
- FoliageCore_Chunk **middleFoliageChunkFromNeighbors** [get]
- float **grassCheckDistance** [get, set]
- Camera **playerCamera** [get, set]
- RenderingQueueReceiver **queueInstance** [get, set]

Additional Inherited Members

5.22.1 Member Function Documentation

5.22.1.1 override void uNature.Core.FoliageClasses.FoliageReceiver.OnDisable ( ) [protected], [virtual]

Called when the object is disabled

Reimplemented from uNature.Core.Threading.ThreadItem.

5.22.1.2 override void uNature.Core.FoliageClasses.FoliageReceiver.Update ( ) [protected], [virtual]

Update...

Reimplemented from uNature.Core.Threading.ThreadItem.

Reimplemented in uNature.Core.Seekers.UNSeeker.

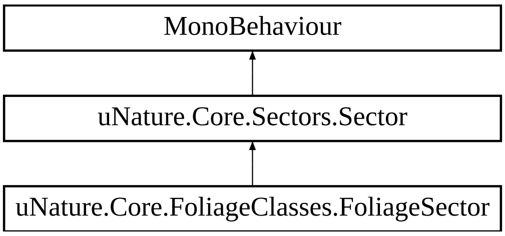
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foliage/GPU_Utilities/FoliageReceiver.cs

5.23 uNature.Core.FoliageClasses.FoliageSector Class Reference

An sector class dedicated only to Foliage.

Inheritance diagram for uNature.Core.FoliageClasses.FoliageSector:


MonoBehaviour
uNature.Core.Sectors.Sector
uNature.Core.FoliageClasses.FoliageSector

Public Attributes

- List< FoliageChunk > **FoliageChunks** = new List<FoliageChunk>()

Protected Member Functions

- override void OnChunkCreated (Chunk chunk)

  Called when a chunk is created to allow custom logic on the inherited sectors.
- override void OnStartCreatingChunks ()

  Called right before starting to create the chunks.

Additional Inherited Members

5.23.1 Detailed Description

An sector class dedicated only to Foliage.

5.23.2 Member Function Documentation

5.23.2.1 override void uNature.Core.FoliageClasses.FoliageSector.OnChunkCreated (**Chunk** *chunk*) `[protected]`, `[virtual]`

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

chunk	

Reimplemented from uNature.Core.Sectors.Sector.

5.23.2.2 override void uNature.Core.FoliageClasses.FoliageSector.OnStartCreatingChunks ( ) `[protected]`, `[virtual]`

Called right before starting to create the chunks.

Reimplemented from uNature.Core.Sectors.Sector.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageSector.cs

5.24 uNature.Core.Utility.FoliageWorldMaps Class Reference

Public Member Functions

- **FoliageWorldMaps** (Texture2D heightMap, Texture2D normalMap, FoliageManagerInstance mInstance)
- Vector2 NormalizeHeight (float worldHeight)

  Normalize a world height into a converted height.
- void UpdateHeightsAndNormals (float x, float z, int sizeX, int sizeZ, bool save)

Update height on a certain range.

- void UpdateHeightsAndNormals (bool save)

  Update the heights and normals all over the map.
- void UpdateHeightAndNormal (int index, float height)

  Update height
- void UpdateHeightAndNormal (int index, Vector3 normal)

  Update normal
- void UpdateHeightAndNormal (int index, float height, Vector3 normal)

  Update height and normal
- float GetHeight (int index)

  Get an height on the height map.
- void Save ()

  Save the world maps
- void SetPixels32Delayed ()

  Set pixels on the world maps, delayed.
- void RestoreChanges ()

  Reset the changes of runtime.
- override bool **Equals** (object obj)
- override int **GetHashCode** ()

Static Public Member Functions

- static Vector3 TransformNormals (Vector3 normal)

  Normalize the world normals into converted normals.
- static float ClampNegativeIntoPositive (float value)

  Convert point from negative to positive.
- static void ReGenerateGlobally ()

  Regenerate all of the world maps.
- static void SaveAllMaps ()

  Save all of the world maps.
- static void **ApplyAreaSizeChange** (FoliageManagerInstance mInstance)

Properties

- FoliageHeightMap **heightMap** `[get, set]`
- FoliageNormalMap **normalMap** `[get, set]`
- bool **dirty** `[get]`
- static bool **globalDirty** `[get]`

5.24.1 Member Function Documentation

5.24.1.1 static float uNature.Core.Utility.FoliageWorldMaps.ClampNegativeIntoPositive (float *value*) `[static]`

Convert point from negative to positive.

-1 = 0; 0 = 0.5; 1 = 1;

Parameters

value	

Returns

5.24.1.2 float uNature.Core.Utility.FoliageWorldMaps.GetHeight (int *index*)

Get an height on the height map.

Parameters

pixel	

Returns

5.24.1.3 Vector2 uNature.Core.Utility.FoliageWorldMaps.NormalizeHeight (float *worldHeight*)

Normalize a world height into a converted height.

Parameters

worldHeight	

Returns

5.24.1.4 static void uNature.Core.Utility.FoliageWorldMaps.ReGenerateGlobally ( ) [static]

Regenerate all of the world maps.

5.24.1.5 void uNature.Core.Utility.FoliageWorldMaps.RestoreChanges ( )

Reset the changes of runtime.

5.24.1.6 void uNature.Core.Utility.FoliageWorldMaps.Save ( )

Save the world maps

5.24.1.7 static void uNature.Core.Utility.FoliageWorldMaps.SaveAllMaps ( ) [static]

Save all of the world maps.

5.24.1.8 void uNature.Core.Utility.FoliageWorldMaps.SetPixels32Delayed ( )

Set pixels on the world maps, delayed.

5.24.1.9 static Vector3 uNature.Core.Utility.FoliageWorldMaps.TransformNormals (Vector3 *normal*) [static]

Normalize the world normals into converted normals.

Parameters

normal	

Returns

5.24.1.10 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightAndNormal (int *index,* float *height*)

Update height

Parameters

worldMap	

5.24.1.11 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightAndNormal (int *index,* Vector3 *normal*)

Update normal

Parameters

worldMap	

5.24.1.12 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightAndNormal (int *index,* float *height,* Vector3 *normal*)

Update height and normal

Parameters

worldMap	

5.24.1.13 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightsAndNormals (float *x*, float *z*, int *sizeX*, int *sizeZ*, bool *save*)

Update height on a certain range.

Parameters

x	
z	
sizeX	
sizeZ	

5.24.1.14 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightsAndNormals (bool *save*)

Update the heights and normals all over the map.

Parameters

save	

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.25 uNature.Core.FoliageClasses.GPUMesh Class Reference

A class used to hold the gpu meshes

Public Member Functions

- **GPUMesh** (Mesh[ ][ ] LODMeshes, FoliageMeshInstancesGroup LODMeshInstance, List< byte > LODLevels, int prototypeIndex, FoliageResolutions resolution)
- void **Destroy** ()
- int **GetMesh** (int density)

Public Attributes

- List< GPUMeshLOD > **meshes** = new List<GPUMeshLOD>()
- FoliageMeshInstancesGroup LODMeshInstances = null

  Dimension 1 : x chunk Dimension 2 : z chunk Dimension 3 : LOD index

5.25.1 Detailed Description

A class used to hold the gpu meshes

5.25.2 Member Data Documentation

5.25.2.1 FoliageMeshInstancesGroup uNature.Core.FoliageClasses.GPUMesh.LODMeshInstances = null

Dimension 1 : x chunk Dimension 2 : z chunk Dimension 3 : LOD index

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageMeshManager.cs

5.26 uNature.Core.FoliageClasses.GPUMeshLOD Class Reference

GPU Mesh Lods.

Public Member Functions

- **GPUMeshLOD** (Mesh[ ] _meshes, int _density, int _prototypeIndex)
- void **Destroy** ()

Public Attributes

- Mesh[ ] **meshes**
- int **density**

5.26.1 Detailed Description

GPU Mesh Lods.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageMeshManager.cs

5.27 uNature.Core.Sectors.GrassLODLevel Class Reference

A class that holds a level which all assigned on different frames.

Public Member Functions

- void **Add** (int x, int value, Vector2 pos)

Static Public Member Functions

- static GrassLODLevel **Create** ()

Public Attributes

- UNDimensionalList< int > **details** = new UNDimensionalList<int>()
- Vector2 **position** = new Vector2(Mathf.Infinity, Mathf.Infinity)

5.27.1 Detailed Description

A class that holds a level which all assigned on different frames.

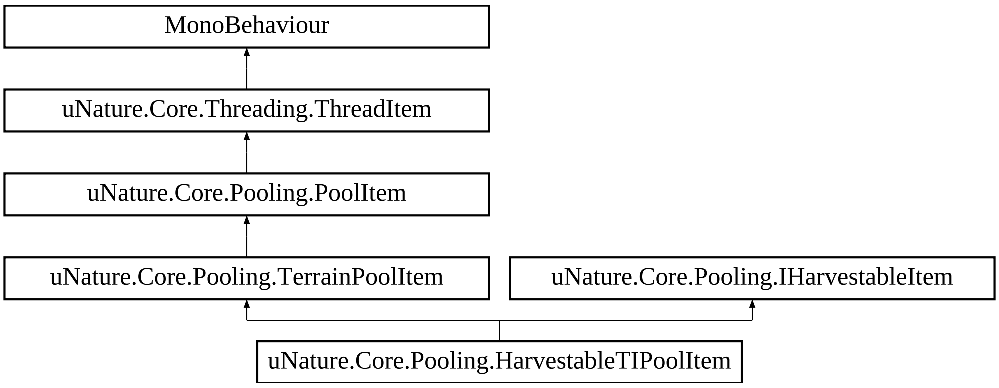
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Chunk.cs

5.28 uNature.Core.Pooling.HarvestableTIPoolItem Class Reference

A Pool item for terrain where the tree instances should be harvestable. (Tree cutting for instance) Inherite from this class to create your own harvestable type.

Inheritance diagram for uNature.Core.Pooling.HarvestableTIPoolItem:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Pooling.PoolItem
uNature.Core.Pooling.TerrainPoolItem
uNature.Core.Pooling.IHarvestableItem
uNature.Core.Pooling.HarvestableTIPoolItem

Public Member Functions

- override void Awake ()

  Called on awake.
- virtual void HandleHealthChange (int damage)

  Handle the health change, remove the current tree instance if needed, and instantiate a replacment.

Parameters

damage	*The amount of damage that the tree has received.*

- virtual void HandleColliderDeath ()

  Handle death of the colliders (remove tree instance from terrain and replace it with actual tree instance prefab)
- virtual void HandleTreeInstanceDeath ()

  Handle death of an actual tree instance (Add gravity and make it fall)

- virtual void Hit ()

    Hit this harvestable building and apply damage
- virtual void Hit (int damage)

    Hit this harvestable building and apply damage

Parameters

damage	*apply the damage*

- override void OnReturnedToPool ()

    Called when the item returns to the Pool, reset the propoties
- override void OnPool ()

    Called when the item pulled to the Pool

Public Attributes

- int minHealth = 0

    the minimum health possible to be assigned to the tree instance (For example - 0).
- int maxHealth = 100

    The maximum amount of health that can be assigned to this tree instance, which will also be assigned on default (For example - 100).
- float **respawnTimeInMinutes** = 2
- float **minFallDisappearTime** = 2
- float **maxFallDisappearTime** = 10

Static Public Attributes

- static bool canHarvestCollider = true

    Can this machine harvest a COLLIDER ?

Properties

- int **health** `[get, set]`

Events

- static OnItemStateChanged OnItemPooledEvent

    Called when an HarvestableTreeInstance has been Pooled
- static OnItemDamaged OnItemDamagedEvent

    Called when any harvestable item has been damaged.
- static OnItemStateChanged OnItemReturnedToPoolEvent

    Called when an HarvestableTreeInstance has been returned to Pool
- OnHealthChanged **OnHealthChangedEvent**

Additional Inherited Members

5.28.1 Detailed Description

A Pool item for terrain where the tree instances should be harvestable. (Tree cutting for instance) Inherite from this class to create your own harvestable type.

5.28.2 Member Function Documentation

5.28.2.1 override void uNature.Core.Pooling.HarvestableTIPoolItem.Awake ( ) [virtual]

Called on awake.

Reimplemented from uNature.Core.Pooling.PoolItem.

5.28.2.2 virtual void uNature.Core.Pooling.HarvestableTIPoolItem.HandleColliderDeath ( ) [virtual]

Handle death of the colliders (remove tree instance from terrain and replace it with actual tree instance prefab)

5.28.2.3 virtual void uNature.Core.Pooling.HarvestableTIPoolItem.HandleHealthChange (int *damage*) [virtual]

Handle the health change, remove the current tree instance if needed, and instantiate a replacment.

Parameters

damage	The amount of damage that the tree has received.

5.28.2.4 virtual void uNature.Core.Pooling.HarvestableTIPoolItem.HandleTreeInstanceDeath ( ) [virtual]

Handle death of an actual tree instance (Add gravity and make it fall)

5.28.2.5 virtual void uNature.Core.Pooling.HarvestableTIPoolItem.Hit ( ) [virtual]

Hit this harvestable building and apply damage

Implements uNature.Core.Pooling.IHarvestableItem.

5.28.2.6 virtual void uNature.Core.Pooling.HarvestableTIPoolItem.Hit (int *damage*) [virtual]

Hit this harvestable building and apply damage

Parameters

damage	apply the damage

Implements uNature.Core.Pooling.IHarvestableItem.

5.28.2.7 override void uNature.Core.Pooling.HarvestableTIPoolItem.OnPool ( ) [virtual]

Called when the item pulled to the Pool

Reimplemented from uNature.Core.Pooling.PoolItem.

5.28.2.8 override void uNature.Core.Pooling.HarvestableTIPoolItem.OnReturnedToPool ( ) [virtual]

Called when the item returns to the Pool, reset the propoties

Reimplemented from uNature.Core.Pooling.PoolItem.

5.28.3 Member Data Documentation

5.28.3.1 bool uNature.Core.Pooling.HarvestableTIPoolItem.canHarvestCollider = true [static]

Can this machine harvest a COLLIDER ?

5.28.3.2 int uNature.Core.Pooling.HarvestableTIPoolItem.maxHealth = 100

The maximum amount of health that can be assigned to this tree instance, which will also be assigned on default (For example - 100).

5.28.3.3 int uNature.Core.Pooling.HarvestableTIPoolItem.minHealth = 0

the minimum health possible to be assigned to the tree instance (For example - 0).

5.28.4 Event Documentation

5.28.4.1 OnItemDamaged uNature.Core.Pooling.HarvestableTIPoolItem.OnItemDamagedEvent [static]

Called when any harvestable item has been damaged.

5.28.4.2 OnItemStateChanged uNature.Core.Pooling.HarvestableTIPoolItem.OnItemPooledEvent [static]

Called when an HarvestableTreeInstance has been Pooled

5.28.4.3 OnItemStateChanged uNature.Core.Pooling.HarvestableTIPoolItem.OnItemReturnedToPoolEvent [static]

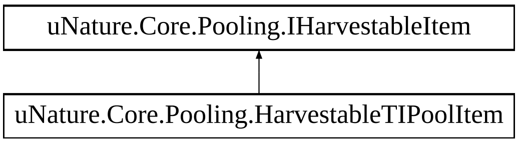
Called when an HarvestableTreeInstance has been returned to Pool

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/PoolItems/HarvestableTIPoolItem.cs

5.29 uNature.Core.Pooling.IHarvestableItem Interface Reference

Inheritance diagram for uNature.Core.Pooling.IHarvestableItem:


uNature.Core.Pooling.IHarvestableItem
uNature.Core.Pooling.HarvestableTIPoolItem

Public Member Functions

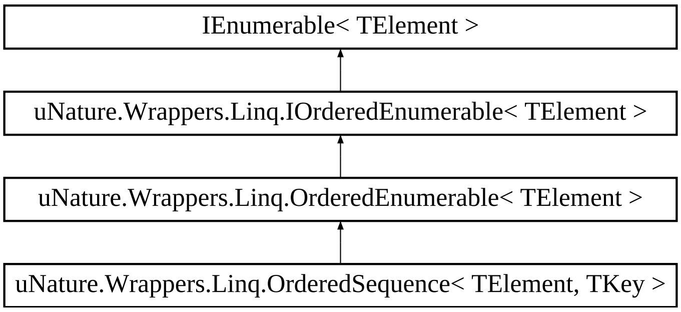
- void **Hit** ()
- void **Hit** (int damage)

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/PoolItems/HarvestableTIPoolItem.cs

5.30 uNature.Wrappers.Linq.IOrderedEnumerable< TElement > Interface Template Reference

Inheritance diagram for uNature.Wrappers.Linq.IOrderedEnumerable< TElement >:


IEnumerable< TElement >
uNature.Wrappers.Linq.IOrderedEnumerable< TElement >
uNature.Wrappers.Linq.OrderedEnumerable< TElement >
uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >

Public Member Functions

- IOrderedEnumerable< TElement > **CreateOrderedEnumerable< TKey >** (Func< TElement, TKey > keySelector, IComparer< TKey > comparer, bool descending)

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

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.31 uNature.Core.Pooling.IPoolComponent Interface Reference

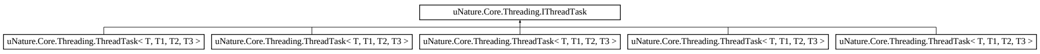
The documentation for this interface was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/Abstracts/IPoolComponent.cs

5.32 uNature.Core.Threading.IThreadTask Interface Reference

A thread task interface. Implement on any customely created thread task.

Inheritance diagram for uNature.Core.Threading.IThreadTask:


uNature.Core.Threading.IThreadTask
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >

Public Member Functions

- void **Invoke** ()

Properties

- int **creationFrame** `[get]`

5.32.1 Detailed Description

A thread task interface. Implement on any customely created thread task.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.33 uNature.Core.IUTCPhysicsIgnored Interface Reference

Ignore all physics on this script.

Properties

- bool **ignore** `[get]`

5.33.1 Detailed Description

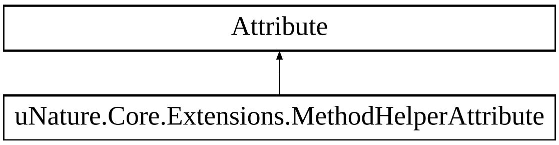
Ignore all physics on this script.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysics.cs

5.34 uNature.Core.Extensions.MethodHelperAttribute Class Reference

Inheritance diagram for uNature.Core.Extensions.MethodHelperAttribute:

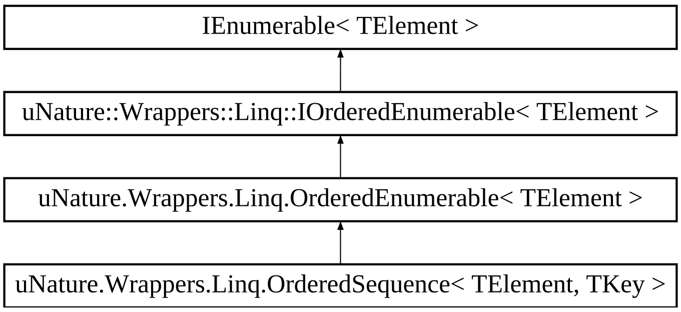


The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Base/UNExtension.cs

5.35 uNature.Wrappers.Linq.OrderedEnumerable< TElement > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.OrderedEnumerable< TElement >:


IEnumerable< TElement >
uNature::Wrappers::Linq::IOrderedEnumerable< TElement >
uNature.Wrappers.Linq.OrderedEnumerable< TElement >
uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >

Public Member Functions

- virtual IEnumerator< TElement > **GetEnumerator** ()
- abstract SortContext< TElement > **CreateContext** (SortContext< TElement > current)
- IOrderedEnumerable< TElement > **CreateOrderedEnumerable**< **TKey** > (Func< TElement, TKey > selector, IComparer< TKey > comparer, bool descending)

Protected Member Functions

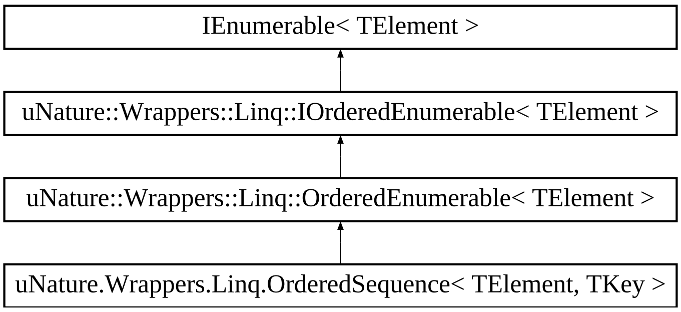
- **OrderedEnumerable** (IEnumerable< TElement > source)
- abstract IEnumerable< TElement > **Sort** (IEnumerable< TElement > source)

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

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.36 uNature.Wrappers.Linq.OrderedSequence< TElement, TKey > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >:


IEnumerable< TElement >
uNature::Wrappers::Linq::IOrderedEnumerable< TElement >
uNature::Wrappers::Linq::OrderedEnumerable< TElement >
uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >

Public Member Functions

- override IEnumerator< TElement > **GetEnumerator** ()
- override SortContext< TElement > **CreateContext** (SortContext< TElement > current)

Protected Member Functions

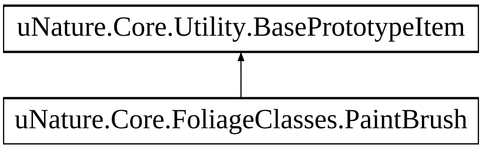
- override IEnumerable< TElement > **Sort** (IEnumerable< TElement > source)

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

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.37 uNature.Core.FoliageClasses.PaintBrush Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.PaintBrush:


uNature.Core.Utility.BasePrototypeItem
uNature.Core.FoliageClasses.PaintBrush

Public Member Functions

- **PaintBrush** (Texture2D _texture)
- void **TryToResize** (int size)

Public Attributes

- Texture2D **brushTexture**

Protected Member Functions

- override Texture2D **GetPreview** ()

Properties

- Texture2D **instancedTexture** `[get]`
- Color32[,] **pixels** `[get]`

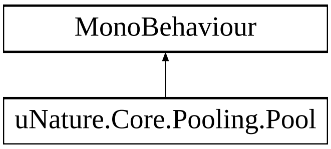
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/PaintBrush.cs

5.38 uNature.Core.Pooling.Pool Class Reference

A class that manages the Pooling of the system, Which allows huge runtime performance increase.

Inheritance diagram for uNature.Core.Pooling.Pool:



Public Member Functions

- void AddToPool (PoolItem item, int itemID, int itemID_Offset)

  Add an item to the Pool.
- void RemoveFromPool (PoolItem item)

  Remove an item from the Pool
- void ReturnToPool (PoolItem item, bool force)

  Return a certain item to Pool.
- void TryResetOnUID (int uid, bool forceReset)

  Reset a certain item which is on a certain UID

Parameters

uid	*the targeted UID*
forceReset	*Force reset will make it ignore the locked state of the item.*

- T TryPool< T > (int itemUID, int itemID_Offset, int uid, bool locked)

  Try to Pool an item, will return null if no target is found.
- List< PoolItem > GetPoolOfItem (int itemUID, int itemID_Offset)

  Get Pool of a certain item
- void PoolItem (PoolItem PoolItem, bool locked, int uid)

  Pool the certain item.
- void ResetFarAway ()

  This method will find and reset far away items to be "recycled"
- bool IsAlreadyPooled (int uid)

  Check if a certain uid is already Pooled.
- T TryGetType< T > ()

  Try to get an object from the Pool with a certain component.

Static Public Member Functions

- static Pool CreatePool (string name, GameObject requester)

  Create a new Pool
- static void RemoveDuplications (string name)

  Remove Pool duplications.

Public Attributes

- List< PoolItem > items = new List<PoolItem>()

  A list that holds all of the Pool items in our Pool.
- GameObject owner

  Who created this Pool?

5.38.1 Detailed Description

A class that manages the Pooling of the system, Which allows huge runtime performance increase.

5.38.2 Member Function Documentation

5.38.2.1 void uNature.Core.Pooling.Pool.AddToPool (PoolItem *item,* int *itemID,* int *itemID_Offset*)

Add an item to the Pool.

Parameters

item	the item.
itemID	The targeted item id
itemID_Offset	The offset of the item id to make it unique.

5.38.2.2 static **Pool** uNature.Core.Pooling.Pool.CreatePool (string *name,* GameObject *requester*) [static]

Create a new Pool

Parameters

name	the Pool name (Without Pool at the end)
requester	who is the owner of this Pool

Returns

the newely created Pool.

5.38.2.3 List<**PoolItem**> uNature.Core.Pooling.Pool.GetPoolOfItem (int *itemUID,* int *itemID_Offset*)

Get Pool of a certain item

Parameters

itemUID	
itemID_Offset	

Returns

5.38.2.4 bool uNature.Core.Pooling.Pool.IsAlreadyPooled (int *uid*)

Check if a certain uid is already Pooled.

Parameters

uid	the uid of the targeted item

Returns

is this item already Pooled?

5.38.2.5 void uNature.Core.Pooling.Pool.PoolItem (**PoolItem** *PoolItem,* bool *locked,* int *uid*)

Pool the certain item.

Template Parameters

T	

Parameters

PoolItem	

5.38.2.6 static void uNature.Core.Pooling.Pool.RemoveDuplications (string *name*) [static]

Remove Pool duplications.

Parameters

name	

5.38.2.7 void uNature.Core.Pooling.Pool.RemoveFromPool (PoolItem *item*)

Remove an item from the Pool

Parameters

item	the item.

5.38.2.8 void uNature.Core.Pooling.Pool.ResetFarAway ( )

This method will find and reset far away items to be "recycled"

5.38.2.9 void uNature.Core.Pooling.Pool.ReturnToPool (PoolItem *item,* bool *force*)

Return a certain item to Pool.

Parameters

item	the item.

\

Parameters

force	making force true, will make the system ignore the locked state of the item. (if exists)

5.38.2.10 T uNature.Core.Pooling.Pool.TryGetType< T > ( )

Try to get an object from the Pool with a certain component.

Template Parameters

T	the type of the component

Returns

Type Constraints

T* : *Component

5.38.2.11 T uNature.Core.Pooling.Pool.TryPool< T > (int *itemUID,* int *itemID_Offset,* int *uid,* bool *locked*)

Try to Pool an item, will return null if no target is found.

Parameters

itemUID	the uid of the item (without offset)
itemID_Offset	the offset of the required item id
uid	a unique id of the object which will be attached to this game object. (HAS TO BE UNIQUE...)
locked	if the Pool item is locked, it wont be able to return to Pool unless its unlocked.

Returns

A Pool item.

Type Constraints

T* : *PoolItem

5.38.2.12 void uNature.Core.Pooling.Pool.TryResetOnUID (int *uid,* bool *forceReset*)

Reset a certain item which is on a certain UID

Parameters

uid	the targeted UID
forceReset	Force reset will make it ignore the locked state of the item.

5.38.3 Member Data Documentation

5.38.3.1 List<PoolItem> uNature.Core.Pooling.Pool.items = new List<PoolItem>()

A list that holds all of the Pool items in our Pool.

5.38.3.2 GameObject uNature.Core.Pooling.Pool.owner

Who created this Pool?

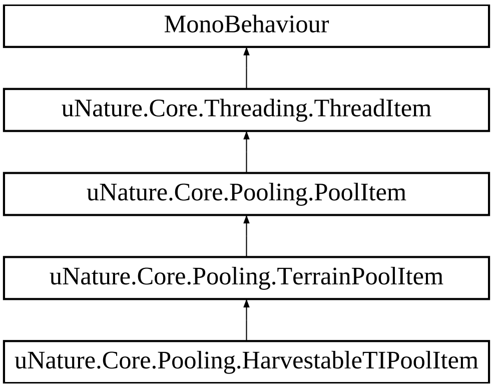
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/Pool.cs

5.39 uNature.Core.Pooling.PoolItem Class Reference

An abstract class that handles the Pool items.

Inheritance diagram for uNature.Core.Pooling.PoolItem:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Pooling.PoolItem
uNature.Core.Pooling.TerrainPoolItem
uNature.Core.Pooling.HarvestableTIPoolItem

Public Member Functions

- virtual void Awake ()

  Called on awake.
- virtual void OnPool ()

  Called when the item has been Pooled.
- virtual void OnReturnedToPool ()

  Called when the item has returned to the Pool
- virtual void OnCreated ()

  Called when the item has been created.
- virtual void MoveItem (Vector3 position)

  Move the item to a certain position. NOTE: in order to move the item, use this method and DONT change the position externally!!

Public Attributes

- Pool Pool

  What Pool are we belonged to?
- GameObject _gameObject

  An gameobject reference which can be used on a different thread.
- bool used

  is the item currently used?
- bool locked

  Is this Pool item locked? If so, dont let it return back to Pool unless forced.
- int realItemID

  The Pool item unique id, which is used to identify the item. (not including offset)
- int itemID_Offset

  The offset of the item id which allows the item id to be more unique. Can be left 0.
- int uid = -1

  What is the uid of the item we are attached to.

Protected Member Functions

- override void OnEnable ()

  Called when the object is enabled.
- override void OnDisable ()

  Called when the object is disabled.

Properties

- static System.Type[ ] PoolTypes `[get]`

  All the Pool types in the assembly.
- int itemID `[get]`

  The Pool item unique id, which is used to identify the item. (including offset)

Additional Inherited Members

5.39.1 Detailed Description

An abstract class that handles the Pool items.

5.39.2 Member Function Documentation

5.39.2.1 virtual void uNature.Core.Pooling.PoolItem.Awake ( ) `[virtual]`

Called on awake.

Reimplemented in uNature.Core.Pooling.HarvestableTIPoolItem.

5.39.2.2 virtual void uNature.Core.Pooling.PoolItem.MoveItem (Vector3 *position*) `[virtual]`

Move the item to a certain position. NOTE: in order to move the item, use this method and DONT change the position externally!!

Parameters

position	target position.

Reimplemented in uNature.Core.Pooling.TerrainPoolItem.

5.39.2.3 virtual void uNature.Core.Pooling.PoolItem.OnCreated ( ) [virtual]

Called when the item has been created.

5.39.2.4 override void uNature.Core.Pooling.PoolItem.OnDisable ( ) [protected], [virtual]

Called when the object is disabled.

Reimplemented from uNature.Core.Threading.ThreadItem.

5.39.2.5 override void uNature.Core.Pooling.PoolItem.OnEnable ( ) [protected], [virtual]

Called when the object is enabled.

Reimplemented from uNature.Core.Threading.ThreadItem.

5.39.2.6 virtual void uNature.Core.Pooling.PoolItem.OnPool ( ) [virtual]

Called when the item has been Pooled.

Reimplemented in uNature.Core.Pooling.HarvestableTIPoolItem.

5.39.2.7 virtual void uNature.Core.Pooling.PoolItem.OnReturnedToPool ( ) [virtual]

Called when the item has returned to the Pool

Reimplemented in uNature.Core.Pooling.HarvestableTIPoolItem.

5.39.3 Member Data Documentation

5.39.3.1 GameObject uNature.Core.Pooling.PoolItem._gameObject

An gameobject reference which can be used on a different thread.

5.39.3.2 int uNature.Core.Pooling.PoolItem.itemID_Offset

The offset of the item id which allows the item id to be more unique. Can be left 0.

5.39.3.3 bool uNature.Core.Pooling.PoolItem.locked

Is this Pool item locked? If so, dont let it return back to Pool unless forced.

5.39.3.4 Pool uNature.Core.Pooling.PoolItem.Pool

What Pool are we belonged to?

5.39.3.5 int uNature.Core.Pooling.PoolItem.realItemID

The Pool item unique id, which is used to identify the item. (not including offset)

5.39.3.6 int uNature.Core.Pooling.PoolItem.uid = -1

What is the uid of the item we are attached to.

5.39.3.7 bool uNature.Core.Pooling.PoolItem.used

is the item currently used?

5.39.4 Property Documentation

5.39.4.1 int uNature.Core.Pooling.PoolItem.itemID [get]

The Pool item unique id, which is used to identify the item. (including offset)

5.39.4.2 System.Type [ ] uNature.Core.Pooling.PoolItem.PoolTypes [static], [get]

All the Pool types in the assembly.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/Abstracts/PoolItem.cs

5.40 uNature.Wrappers.Linq.QuickSort< TElement > Class Template Reference

Static Public Member Functions

- static IEnumerable< TElement > **Sort** (IEnumerable< TElement > source, SortContext< TElement > context)

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

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.41 uNature.Core.FoliageClasses.RenderingQueue Class Reference

An rendering queue.

Public Member Functions

- **RenderingQueue** (FoliageManagerInstance mInstance, FoliageCore_Chunk mChunk, Vector3 snapPosition, Dictionary< int, RenderingQueueInstance > fetchedQueueInstance)
- void **DrawDebug** ()
- UNFastList< RenderingQueueMeshInstanceSimulator > **GetFromPool** (FoliageManagerInstance m←Instance, FoliagePrototype prototype)
- void **AddToPool** (UNFastList< RenderingQueueMeshInstanceSimulator > obj, FoliagePrototype prototype)

Public Attributes

- const int **GENERATION_RADIUS** = 3
- const int **GENERATION_RADIUS_OFFSET** = 1
- FoliageManagerInstance **mInstance**
- Dictionary< int, RenderingQueueInstance > **queueInstance**
- bool **queueInstanceNull** = true

Static Public Attributes

- static Dictionary< int, Stack< UNFastList< RenderingQueueMeshInstanceSimulator > > > **Rendering←QueueInstancesPool**

5.41.1 Detailed Description

An rendering queue.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.42 uNature.Core.FoliageClasses.RenderingQueueInstance Class Reference

An rendering queue instance.

Public Member Functions

- **RenderingQueueInstance** (FoliageManagerInstance mInstance, FoliagePrototype prototype, GPUMesh gpuMesh, FoliageGrassMap grassMap)
- void **DrawDebug** ()

Static Public Attributes

- static int **count** = 0

5.42.1 Detailed Description

An rendering queue instance.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.43 uNature.Core.FoliageClasses.RenderingQueueMeshInstanceSimulator Struct Reference

An imposter struct to simulate an instance.

Public Member Functions

- **RenderingQueueMeshInstanceSimulator** (Vector3 position, FoliageMeshInstance _meshInstance, FoliageManagerInstance mInstance, FoliagePrototype _prototype)
- RenderingQueueMeshInstanceSimulator **UpdateDensity** (GPUMesh gpuMesh, byte maxDensity, Foliage↩ GrassMap grassMap, int mapWidth, float densityMultiplier)
- void **Render** (MaterialPropertyBlock mBlock, Camera camera, Plane[ ] cameraPlanes, Vector3 normalized↩ CameraPosition, bool isPlaying, bool useQualitySettingsShadow, float shadowsDistance)
- void **DrawDebug** ()

Public Attributes

- float **x**
- float **z**
- float **worldX**
- float **worldZ**

Properties

- byte **density** `[get]`
- Mesh **mesh** `[get]`

5.43.1 Detailed Description

An imposter struct to simulate an instance.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.44 uNature.Core.FoliageClasses.RenderingQueueReceiver Class Reference

An object that can handle an object queue.

Public Member Functions

- void **CheckPositionChange** ()
- void **ResetDensity** ()

Public Attributes

- Transform **transform**
- Camera **camera**

Properties

- FoliageCore_Chunk[ ] **neighbors** `[get, set]`

5.44.1 Detailed Description

An object that can handle an object queue.

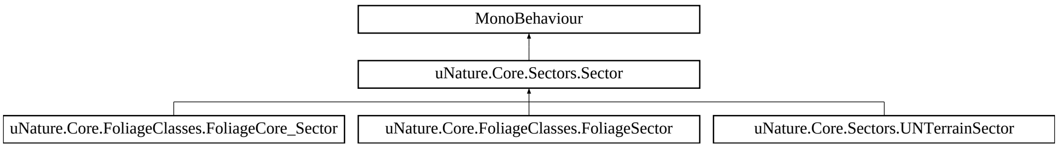
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.45 uNature.Core.Sectors.Sector Class Reference

A sector which is used to divide the UNTerrain objects in the world to increase performance (can handle more than 200k trees!!)

Inheritance diagram for uNature.Core.Sectors.Sector:


MonoBehaviour
uNature.Core.Sectors.Sector
uNature.Core.FoliageClasses.FoliageCore_Sector
uNature.Core.FoliageClasses.FoliageSector
uNature.Core.Sectors.UNTerrainSector

Public Member Functions

- virtual void OnCreated (Transform owner, int resolution)

    Called when the object is created.

Parameters

terrain	*The terrain we belong to.*

- virtual void Awake ()

    Called on awake.
- void ResetChunks ()

    This method will reset the chunks' propoties, so it can be used again instead of recreating the whole sector.
- Chunk getChunk (Vector2 pos, float offset)

    Get a chunk on a certain local space position
- Chunk getChunk (Vector3 pos, float offset)

    Get a chunk on a certain local space position
- Chunk getChunk (Vector3 pos)

    Get a chunk on a certain local space position
- List< Chunk > getChunks (Vector2 pos, float offset, bool sortResult)

    Get all of the chunks that contains this specific position
- List< Chunk > getChunks (Vector3 pos, float offset, bool sortResult)

    Get all of the chunks that contains this specific position
- virtual void ApplicationQuit ()

    This method will be called when the application quits, used to revert all changes on terrain.

Static Public Member Functions

- static T GenerateSector< T, T1 > (Transform owner, Vector3 bounds, T sector, int res)

    Generate a new sector

Public Attributes

- const int **resolutionLimit** = 40
- int **sectorResolution**
- Transform **sectorOwner**
- List< Chunk > **chunks** = new List<Chunk>()

Protected Member Functions

- virtual void OnResolutionChanged ()

    Called when the resolution has been updated.
- virtual void OnChunkCreated (Chunk chunk)

    Called when a chunk is created to allow custom logic on the inherited sectors.
- virtual void OnStartCreatingChunks ()

    Called right before starting to create the chunks.

Properties

- Vector2 **chunkSize** `[get]`

Events

- SectorRecalculated **OnSectorRecalculated**

5.45.1 Detailed Description

A sector which is used to divide the UNTerrain objects in the world to increase performance (can handle more than 200k trees!!)

5.45.2 Member Function Documentation

5.45.2.1 virtual void uNature.Core.Sectors.Sector.ApplicationQuit ( ) [virtual]

This method will be called when the application quits, used to revert all changes on terrain.

Reimplemented in uNature.Core.Sectors.UNTerrainSector.

5.45.2.2 virtual void uNature.Core.Sectors.Sector.Awake ( ) [virtual]

Called on awake.

Parameters

terrain	The terrain we belong to

Reimplemented in uNature.Core.Sectors.UNTerrainSector.

5.45.2.3 static T uNature.Core.Sectors.Sector.GenerateSector< T, T1 > (Transform *owner,* Vector3 *bounds,* T *sector,* int *res*) [static]

Generate a new sector

Parameters

terrain	The terrain this sector will be generated on
res	the resolution of the sector (how many times will it be sliced

Returns

The new generated sector.

Type Constraints

T* : *Sector

T1* : *Chunk

5.45.2.4 Chunk uNature.Core.Sectors.Sector.getChunk (Vector2 *pos,* float *offset*)

Get a chunk on a certain local space position

Parameters

pos	the local space position
offset	the offset (The bigger it is, the farder chunks it will find)

Returns

5.45.2.5 Chunk uNature.Core.Sectors.Sector.getChunk (Vector3 *pos,* float *offset*)

Get a chunk on a certain local space position

Parameters

pos	the local space position
offset	the offset (The bigger it is, the farder chunks it will find)

Returns

5.45.2.6 Chunk uNature.Core.Sectors.Sector.getChunk (Vector3 *pos*)

Get a chunk on a certain local space position

Parameters

pos	the local space position
offset	the offset (The bigger it is, the farder chunks it will find)

Returns

5.45.2.7 List<Chunk> uNature.Core.Sectors.Sector.getChunks (Vector2 *pos,* float *offset,* bool *sortResult*)

Get all of the chunks that contains this specific position

Parameters

pos	a local space position
offset	the offset (The bigger it is, the farder chunks it will find)

Returns

The chunks that contains the local space position

5.45.2.8 List<Chunk> uNature.Core.Sectors.Sector.getChunks (Vector3 *pos,* float *offset,* bool *sortResult*)

Get all of the chunks that contains this specific position

Parameters

pos	a local space position
offset	the offset (The bigger it is, the farder chunks it will find)

Returns

The chunks that contains the local space position

5.45.2.9 virtual void uNature.Core.Sectors.Sector.OnChunkCreated (Chunk *chunk*) [protected], [virtual]

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

chunk	

Reimplemented in uNature.Core.Sectors.UNTerrainSector, uNature.Core.FoliageClasses.FoliageSector, and uNature.Core.FoliageClasses.FoliageCore_Sector.

5.45.2.10 virtual void uNature.Core.Sectors.Sector.OnCreated (Transform *owner,* int *resolution*) [virtual]

Called when the object is created.

Parameters

terrain	The terrain we belong to.

Reimplemented in uNature.Core.Sectors.UNTerrainSector.

5.45.2.11 virtual void uNature.Core.Sectors.Sector.OnResolutionChanged ( ) [protected], [virtual]

Called when the resolution has been updated.

Reimplemented in uNature.Core.FoliageClasses.FoliageCore_Sector.

5.45.2.12 virtual void uNature.Core.Sectors.Sector.OnStartCreatingChunks ( ) [protected], [virtual]

Called right before starting to create the chunks.

Reimplemented in uNature.Core.Sectors.UNTerrainSector, uNature.Core.FoliageClasses.FoliageSector, and uNature.Core.FoliageClasses.FoliageCore_Sector.

5.45.2.13 void uNature.Core.Sectors.Sector.ResetChunks ( )

This method will reset the chunks' propoties, so it can be used again instead of recreating the whole sector.

Resets:

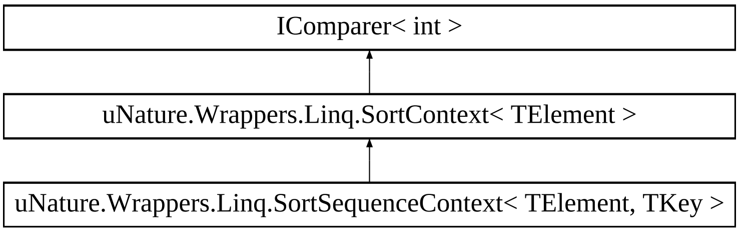
TreeInstances

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Sector.cs

5.46 uNature.Wrappers.Linq.SortContext< TElement > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.SortContext< TElement >:


IComparer< int >
uNature.Wrappers.Linq.SortContext< TElement >
uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey >

Public Member Functions

- abstract void **Initialize** (TElement[ ] elements)
- abstract int **Compare** (int first_index, int second_index)

Protected Member Functions

- **SortContext** (SortDirection direction, SortContext< TElement > child_context)

Protected Attributes

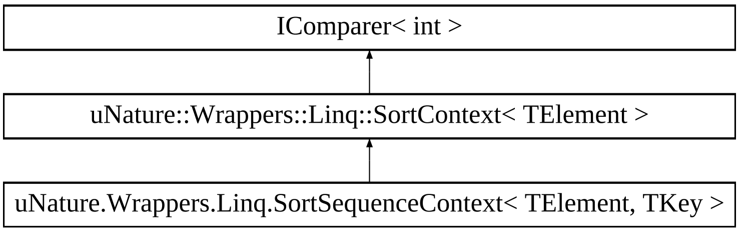
- SortDirection **direction**
- SortContext< TElement > **child_context**

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

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.47 uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey >:


IComparer< int >
uNature::Wrappers::Linq::SortContext< TElement >
uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey >

Public Member Functions

- **SortSequenceContext** (Func< TElement, TKey > selector, IComparer< TKey > comparer, SortDirection direction, SortContext< TElement > child_context)
- override void **Initialize** (TElement[ ] elements)
- override int **Compare** (int first_index, int second_index)

Additional Inherited Members

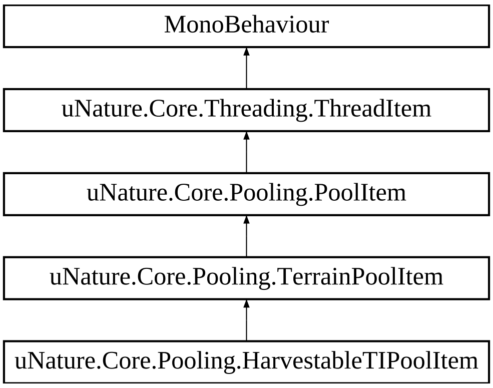
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.48 uNature.Core.Pooling.TerrainPoolItem Class Reference

A Pool item for terrain. (Tree instances)

Inheritance diagram for uNature.Core.Pooling.TerrainPoolItem:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Pooling.PoolItem
uNature.Core.Pooling.TerrainPoolItem
uNature.Core.Pooling.HarvestableTIPoolItem

Public Member Functions

- override void MoveItem (Vector3 position)

  Move with rigidbody to avoid colliders movement.

Static Public Member Functions

- static void RemoveTreeInstanceFromTerrain (Terrain terrain, int treeInstanceUID)

  Remove a tree instance from the terrain, Allowing you to replace it with anything else - for instance, the actual game object of the tree.
- static void ConvertTreeInstanceOnTerrain (Terrain terrain, int treeInstanceUID)

  Remove a tree instance from the terrain, And replace it with a Pool item.
- static void RestoreTreeInstanceToTerrain (Terrain terrain, int treeInstanceUID)

  Restore the tree instance back into the terrain.

Public Attributes

- bool isCollider

  is this instance a collider ? or an actual tree instance ?
- Terrain _terrain

  The terrain which owns this Pool item.

Static Public Attributes

- static bool canModify = true

  Can this machine modify tree instances?
- static bool canRestore = true

  Can this machine restore tree instances?

Properties

- Rigidbody **rigid** `[get]`
- Terrain **terrain** `[get]`

Events

- static OnTreeInstanceStateChanged **OnTreeInstanceRemoved**
- static OnTreeInstanceStateChanged **OnTreeInstanceRestored**

Additional Inherited Members

5.48.1 Detailed Description

A Pool item for terrain. (Tree instances)

5.48.2 Member Function Documentation

5.48.2.1 static void uNature.Core.Pooling.TerrainPoolItem.ConvertTreeInstanceOnTerrain (Terrain *terrain,* int *treeInstanceUID*) `[static]`

Remove a tree instance from the terrain, And replace it with a Pool item.

5.48.2.2 override void uNature.Core.Pooling.TerrainPoolItem.MoveItem (Vector3 *position*) `[virtual]`

Move with rigidbody to avoid colliders movement.

Parameters

position	target position

Reimplemented from uNature.Core.Pooling.PoolItem.

5.48.2.3 static void uNature.Core.Pooling.TerrainPoolItem.RemoveTreeInstanceFromTerrain (Terrain *terrain*, int *treeInstanceUID*) [static]

Remove a tree instance from the terrain, Allowing you to replace it with anything else - for instance, the actual game object of the tree.

5.48.2.4 static void uNature.Core.Pooling.TerrainPoolItem.RestoreTreeInstanceToTerrain (Terrain *terrain*, int *treeInstanceUID*) [static]

Restore the tree instance back into the terrain.

5.48.3 Member Data Documentation

5.48.3.1 Terrain uNature.Core.Pooling.TerrainPoolItem._terrain

The terrain which owns this Pool item.

5.48.3.2 bool uNature.Core.Pooling.TerrainPoolItem.canModify = true [static]

Can this machine modify tree instances?

5.48.3.3 bool uNature.Core.Pooling.TerrainPoolItem.canRestore = true [static]

Can this machine restore tree instances?

5.48.3.4 bool uNature.Core.Pooling.TerrainPoolItem.isCollider

is this instance a collider ? or an actual tree instance ?

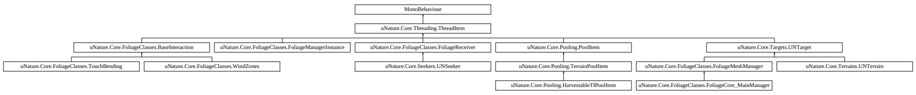
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/PoolItems/TerrainPoolItem.cs

5.49 uNature.Core.Threading.ThreadItem Class Reference

This class handles assigning parameters before multi-threaded actions that can be called from outside of unity's main thread. for example : position.

Inheritance diagram for uNature.Core.Threading.ThreadItem:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.BaseInteraction
uNature.Core.FoliageClasses.FoliageManagerInstance
uNature.Core.FoliageClasses.FoliageReceiver
uNature.Core.Pooling.PoolItem
uNature.Core.Targets.UNTarget
uNature.Core.FoliageClasses.TouchBending
uNature.Core.FoliageClasses.WindZones
uNature.Core.Seekers.UNSeeker
uNature.Core.Pooling.TerrainPoolItem
uNature.Core.FoliageClasses.FoliageMeshManager
uNature.Core.Terrains.UNTerrain
uNature.Core.Pooling.HarvestableTIPoolItem
uNature.Core.FoliageClasses.FoliageCore_MainManager

Public Member Functions

- virtual void UpdateItem ()

  This method will update this thread item, called externally from unity's main thread.

Static Public Attributes

- static List< ThreadItem > _threadItems

  A list that holds all of the thread items in the scene.

Protected Member Functions

- virtual void **OnEnable** ()
- virtual void OnDisable ()

  Called when the object is disabled
- virtual void Update ()

  Update...
- virtual void OnPositionChanged (Vector3 newPosition)

  Called when the item's position changed

Properties

- static List< ThreadItem > **threadItems** `[get]`
- Vector3 **threadPosition** `[get, set]`
- Vector2 **threadPositionDepth** `[get, set]`

5.49.1 Detailed Description

This class handles assigning parameters before multi-threaded actions that can be called from outside of unity's main thread. for example : position.

5.49.2 Member Function Documentation

5.49.2.1 virtual void uNature.Core.Threading.ThreadItem.OnDisable ( ) [protected], [virtual]

Called when the object is disabled

Reimplemented in uNature.Core.Terrains.UNTerrain, uNature.Core.FoliageClasses.FoliageManagerInstance, uNature.Core.FoliageClasses.FoliageCore_MainManager, uNature.Core.FoliageClasses.FoliageMeshManager, uNature.Core.FoliageClasses.TouchBending, uNature.Core.Targets.UNTarget, uNature.Core.FoliageClasses.FoliageReceiver, uNature.Core.Pooling.PoolItem, and uNature.Core.FoliageClasses.BaseInteraction.

5.49.2.2 virtual void uNature.Core.Threading.ThreadItem.OnPositionChanged (Vector3 *newPosition*) [protected], [virtual]

Called when the item's position changed

Reimplemented in uNature.Core.Terrains.UNTerrain, uNature.Core.FoliageClasses.TouchBending, and uNature.Core.FoliageClasses.BaseInteraction.

5.49.2.3 virtual void uNature.Core.Threading.ThreadItem.Update ( ) [protected], [virtual]

Update...

Reimplemented in uNature.Core.FoliageClasses.FoliageMeshManager, uNature.Core.FoliageClasses.FoliageCore_MainManager, uNature.Core.FoliageClasses.FoliageReceiver, uNature.Core.Targets.UNTarget, and uNature.Core.Seekers.UNSeeker.

5.49.2.4 virtual void uNature.Core.Threading.ThreadItem.UpdateItem ( ) [virtual]

This method will update this thread item, called externally from unity's main thread.

5.49.3 Member Data Documentation

5.49.3.1 List<ThreadItem> uNature.Core.Threading.ThreadItem._threadItems [static]

A list that holds all of the thread items in the scene.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/ThreadItem.cs

5.50 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.50.1 Detailed Description

A thread task that takes no parameters.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.51 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.51.1 Detailed Description

A thread task that takes no parameters.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.52 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.52.1 Detailed Description

A thread task that takes no parameters.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.53 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.53.1 Detailed Description

A thread task that takes no parameters.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.54 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.54.1 Detailed Description

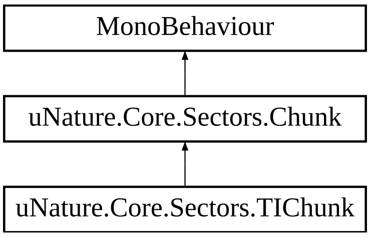
A thread task that takes no parameters.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.55 uNature.Core.Sectors.TIChunk Class Reference

Inheritance diagram for uNature.Core.Sectors.TIChunk:



Public Member Functions

- override void Awake ()

  Called on awake
- override void OnDrawGizmos ()

  Draw gizmos.
- override void OnCreated ()

  Called when created.
- virtual void GenerateTreeInstances (TreeInstance[ ] trees, Vector3 terrainSize, TerrainData tData, Vector3 terrainPos)

  Generate tree intances, derived from a certain provided tree instances
- void AddTreeInstance (int instanceID, Vector3 terrainSize, TreeInstance treeInstance, TerrainData terrainData, Vector3 terrainPos, UNTerrainSector sector)

  Add a tree instance into this chunk
- override void ResetChunk ()

  Reset chunk.
- void CheckForNearbyTreeInstances (UNSeeker seeker, UNTerrain terrain)

  Check and assign nearby tree instances.

Public Attributes

- List< int > **objectsInstanceIDs** = new List<int>()
- List< ChunkObject > **objects** = new List<ChunkObject>()

Protected Member Functions

- override void OnSizeChanged ()

    Called when the size is changed

Properties

- override string **chunkType** `[get]`
- Terrain **terrain** `[get, set]`

Additional Inherited Members

5.55.1 Member Function Documentation

5.55.1.1 void uNature.Core.Sectors.TIChunk.AddTreeInstance (int *instanceID,* Vector3 *terrainSize,* TreeInstance *treeInstance,* TerrainData *terrainData,* Vector3 *terrainPos,* UNTerrainSector *sector*)

Add a tree instance into this chunk

Parameters

instanceID	the targeted tree instance.
treeInstance	the tree instance you want to add
terrainData	the terrain data that this chunk belongs to

5.55.1.2 override void uNature.Core.Sectors.TIChunk.Awake ( ) `[virtual]`

Called on awake

Parameters

terrain	
terrainBase	

Reimplemented from uNature.Core.Sectors.Chunk.

5.55.1.3 void uNature.Core.Sectors.TIChunk.CheckForNearbyTreeInstances (UNSeeker *seeker,* UNTerrain *terrain*)

Check and assign nearby tree instances.

5.55.1.4 virtual void uNature.Core.Sectors.TIChunk.GenerateTreeInstances (TreeInstance[ ] *trees,* Vector3 *terrainSize,* TerrainData *tData,* Vector3 *terrainPos*) `[virtual]`

Generate tree intances, derived from a certain provided tree instances

Parameters

trees	the tree instances
tData	the terrain data

5.55.1.5 override void uNature.Core.Sectors.TIChunk.OnCreated ( ) [virtual]

Called when created.

Reimplemented from uNature.Core.Sectors.Chunk.

5.55.1.6 override void uNature.Core.Sectors.TIChunk.OnDrawGizmos ( ) [virtual]

Draw gizmos.

Reimplemented from uNature.Core.Sectors.Chunk.

5.55.1.7 override void uNature.Core.Sectors.TIChunk.OnSizeChanged ( ) [protected], [virtual]

Called when the size is changed

Reimplemented from uNature.Core.Sectors.Chunk.

5.55.1.8 override void uNature.Core.Sectors.TIChunk.ResetChunk ( ) [virtual]

Reset chunk.

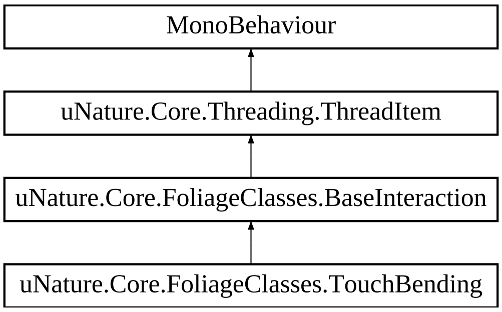
Reimplemented from uNature.Core.Sectors.Chunk.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/TIChunk.cs

5.56 uNature.Core.FoliageClasses.TouchBending Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.TouchBending:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.BaseInteraction
uNature.Core.FoliageClasses.TouchBending

Public Attributes

- bool **inBounds** = false

Static Public Attributes

- static Vector4[ ] **bendingTargets** = new Vector4[20]

Protected Member Functions

- override void **OnEnable** ()
- override void OnDisable ()

  Called when the object is disabled
- virtual void **OnDrawGizmos** ()
- override void OnPositionChanged (Vector3 newPosition)

  Called when the item's position changed

Properties

- bool **simulateOnEditorTime** `[get, set]`
- float **radius** `[get, set]`
- float **seekingRange** `[get, set]`
- int **id** `[get]`
- bool **simulate** `[get]`
- override bool **includedInInteractionMap** `[get]`

Additional Inherited Members

5.56.1 Member Function Documentation

5.56.1.1 override void uNature.Core.FoliageClasses.TouchBending.OnDisable ( ) `[protected]`, `[virtual]`

Called when the object is disabled

Reimplemented from uNature.Core.FoliageClasses.BaseInteraction.

5.56.1.2 override void uNature.Core.FoliageClasses.TouchBending.OnPositionChanged (Vector3 *newPosition*) `[protected]`, `[virtual]`

Called when the item's position changed

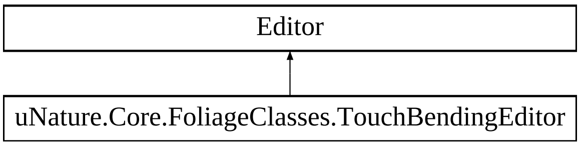
Reimplemented from uNature.Core.FoliageClasses.BaseInteraction.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/Interaction/TouchBending.cs

5.57 uNature.Core.FoliageClasses.TouchBendingEditor Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.TouchBendingEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

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

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/TouchBendingEditor.cs

5.58 uNature.Core.Sectors.TreeFetchingTask_MultiThreaded Struct Reference

Public Member Functions

- **TreeFetchingTask_MultiThreaded** (TreeInstance[ ] treeInstances, TreePrototype[ ] treePrototypes, Terrain↩ Data tData, bool isRunning, System.Action OnFinish)

Public Attributes

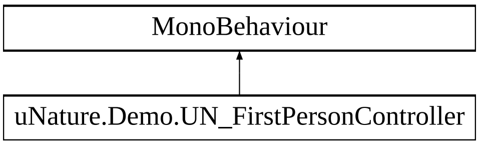
- TreeInstance[ ] **treeInstances**
- TreePrototype[ ] **treePrototypes**
- TerrainData **tData**
- bool **isRunning**
- System.Action **OnFinish**

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/UNTerrainSector.cs

5.59 uNature.Demo.UN_FirstPersonController Class Reference

Inheritance diagram for uNature.Demo.UN_FirstPersonController:



Public Attributes

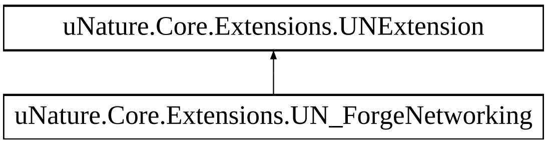
- bool **getInputsMouse** = true

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

- D:/Projects/uNature/Assets/uNature/Scripts/Demo/UN_FirstPersonController.cs

5.60 uNature.Core.Extensions.UN_ForgeNetworking Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_ForgeNetworking:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_ForgeNetworking

Public Member Functions

- void **CreateManager** ()

Properties

- override string **AssetName** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetNameSpace** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **PublisherName** `[get]`
- override string **AssetLogoName** `[get]`
- override string **AssetDocumentationName** `[get]`

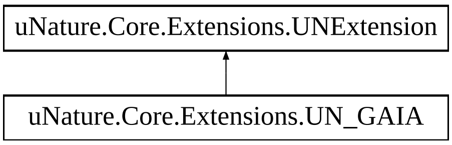
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/ForgeNetworking/UN_ForgeNetworking.cs

5.61 uNature.Core.Extensions.UN_GAIA Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_GAIA:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_GAIA

Properties

- override string **AssetName** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetNameSpace** `[get]`
- override bool **IsDefault** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **PublisherName** `[get]`
- override string **AssetLogoName** `[get]`
- override string **AssetDocumentationName** `[get]`

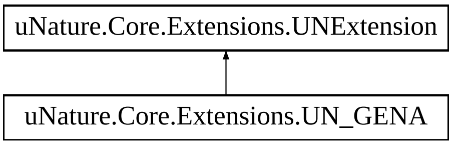
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/GAIA/UN_GAIA.cs

5.62 uNature.Core.Extensions.UN_GENA Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_GENA:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_GENA

Properties

- override string **AssetName** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetNameSpace** `[get]`
- override bool **IsDefault** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **PublisherName** `[get]`
- override string **AssetLogoName** `[get]`
- override string **AssetDocumentationName** `[get]`

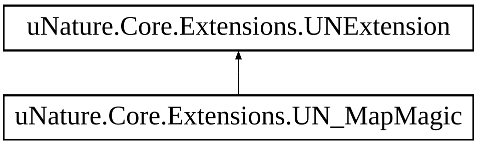
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/GENA/UN_GENA.cs

5.63 uNature.Core.Extensions.UN_MapMagic Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_MapMagic:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_MapMagic

Properties

- override string **AssetName** `[get]`
- override string **AssetNameSpace** `[get]`
- override string **PublisherName** `[get]`
- override bool **Featured** `[get]`
- override string **AssetDocumentationName** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **AssetLogoName** `[get]`

Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/MapMagic/UN_MapMagic.cs

5.64 uNature.Demo.UN_MouseLook Class Reference

Public Member Functions

- void **Init** (Transform character, Transform camera)
- void **LookRotation** (Transform character, Transform camera)

Public Attributes

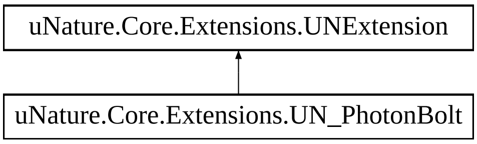
- float **XSensitivity** = 2f
- float **YSensitivity** = 2f
- bool **clampVerticalRotation** = true
- float **MinimumX** = -90F
- float **MaximumX** = 90F
- bool **smooth**
- float **smoothTime** = 5f

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

- D:/Projects/uNature/Assets/uNature/Scripts/Demo/UN_MouseLook.cs

5.65 uNature.Core.Extensions.UN_PhotonBolt Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_PhotonBolt:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_PhotonBolt

Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override string **AssetDocumentationName** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAdress** [get]
- override string **AssetLogoName** [get]

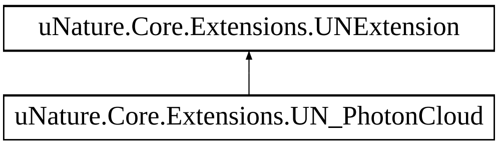
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/PhotonBolt/UN_PhotonBolt.cs

5.66 uNature.Core.Extensions.UN_PhotonCloud Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_PhotonCloud:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_PhotonCloud

Public Member Functions

- void **CreateManager** ()

Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override string **AssetDocumentationName** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAdress** [get]
- override string **AssetLogoName** [get]

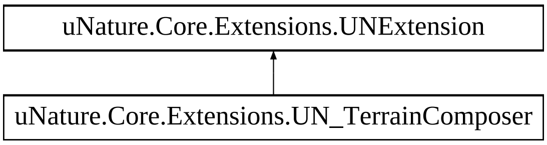
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/PhotonCloud/UN_PhotonCloud.cs

5.67 uNature.Core.Extensions.UN_TerrainComposer Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_TerrainComposer:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_TerrainComposer

Properties

- override string **AssetName** `[get]`
- override string **AssetNameSpace** `[get]`
- override string **PublisherName** `[get]`
- override bool **Featured** `[get]`
- override bool **IsDefault** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **AssetLogoName** `[get]`

Additional Inherited Members

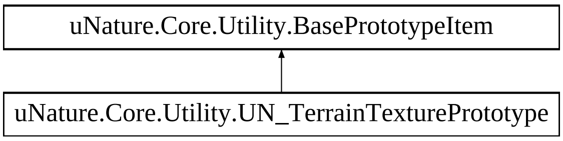
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/TC2/UN_TerrainComposer.cs

5.68 uNature.Core.Utility.UN_TerrainTexturePrototype Class Reference

The prototype of the terrain texture. [Used for listing the paintable surfaces]

Inheritance diagram for uNature.Core.Utility.UN_TerrainTexturePrototype:


uNature.Core.Utility.BasePrototypeItem
uNature.Core.Utility.UN_TerrainTexturePrototype

Public Member Functions

- **UN_TerrainTexturePrototype** (Texture2D splatTexture)

Public Attributes

- Texture2D **splatTexture**

Protected Member Functions

- override Texture2D GetPreview ()

  Get a preview of the splat texture.

Additional Inherited Members

5.68.1 Detailed Description

The prototype of the terrain texture. [Used for listing the paintable surfaces]

5.68.2 Member Function Documentation

5.68.2.1 override Texture2D uNature.Core.Utility.UN_TerrainTexturePrototype.GetPreview ( ) [protected], [virtual]

Get a preview of the splat texture.

Returns

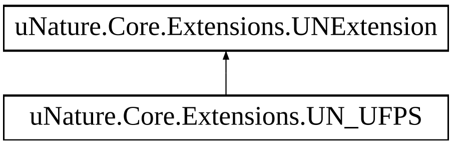
Reimplemented from uNature.Core.Utility.BasePrototypeItem.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNBrushUtility.cs

5.69 uNature.Core.Extensions.UN_UFPS Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_UFPS:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_UFPS

Public Member Functions

- void **ApplyOnCurrentPool** ()

Properties

- override string **AssetName** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetLogoName** `[get]`
- override string **AssetNameSpace** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **PublisherName** `[get]`
- override string **AssetDocumentationName** `[get]`

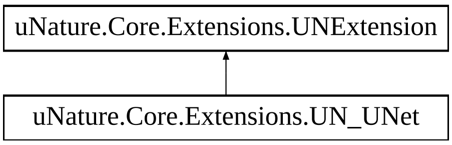
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/UFPS/UN_UFPS.cs

5.70 uNature.Core.Extensions.UN_UNet Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_UNet:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_UNet

Public Member Functions

- void **CreateManager** ()

Properties

- override string **AssetName** `[get]`
- override string **AssetNameSpace** `[get]`
- override string **PublisherName** `[get]`
- override string **AssetDocumentationName** `[get]`

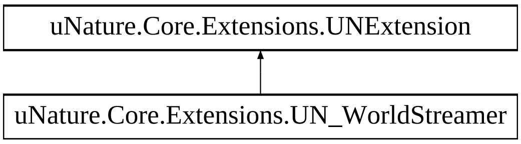
Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/UNet/UN_UNet.cs

5.71 uNature.Core.Extensions.UN_WorldStreamer Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_WorldStreamer:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_WorldStreamer

Properties

- override string **AssetName** `[get]`
- override string **AssetNameSpace** `[get]`
- override string **PublisherName** `[get]`
- override bool **Featured** `[get]`
- override string **AssetDocumentationName** `[get]`
- override string **AssetDescription** `[get]`
- override string **AssetStoreAdress** `[get]`
- override string **AssetLogoName** `[get]`

Additional Inherited Members

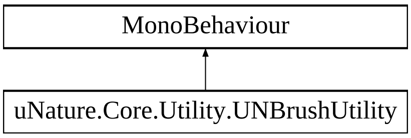
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/WorldStreamer/UN_WorldStreamer.cs

5.72 uNature.Core.Utility.UNBrushUtility Class Reference

Using this class you can paint an brush on the scene.

Inheritance diagram for uNature.Core.Utility.UNBrushUtility:



Public Member Functions

- void DrawBrush (Texture2D brushTexture, Color brushColor, Vector3 originPosition, Quaternion originRotation, float brushSize)

    Draw a brush on the scene.

Static Public Member Functions

- static Texture2D Resize (Texture2D source, int newWidth, int newHeight)

  Resize texture by Justin Markwell and Smoke.
- static float CheckSplatPaint (RaycastHit hit, Vector3 worldPosition, List< UN_TerrainTexturePrototype > chosenSplats)

  Checks if the splats are in that specific position.

Properties

- static UNBrushUtility **instance** `[get]`
- static Projector **projector** `[get]`
- List< UN_TerrainTexturePrototype > **splatPrototypes** `[get]`

5.72.1 Detailed Description

Using this class you can paint an brush on the scene.

5.72.2 Member Function Documentation

5.72.2.1 static float uNature.Core.Utility.UNBrushUtility.CheckSplatPaint (RaycastHit *hit,* Vector3 *worldPosition,* List< UN_TerrainTexturePrototype > *chosenSplats*) `[static]`

Checks if the splats are in that specific position.

Parameters

hit	
worldPosition	

Returns

5.72.2.2 void uNature.Core.Utility.UNBrushUtility.DrawBrush (Texture2D *brushTexture,* Color *brushColor,* Vector3 *originPosition,* Quaternion *originRotation,* float *brushSize*)

Draw a brush on the scene.

Parameters

brushTexture	The brush's texture.
brushColor	The brush's color.
position	The brush's origin position (for example the camera's position).
rotation	The brush's origin rotation (for example the camera's rotation).
brushSize	The brush's size. (Varies from 1 -> 100)

5.72.2.3 static Texture2D uNature.Core.Utility.UNBrushUtility.Resize (Texture2D *source,* int *newWidth,* int *newHeight*) `[static]`

Resize texture by Justin Markwell and Smoke.

Parameters

source	
newWidth	
newHeight	

Returns

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNBrushUtility.cs

5.73 uNature.Core.Utility.UNCombineInstance Struct Reference

Public Member Functions

- **UNCombineInstance** (Matrix4x4 transform, Mesh mesh, float spread, int density, int id)

Public Attributes

- Matrix4x4 **transform**
- Mesh **mesh**
- Vector2 **densityOffset**
- int **density**

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNBatchUtility.cs

5.74 uNature.Core.Utility.UNDictionary$<$ T, T1 $>$ Class Template Reference

Public Member Functions

- void **Add** (T key, T1 value)
- void **RemoveAt** (int index)
- void **Remove** (T key)
- int **TryGetKeyIndex** (T key)

Properties

- List< T > **Keys** `[get]`
- List< T1 > **Values** `[get]`
- int **Count** `[get]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNDictionary.cs

5.75 uNature.Core.Collections.UNDimensionalList< T > Class Template Reference

A 2 dimensional list which is used by certain mechanics in uNature.

Public Member Functions

- bool ContainsKey (int key)

  Checks if the list contains a certain key.
- bool ContainsValue (T value)

  Does the two dimensional list contain this value?
- void TryAddKey (List< T > value)

  Try to add a key.

Properties

- List< T > this[int index] `[get, set]`

  Get the stashed list.
- int Count `[get]`

  Count of the two dimensional list elements.

5.75.1 Detailed Description

A 2 dimensional list which is used by certain mechanics in uNature.

5.75.2 Member Function Documentation

5.75.2.1 bool **uNature.Core.Collections.UNDimensionalList< T >**.ContainsKey (int *key*)

Checks if the list contains a certain key.

5.75.2.2 bool **uNature.Core.Collections.UNDimensionalList< T >**.ContainsValue (T *value*)

Does the two dimensional list contain this value?

Parameters

value	the value

Returns

is it contained ?

5.75.2.3 void **uNature.Core.Collections.UNDimensionalList**< T >.TryAddKey (List< T > *value*)

Try to add a key.

Parameters

value	the value

5.75.3 Property Documentation

5.75.3.1 int **uNature.Core.Collections.UNDimensionalList**< T >.Count `[get]`

Count of the two dimensional list elements.

5.75.3.2 List<T> **uNature.Core.Collections.UNDimensionalList**< T >.this[int index] `[get]`, `[set]`

Get the stashed list.

Parameters

index	index

Returns

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Collections/UNDimensionalList.cs

5.76 uNature.Core.Editor.Helpers.UNEditorHelpers Class Reference

Static Public Member Functions

- static void **SetupSceneTerrains** ()

- static void **FixCorruptedTreeInstanceOnSceneTerrains** ()
- static void **CopySelectedTerrains** ()
- static void **UpdateSelectedTerrains** ()
- static void **ShowDebugWindow** ()

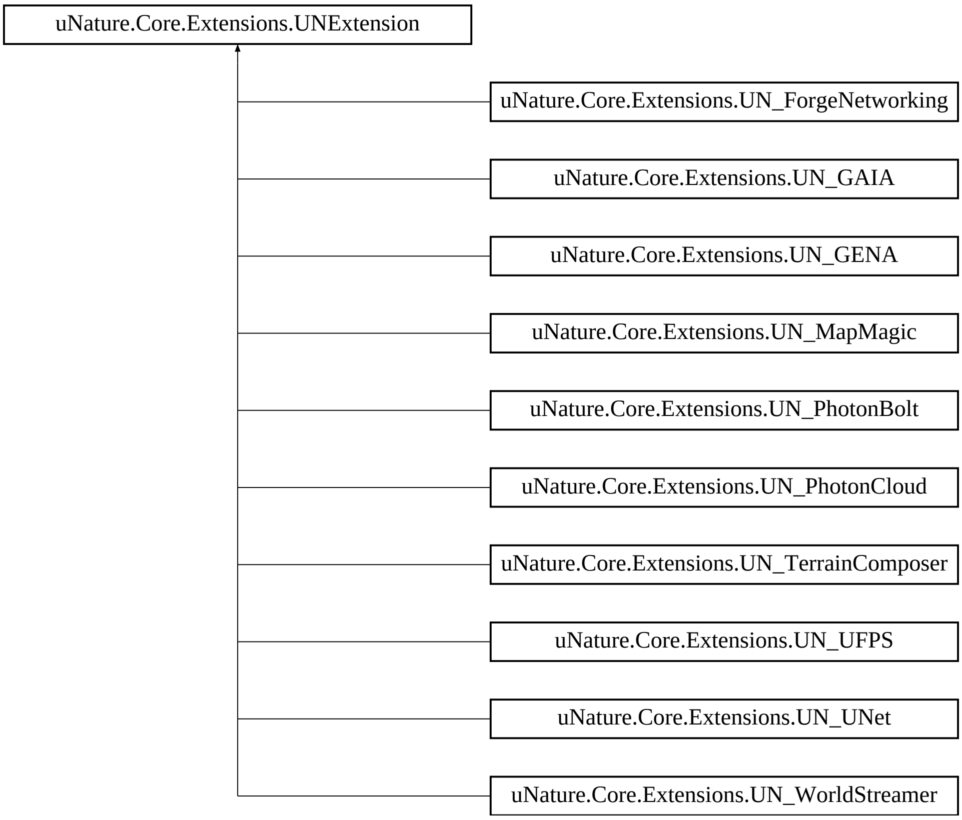
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNEditorHelpers.cs

5.77 uNature.Core.Extensions.UNExtension Class Reference

A uConstruct extension that will allow other 3d party systems to work with uConstruct.

Inheritance diagram for uNature.Core.Extensions.UNExtension:


uNature.Core.Extensions.UNExtension
uNature.Core.Extensions.UN_ForgeNetworking
uNature.Core.Extensions.UN_GAIA
uNature.Core.Extensions.UN_GENA
uNature.Core.Extensions.UN_MapMagic
uNature.Core.Extensions.UN_PhotonBolt
uNature.Core.Extensions.UN_PhotonCloud
uNature.Core.Extensions.UN_TerrainComposer
uNature.Core.Extensions.UN_UFPS
uNature.Core.Extensions.UN_UNet
uNature.Core.Extensions.UN_WorldStreamer

Static Public Member Functions

- static void OpenDocs (UNExtension instance)

    Open the documentation of the extension

Parameters

instance	*Extension instance*

- static void OpenAssetStore (UNExtension instance)

    Open the asset store page of the extension

Parameters

instance	*Extension instance*

- static Texture GetLogo (UNExtension instance)

    Get the extension logo
- static void LoadMethods (UNExtension instance, Type type)

    Load helper methods from an instance.

Public Attributes

- bool isViewed

    Is the extension viewed?
- List< MethodInfo > HelperMethods

    Loaded methods that are created to give tools to people who activated the extension.

Properties

- virtual string AssetName `[get]`

    The asset name (for example TreesManagerSystem).
- virtual string AssetDescription `[get]`

    The asset description (for example :
- virtual bool Featured `[get]`

    Is this asset featured?
- virtual string AssetLogoName `[get]`

    Asset logo name that will be searched on the project.
- virtual string PublisherName `[get]`

    The asset publisher name (for example EEProductions).
- virtual string AssetDocumentationName `[get]`

    Asset extension documentation name.
- virtual string AssetStoreAdress `[get]`

    Asset extension asset store adress - (For exmaple - `https://www.assetstore.unity3d.com/en/#!/content/43129`*).*
- virtual string AssetNameSpace `[get]`

    The namespace that will be added to the defines when the extension is activated.
- virtual bool IsDefault `[get]`

    Default means that this asset doesnt require it to be enabled, that means that its working with uConstruct out of the box.
- bool isActivated `[get, set]`

    Is the extension activated currently?

5.77.1 Detailed Description

A uConstruct extension that will allow other 3d party systems to work with uConstruct.

5.77.2 Member Function Documentation

5.77.2.1 static Texture uNature.Core.Extensions.UNExtension.GetLogo (UNExtension *instance*) `[static]`

Get the extension logo

Parameters

instance	Extension instance

5.77.2.2 static void uNature.Core.Extensions.UNExtension.LoadMethods (UNExtension *instance,* Type *type*) [static]

Load helper methods from an instance.

Parameters

instance	Extension instance

5.77.2.3 static void uNature.Core.Extensions.UNExtension.OpenAssetStore (UNExtension *instance*) [static]

Open the asset store page of the extension

Parameters

instance	Extension instance

5.77.2.4 static void uNature.Core.Extensions.UNExtension.OpenDocs (UNExtension *instance*) [static]

Open the documentation of the extension

Parameters

instance	Extension instance

5.77.3 Member Data Documentation

5.77.3.1 List<MethodInfo> uNature.Core.Extensions.UNExtension.HelperMethods

Loaded methods that are created to give tools to people who activated the extension.

5.77.3.2 bool uNature.Core.Extensions.UNExtension.isViewed

Is the extension viewed?

5.77.4 Property Documentation

5.77.4.1 virtual string uNature.Core.Extensions.UNExtension.AssetDescription [get]

The asset description (for example :

An asset used for optimizing terrain & game world. Features : ....

5.77.4.2 virtual string uNature.Core.Extensions.UNExtension.AssetDocumentationName [get]

Asset extension documentation name.

5.77.4.3 virtual string uNature.Core.Extensions.UNExtension.AssetLogoName [get]

Asset logo name that will be searched on the project.

For example: uConstructLogo

5.77.4.4 virtual string uNature.Core.Extensions.UNExtension.AssetName [get]

The asset name (for example TreesManagerSystem).

5.77.4.5 virtual string uNature.Core.Extensions.UNExtension.AssetNameSpace [get]

The namespace that will be added to the defines when the extension is activated.

5.77.4.6 virtual string uNature.Core.Extensions.UNExtension.AssetStoreAdress [get]

Asset extension asset store adress - (For exmaple - https://www.assetstore.unity3d.com/en/#!/content/43129).

5.77.4.7 virtual bool uNature.Core.Extensions.UNExtension.Featured [get]

Is this asset featured?

5.77.4.8 bool uNature.Core.Extensions.UNExtension.isActivated [get], [set]

Is the extension activated currently?

5.77.4.9 virtual bool uNature.Core.Extensions.UNExtension.IsDefault [get]

Default means that this asset doesnt require it to be enabled, that means that its working with uConstruct out of the box.

5.77.4.10 virtual string uNature.Core.Extensions.UNExtension.PublisherName [get]

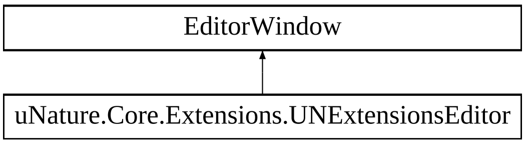
The asset publisher name (for example EEProductions).

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Base/UNExtension.cs

5.78 uNature.Core.Extensions.UNExtensionsEditor Class Reference

Inheritance diagram for uNature.Core.Extensions.UNExtensionsEditor:



Static Public Member Functions

- static void **Open** ()
- static void **HandleCompile** ()

Public Attributes

- const string **UN_DEFINE** = "UN_INSTALLED"

Properties

- static Texture2D **featuredIcon** `[get]`
- static GUIStyle **featuredFoldoutStyle** `[get]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Editor/UNExtensionsEditor.cs

5.79 uNature.Core.Utility.UNFastList< T > Class Template Reference

A list that requires you to set

Public Member Functions

- **UNFastList** (int maxCapacity)
- void **Add** (T item)
- void **Clear** ()

Properties

- int **Count** `[get]`

5.79.1 Detailed Description

A list that requires you to set

Template Parameters

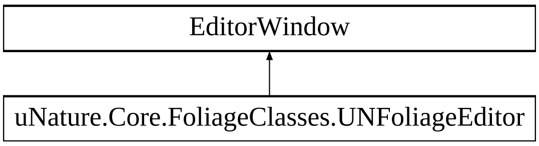
T	

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNFastList.cs

5.80 uNature.Core.FoliageClasses.UNFoliageEditor Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.UNFoliageEditor:



Public Member Functions

- void **OnGUI** ()

Static Public Member Functions

- static void **OpenWindow** ()

Properties

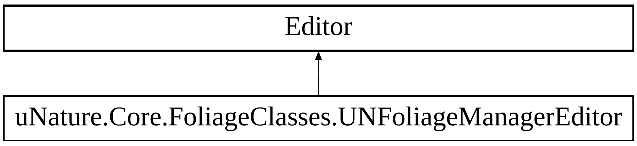
- FoliagePrototype **currentPrototype** `[get, set]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNFoliageEditor.cs

5.81 uNature.Core.FoliageClasses.UNFoliageManagerEditor Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.UNFoliageManagerEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

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

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNFoliageManagerEditor.cs

5.82 uNature.Core.Collections.UNList< T > Class Template Reference

A custom list which is used on some important interfaces in UN.

Public Member Functions

- void Add (T item)

  Add an item to the list.
- void Remove (T item)

  Remove an item from the list.
- T TryGet (System.Object similarItem)

  Get a similar instance by a custom Equals which needs to be initialized on the item.
- bool Contains (System.Object item)

  Is this item contained in the list?

Properties

- int Count `[get]`

  Get list count
- T this[int index] `[get, set]`

  Get an element from the list.

5.82.1 Detailed Description

A custom list which is used on some important interfaces in UN.

Template Parameters

T	the list type.

5.82.2 Member Function Documentation

5.82.2.1 void uNature.Core.Collections.UNList< T >.Add (T *item*)

Add an item to the list.

Parameters

item	

5.82.2.2 bool uNature.Core.Collections.UNList< T >.Contains (System.Object *item*)

Is this item contained in the list?

Parameters

item	

Returns

5.82.2.3 void uNature.Core.Collections.UNList< T >.Remove (T *item*)

Remove an item from the list.

Parameters

item	

5.82.2.4 T uNature.Core.Collections.UNList< T >.TryGet (System.Object *similarItem*)

Get a similar instance by a custom Equals which needs to be initialized on the item.

Parameters

similarItem	

Returns

5.82.3 Property Documentation

5.82.3.1 int uNature.Core.Collections.UNList< T >.Count [get]

Get list count

5.82.3.2 T uNature.Core.Collections.UNList< T >.this[int index] [get], [set]

Get an element from the list.

Parameters

index	

Returns

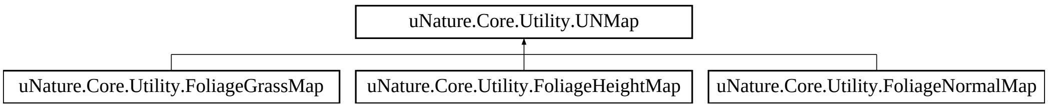
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Collections/UNList.cs

5.83 uNature.Core.Utility.UNMap Class Reference

The abstract Map class.

Inheritance diagram for uNature.Core.Utility.UNMap:


uNature.Core.Utility.UNMap
uNature.Core.Utility.FoliageGrassMap
uNature.Core.Utility.FoliageHeightMap
uNature.Core.Utility.FoliageNormalMap

Public Member Functions

- void **RestoreChanges** ()
- void **Apply** (Color32[ ] pixels)
- void **SetPixels32** ()
- void **SetPixels32** (Color32[ ] pixels)
- void **SetPixelsNoApply** ()
- void **SetPixels32Delayed** ()
- byte[ ] **EncodeToPNG** ()
- void **Resize** (int size)
- void **Clear** (bool autoApply, Color32 defaultColor)

Protected Member Functions

- **UNMap** (Texture2D texture, Color32[ ] pixels, FoliageManagerInstance mInstance)
- virtual void **OnDirty** (bool value)

Protected Attributes

- Texture2D **_map**
- Color32[ ] **originalMapPixels** = null

Properties

- Texture2D **map** `[get, set]`
- Color32[ ] **mapPixels** `[get, set]`
- int **mapWidth** `[get]`
- bool **dirty** `[get, set]`
- FoliageManagerInstance **mInstance** `[get]`

5.83.1 Detailed Description

The abstract Map class.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.84 uNature.Core.FoliageClasses.UNMeshData Class Reference

Public Member Functions

- **UNMeshData** (Mesh mesh)

Public Attributes

- List< Vector3 > **vertices**
- List< Vector3 > **normals**
- List< int > **triangles**
- List< Vector2 > **uv1s**
- int **verticesLength**
- int **normalsLength**
- int **uv1sLength**
- int **trianglesLength**

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

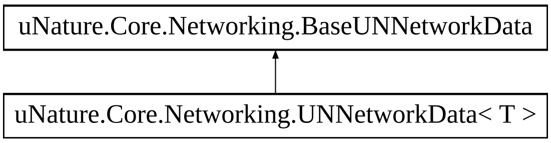
5.85 uNature.Core.Networking.UNNetworkData< T > Class Template Reference

A class which can be used for an abstract networking data.

Template Parameters

T	The network connection type which the networking library uses.

Inheritance diagram for uNature.Core.Networking.UNNetworkData< T >:


uNature.Core.Networking.BaseUNNetworkData
uNature.Core.Networking.UNNetworkData< T >

Public Member Functions

- override void UnPack ()

  Unpack the data
- virtual void SendToServer ()

  Send data to server
- virtual void SendToConnection (T connection)

  Send data to connection

Parameters

connection	*the targeted connection*

- virtual void SendToClients ()

  Send data to clients
- virtual void SendToOthers ()

  Send data to other connections
- override bool Equals (object obj)

  Create equal state which checks whether those 2 instances of NetworkData are equal
- override int GetHashCode ()

  Overrided this method only to get rid of a warning.
- virtual byte[ ] Serialize ()

  Serialize the object

Static Public Member Functions

- static T2 Pack< T1, T2 > (Terrain terrain, int treeInstanceID, int health, PacketType type)

  Pack the data and create a data instance
- static UNNetworkData< T > Deserialize (byte[ ] bytes)

  Deserialize the data.

Additional Inherited Members

5.85.1 Detailed Description

A class which can be used for an abstract networking data.

Template Parameters

T	The network connection type which the networking library uses.

5.85.2 Member Function Documentation

5.85.2.1 static UNNetworkData<T> uNature.Core.Networking.UNNetworkData< T >.Deserialize (byte[ ] *bytes*) [static]

Deserialize the data.

Parameters

bytes	the data.

Returns

the deserialized object.

5.85.2.2 override bool uNature.Core.Networking.UNNetworkData< T >.Equals (object *obj*)

Create equal state which checks whether those 2 instances of NetworkData are equal

Parameters

obj	

Returns

5.85.2.3 override int uNature.Core.Networking.UNNetworkData< T >.GetHashCode ( )

Overrided this method only to get rid of a warning.

Returns

5.85.2.4 static T2 uNature.Core.Networking.UNNetworkData< T >.Pack< T1, T2 > (Terrain *terrain,* int *treeInstanceID,* int *health,* PacketType *type*) [static]

Pack the data and create a data instance

Type Constraints

T2* : *UNNetworkData<T1>

5.85.2.5 virtual void uNature.Core.Networking.UNNetworkData< T >.SendToClients ( ) [virtual]

Send data to clients

5.85.2.6 virtual void uNature.Core.Networking.UNNetworkData< T >.SendToConnection (T *connection*) [virtual]

Send data to connection

Parameters

connection	the targeted connection

5.85.2.7 virtual void **uNature.Core.Networking.UNNetworkData< T >.SendToOthers** ( ) [virtual]

Send data to other connections

5.85.2.8 virtual void **uNature.Core.Networking.UNNetworkData< T >.SendToServer** ( ) [virtual]

Send data to server

5.85.2.9 virtual byte [ ] **uNature.Core.Networking.UNNetworkData< T >.Serialize** ( ) [virtual]

Serialize the object

Returns

serialized bytes

5.85.2.10 override void **uNature.Core.Networking.UNNetworkData< T >.UnPack** ( ) [virtual]

Unpack the data

Reimplemented from uNature.Core.Networking.BaseUNNetworkData.

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/UNNetworkData.cs

5.86 uNature.Core.Networking.UNNetworkManager< T1, T2 > Class Template Reference

A template for networking, which can be used by networking extensions to easily get the networking actions done.

Template Parameters

T1	the targeted networking connection
T2	the type of the data

Public Member Functions

- UNNetworkManager (MonoBehaviour managerInstance)

The constructor of this class, initiate basic events.

- virtual void Awake ()

  Called when the network manager is initialized.
- virtual void UpdatePermissions ()

  Reupdate the permissions for the Pool items. (needs to be called when ever there's a networking change for the owner/controller)
- virtual void SendEvent (UNNetworkData< T1 > instance)

  Send event to the correct location
- virtual void SendToConnection (T1 connection, UNNetworkData< T1 > instance)

  Send to certain connection the data
- virtual void SendToClients (UNNetworkData< T1 > instance)

  Send to all of the clients.
- virtual void SendToOthers (UNNetworkData< T1 > instance)

  Send to all other connections.
- virtual void SendToServer (UNNetworkData< T1 > instance)

  Send to the server the data.
- void OnClientConnected (T1 conn)

  Called when the client connects, send all data.

Public Attributes

- const float **STREAM_UPDATE_CHECK_INTERVAL_SECONDS** = 2

Static Public Attributes

- static UNNetworkManager< T1, T2 > manager

  a static instance of this object.

Protected Member Functions

- IEnumerator CheckForStreamingBufferedUpdates ()

  This method is checking every certain amount of seconds for new loaded streamed areas to update data that is waiting for a streamed terrain to be loaded.
- void OnHarvestableTreeInstancePooled (HarvestableTIPoolItem instance)

  Called when an harvestable item instance has been created
- void OnItemDamaged (HarvestableTIPoolItem item, int damage)

  Update item damage and handle synchorization.

Properties

- static UNList< BaseUNNetworkData > bufferedData `[get]`

  The buffered data which will be sent to all of the connecting connections.
- virtual bool isServer `[get]`

  Are we the server?
- virtual bool isAuth `[get]`

  is the server architecture is authoritative?

5.86.1 Detailed Description

A template for networking, which can be used by networking extensions to easily get the networking actions done.

Template Parameters

T1	the targeted networking connection
T2	the type of the data

Type Constraints

***T2* : *UNNetworkData*<*T1*>**

5.86.2 Constructor & Destructor Documentation

5.86.2.1 uNature.Core.Networking.UNNetworkManager< T1, T2 >.UNNetworkManager (MonoBehaviour *managerInstance*)

The constructor of this class, initiate basic events.

5.86.3 Member Function Documentation

5.86.3.1 virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.Awake ( ) [virtual]

Called when the network manager is initialized.

5.86.3.2 IEnumerator uNature.Core.Networking.UNNetworkManager< T1, T2 >.CheckForStreamingBufferedUpdates ( ) [protected]

This method is checking every certain amount of seconds for new loaded streamed areas to update data that is waiting for a streamed terrain to be loaded.

Returns

5.86.3.3 void uNature.Core.Networking.UNNetworkManager< T1, T2 >.OnClientConnected (T1 *conn*)

Called when the client connects, send all data.

Parameters

conn	the connection

5.86.3.4 void uNature.Core.Networking.UNNetworkManager< T1, T2 >.OnHarvestableTreeInstancePooled (HarvestableTIPoolItem *instance*) [protected]

Called when an harvestable item instance has been created

Parameters

instance	the created instance

5.86.3.5 void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.OnItemDamaged (**HarvestableTIPoolItem** *item,* int *damage*) [protected]

Update item damage and handle synchorization.

Parameters

item	
health	

5.86.3.6 virtual void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.SendEvent (**UNNetworkData**< T1 > *instance*) [virtual]

Send event to the correct location

Parameters

instance	the data instance

5.86.3.7 virtual void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.SendToClients (**UNNetworkData**< T1 > *instance*) [virtual]

Send to all of the clients.

Parameters

terrainName	the terrain name (terrain.name)
instanceID	the tree instance
destroy	you want to destroy/ restore the tree?

5.86.3.8 virtual void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.SendToConnection (T1 *connection,* **UNNetworkData**< T1 > *instance*) [virtual]

Send to certain connection the data

Parameters

connection	the connection
terrainName	the terrain name (terrain.name)
instanceID	the tree instance
destroy	you want to destroy/ restore the tree?

5.86.3.9 virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.SendToOthers (UNNetworkData< T1 > *instance*) [virtual]

Send to all other connections.

Parameters

terrainName	the terrain name (terrain.name)
instanceID	the tree instance
destroy	you want to destroy/ restore the tree?

5.86.3.10 virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.SendToServer (UNNetworkData< T1 > *instance*) [virtual]

Send to the server the data.

Parameters

terrainName	the terrain name (terrain.name)
instanceID	the tree instance
destroy	you want to destroy/ restore the tree?

5.86.3.11 virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.UpdatePermissions ( ) [virtual]

Reupdate the permissions for the Pool items. (needs to be called when ever there's a networking change for the owner/controller)

5.86.4 Member Data Documentation

5.86.4.1 UNNetworkManager<T1, T2> uNature.Core.Networking.UNNetworkManager< T1, T2 >.manager [static]

a static instance of this object.

5.86.5 Property Documentation

5.86.5.1 UNList<BaseUNNetworkData> uNature.Core.Networking.UNNetworkManager< T1, T2 >.bufferedData [static], [get]

The buffered data which will be sent to all of the connecting connections.

5.86.5.2 virtual bool uNature.Core.Networking.UNNetworkManager< T1, T2 >.isAuth [get]

is the server architecture is authoritative?

5.86.5.3 virtual bool uNature.Core.Networking.UNNetworkManager< T1, T2 >.isServer [get]

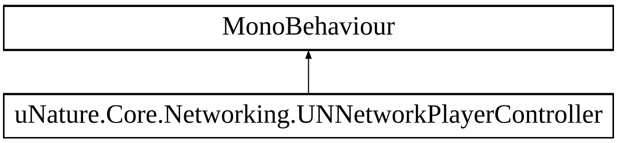
Are we the server?

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/UNNetworkManager.cs

5.87 uNature.Core.Networking.UNNetworkPlayerController Class Reference

Inheritance diagram for uNature.Core.Networking.UNNetworkPlayerController:



Public Member Functions

- virtual void **OnAttached** ()
- void **ManageEnableOnProxies** (bool value)

Public Attributes

- MonoBehaviour[ ] **disableOnProxies**
- Camera **Camera**
- CharacterController **controller**

Protected Member Functions

- virtual void **Awake** ()

Properties

- virtual bool **hasControl** [get]

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/UNNetworkPlayerController.cs

5.88 uNature.Core.UNPhysicsHit_Grass Struct Reference

A class that holds the data for the hit data

Public Attributes

- Vector3 **point**
- float **distance**

5.88.1 Detailed Description

A class that holds the data for the hit data

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysics.cs

5.89 uNature.Core.UNPhysicsHitsArray Class Reference

An custom array that holds all ray results in an array

Public Member Functions

- void **AddToList** (UNPhysicsHit_Grass hit)
- void **Sort** ()

Properties

- UNPhysicsHit_Grass **this[int index]** `[get]`
- int **Count** `[get]`

5.89.1 Detailed Description

An custom array that holds all ray results in an array

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysics.cs

5.90 uNature.Core.UNPhysicsObject Struct Reference

This is a base class for a UCPhysicsObject. Every class that inherites this class will be counted in the physics system.

Public Member Functions

- void UpdateBounds ()

  Update object's bounds
- void OnDrawGizmos ()

  Draw gizmos
- void DrawShape (Matrix4x4 matrix)

  Draw the shape of the bounds
- bool Raycast (Ray ray, out UNPhysicsHit_Grass _hit, LayerMask mask)

  Raycast the physics object

Static Public Attributes

- static UNPhysicsHit_Grass **hit** = new UNPhysicsHit_Grass()

Properties

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

5.90.1 Detailed Description

This is a base class for a UCPhysicsObject. Every class that inherites this class will be counted in the physics system.

5.90.2 Member Function Documentation

5.90.2.1 void uNature.Core.UNPhysicsObject.DrawShape (Matrix4x4 *matrix*)

Draw the shape of the bounds

Parameters

matrix	the matrix of the bounds
selected	is the shape selected in heirachy

5.90.2.2 void uNature.Core.UNPhysicsObject.OnDrawGizmos ( )

Draw gizmos

5.90.2.3 bool uNature.Core.UNPhysicsObject.Raycast (Ray *ray,* out UNPhysicsHit_Grass *_hit,* LayerMask *mask*)

Raycast the physics object

Parameters

origin	ray origin
direction	ray direction
_hit	hit data
distance	max distance
mask	layerMask

Returns

Did we hit something?

5.90.2.4 void uNature.Core.UNPhysicsObject.UpdateBounds ( )

Update object's bounds

Parameters

center	The center of the bounds, worldspace
size	The size of the bounds, worldspace

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysicsObject.cs

5.91 uNature.Core.UNPhysicsTemplate Struct Reference

Public Member Functions

- **UNPhysicsTemplate** (Vector3 position, float spreadX, float spreadZ, int densityIndex, FoliagePrototype prototype)

Public Attributes

- Vector3 **position**
- Vector2 **spread**
- int **densityIndex**
- FoliagePrototype **prototype**

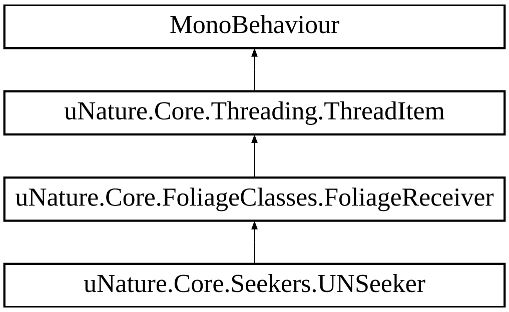
The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysicsTemplate.cs

5.92 uNature.Core.Seekers.UNSeeker Class Reference

Seekers are basically GameObjects in the scene which should interact with the objects in the game.

Inheritance diagram for uNature.Core.Seekers.UNSeeker:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.FoliageReceiver
uNature.Core.Seekers.UNSeeker

Public Member Functions

- virtual IEnumerator Start ()

  Called on start, initiate initial check targets.

Public Attributes

- float seekingDistance = 20f

  How far will it look ?
- bool attackTrees = true

  Disable this if you want to do your own trees logic.
- int raycastMask = 1

  Ignore layer for tree attack.
- float raycastDistance = 10

  Raycast range for tree attack.

Protected Member Functions

- override void Update ()

  Check for movement.
- virtual void CheckTargetsOnMove ()

  Checks the targets when the character moved enoughed.
- virtual void HarvestChecks ()

  Try to harvest trees.

Protected Attributes

- Vector3 lastMovement = Vector3.zero

  What was the last position our AOI was updated on?
- float _treesCheckDistance = 10f

  After how much distance will it update the trees.

Properties

- float **treesCheckDistance** `[get, set]`

Additional Inherited Members

5.92.1 Detailed Description

Seekers are basically GameObjects in the scene which should interact with the objects in the game.

5.92.2 Member Function Documentation

5.92.2.1 virtual void uNature.Core.Seekers.UNSeeker.CheckTargetsOnMove ( ) `[protected]`, `[virtual]`

Checks the targets when the character moved enoughed.

5.92.2.2 virtual void uNature.Core.Seekers.UNSeeker.HarvestChecks ( ) `[protected]`, `[virtual]`

Try to harvest trees.

5.92.2.3 virtual IEnumerator uNature.Core.Seekers.UNSeeker.Start ( ) `[virtual]`

Called on start, initiate initial check targets.

5.92.2.4 override void uNature.Core.Seekers.UNSeeker.Update ( ) `[protected]`, `[virtual]`

Check for movement.

Reimplemented from uNature.Core.FoliageClasses.FoliageReceiver.

5.92.3 Member Data Documentation

5.92.3.1 float uNature.Core.Seekers.UNSeeker._treesCheckDistance = 10f `[protected]`

After how much distance will it update the trees.

5.92.3.2 bool uNature.Core.Seekers.UNSeeker.attackTrees = true

Disable this if you want to do your own trees logic.

5.92.3.3 Vector3 uNature.Core.Seekers.UNSeeker.lastMovement = Vector3.zero [protected]

What was the last position our AOI was updated on?

5.92.3.4 float uNature.Core.Seekers.UNSeeker.raycastDistance = 10

Raycast range for tree attack.

5.92.3.5 int uNature.Core.Seekers.UNSeeker.raycastMask = 1

Ignore layer for tree attack.

5.92.3.6 float uNature.Core.Seekers.UNSeeker.seekingDistance = 20f

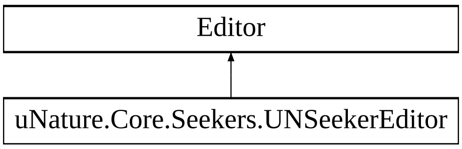
How far will it look ?

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Seekers/UNSeeker.cs

5.93 uNature.Core.Seekers.UNSeekerEditor Class Reference

Inheritance diagram for uNature.Core.Seekers.UNSeekerEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

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

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNSeekerEditor.cs

5.94 uNature.Core.Settings.UNSetting Class Reference

A class which should be used on custom classes that needs to be shown and serialized as a setting.

Public Member Functions

- virtual void DrawGUI ()

  Draw the gui of the setting on this method. This will be called from the UNSettingsEditor.

5.94.1 Detailed Description

A class which should be used on custom classes that needs to be shown and serialized as a setting.

5.94.2 Member Function Documentation

5.94.2.1 virtual void uNature.Core.Settings.UNSetting.DrawGUI ( ) [virtual]

Draw the gui of the setting on this method. This will be called from the UNSettingsEditor.

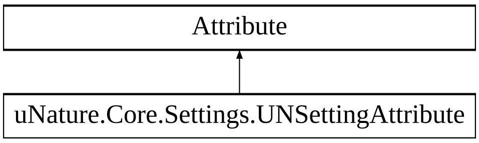
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSetting.cs

5.95 uNature.Core.Settings.UNSettingAttribute Class Reference

The attribute of each setting which handles the drawing of the setting (generically).

Inheritance diagram for uNature.Core.Settings.UNSettingAttribute:



Public Member Functions

- **UNSettingAttribute** (UNSettingCategories category, string name)
- **UNSettingAttribute** (UNSettingCategories category, string name, string desc)
- object **Draw** (object instance)

Public Attributes

- UNSettingCategories **category**
- string **name**
- string **desc**

5.95.1 Detailed Description

The attribute of each setting which handles the drawing of the setting (generically).

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSettings.cs

5.96 uNature.Core.Settings.UNSettingCategory Class Reference

The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.

Public Member Functions

- **UNSettingCategory** (UNSettingCategories category)

Static Public Member Functions

- static UNSettingCategory **GetCategory** (UNSettingCategories category)

Public Attributes

- bool **show**
- UNSettingCategories **type**
- List< UNSettingAttribute > **attributes** = new List<UNSettingAttribute>()
- List< FieldInfo > **fields** = new List<FieldInfo>()

Properties

- static List< UNSettingCategory > **categories** `[get]`

5.96.1 Detailed Description

The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.

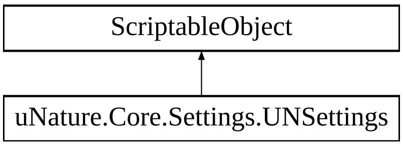
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSettings.cs

5.97 uNature.Core.Settings.UNSettings Class Reference

A class which handles certain settings of aspects in UN

Inheritance diagram for uNature.Core.Settings.UNSettings:



Public Member Functions

- void ResetDefaults ()

  Reset the settings to the default state.

Static Public Member Functions

- static void Log (string context)

  Log a message on the uNature debug mode.

Public Attributes

- const string ProjectVersion = "2.1"

  The version number of this package.
- const string fileName = "UNSettings"

  The file name which will be created for this settings file.
- const string ProjectName = "uNature"

  Project name (UN folder name).
- bool **UN_TreeInstancesRespawnsEnabled** = false
- bool **UN_Networking_Auth** = true
- float **UN_TreeInstancesRespawnsTime** = 1
- bool **UN_Threading_Enabled** = true
- Threading.uNature_Thread_Workers **UN_Threading_WorkersCount** = Threading.uNature_Thread_↩ Workers.One_Worker
- bool **UN_Debugging_Enabled** = false
- bool **UN_Console_Debugging_Enabled** = false
- bool **UN_Foliage_RUNTIME_SAVECHANGES** = false

Properties

- static string ProjectPath [get]

  The found path to the project directory (based on the name provided on ProjectName).
- static UNSettings **instance** [get]

5.97.1 Detailed Description

A class which handles certain settings of aspects in UN

5.97.2 Member Function Documentation

5.97.2.1 static void uNature.Core.Settings.UNSettings.Log (string *context*) [static]

Log a message on the uNature debug mode.

Parameters

context	

5.97.2.2 void uNature.Core.Settings.UNSettings.ResetDefaults ( )

Reset the settings to the default state.

5.97.3 Member Data Documentation

5.97.3.1 const string uNature.Core.Settings.UNSettings.fileName = "UNSettings"

The file name which will be created for this settings file.

5.97.3.2 const string uNature.Core.Settings.UNSettings.ProjectName = "uNature"

Project name (UN folder name).

5.97.3.3 const string uNature.Core.Settings.UNSettings.ProjectVersion = "2.1"

The version number of this package.

5.97.4 Property Documentation

5.97.4.1 string uNature.Core.Settings.UNSettings.ProjectPath `[static], [get]`

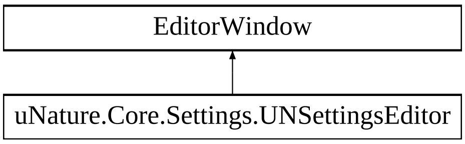
The found path to the project directory (based on the name provided on ProjectName).

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSettings.cs

5.98 uNature.Core.Settings.UNSettingsEditor Class Reference

Inheritance diagram for uNature.Core.Settings.UNSettingsEditor:



Static Public Member Functions

- static void **Open** ()

Public Attributes

- UNSettings **_settings**

Properties

- UNSettings **settings** `[get]`

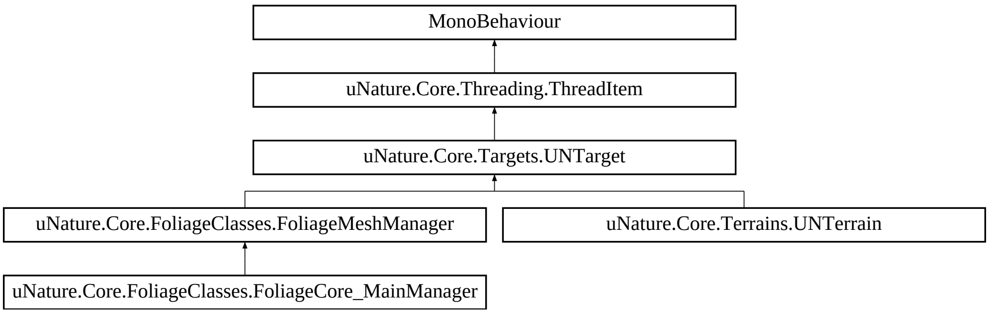
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/Editor/UNSettingsEditor.cs

5.99 uNature.Core.Targets.UNTarget Class Reference

A target is what will be taken into account with the system. For example terrains.

Inheritance diagram for uNature.Core.Targets.UNTarget:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Targets.UNTarget
uNature.Core.FoliageClasses.FoliageMeshManager
uNature.Core.Terrains.UNTerrain
uNature.Core.FoliageClasses.FoliageCore_MainManager

Public Member Functions

- virtual void Check (Seekers.UNSeeker seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)

  Check and apply AOI from seeker.

Parameters

seeker	*Our seeker*
seekerPos	*the seeker position -> in order to maintain multithreading.*

- virtual bool InDistance (UNSeeker seeker)

  Confirm that a seeker is in the range of the target.
- virtual Vector3 FixPosition (Vector3 position)

  Fix the position that is given to the local space position of this target - for example in the terrain you want to reduce the terrain position.

- virtual void OnDrawGizmos ()

  Draw gizmos
- virtual void CreatePool (System.Type PoolItemType)

  Create Pool.

Static Public Member Functions

- static void CheckTargets (UNSeeker seeker, float distance)

  Check and apply aoi from a certain seeekr.

Public Attributes

- Pool Pool

  A Pool which is used to increase performance on runtime, which manages objects smartly than instantiating them manually on runtime each time.
- string PoolTypeSerializedName = ""

  Was the Pool type de-serialized.
- int PoolAmount = 15

  How many objects will be created for each Pool type.

Static Public Attributes

- static List< UNTarget > worldTargets = new List<UNTarget>()

  All of the targets in the scene.

Protected Member Functions

- virtual void Awake ()

  Initiate awake settings.
- override void Update ()

  Called every frame
- override void OnEnable ()

  Add this target to the targets Pool
- override void OnDisable ()

  Remove this target to the targets Pool

Properties

- System.Type **PoolItemType** `[get, set]`
- virtual bool useMultithreadedCheck `[get]`

  Will the system call a multi-threaded task for making the checks ?

5.99.1 Detailed Description

A target is what will be taken into account with the system. For example terrains.

5.99.2 Member Function Documentation

5.99.2.1 virtual void uNature.Core.Targets.UNTarget.Awake ( ) [protected], [virtual]

Initiate awake settings.

Reimplemented in uNature.Core.Terrains.UNTerrain, and uNature.Core.FoliageClasses.FoliageCore_MainManager.

5.99.2.2 virtual void uNature.Core.Targets.UNTarget.Check (Seekers.UNSeeker *seeker*, Vector3 *seekerPos*, float *seekingDistance*, bool *isPlaying*) [virtual]

Check and apply AOI from seeker.

Parameters

seeker	Our seeker
seekerPos	the seeker position -> in order to maintain multithreading.

5.99.2.3 static void uNature.Core.Targets.UNTarget.CheckTargets (UNSeeker *seeker*, float *distance*) [static]

Check and apply aoi from a certain seeekr.

Parameters

seeker	our seeker.
distance	seeking distance

5.99.2.4 virtual void uNature.Core.Targets.UNTarget.CreatePool (System.Type *PoolItemType*) [virtual]

Create Pool.

Reimplemented in uNature.Core.Terrains.UNTerrain.

5.99.2.5 virtual Vector3 uNature.Core.Targets.UNTarget.FixPosition (Vector3 *position*) [virtual]

Fix the position that is given to the local space position of this target - for example in the terrain you want to reduce the terrain position.

Parameters

position	the position

Returns

fixed position

Reimplemented in uNature.Core.Terrains.UNTerrain.

5.99.2.6 virtual bool uNature.Core.Targets.UNTarget.InDistance (UNSeeker *seeker*) [virtual]

Confirm that a seeker is in the range of the target.

Parameters

seeker	The seeker.

Returns

Is the inrange of our target?

5.99.2.7 override void uNature.Core.Targets.UNTarget.OnDisable ( ) [protected], [virtual]

Remove this target to the targets Pool

Reimplemented from uNature.Core.Threading.ThreadItem.

Reimplemented in uNature.Core.Terrains.UNTerrain.

5.99.2.8 virtual void uNature.Core.Targets.UNTarget.OnDrawGizmos ( ) [virtual]

Draw gizmos

5.99.2.9 override void uNature.Core.Targets.UNTarget.OnEnable ( ) [protected], [virtual]

Add this target to the targets Pool

Reimplemented from uNature.Core.Threading.ThreadItem.

Reimplemented in uNature.Core.Terrains.UNTerrain.

5.99.2.10 override void uNature.Core.Targets.UNTarget.Update ( ) [protected], [virtual]

Called every frame

Reimplemented from uNature.Core.Threading.ThreadItem.

5.99.3 Member Data Documentation

5.99.3.1 Pool uNature.Core.Targets.UNTarget.Pool

A Pool which is used to increase performance on runtime, which manages objects smartly than instantiating them manually on runtime each time.

5.99.3.2 int uNature.Core.Targets.UNTarget.PoolAmount = 15

How many objects will be created for each Pool type.

5.99.3.3 string uNature.Core.Targets.UNTarget.PoolTypeSerializedName = ""

Was the Pool type de-serialized.

5.99.3.4 List<UNTarget> uNature.Core.Targets.UNTarget.worldTargets = new List<UNTarget>() [static]

All of the targets in the scene.

5.99.4 Property Documentation

5.99.4.1 virtual bool uNature.Core.Targets.UNTarget.useMultithreadedCheck [get], [protected]

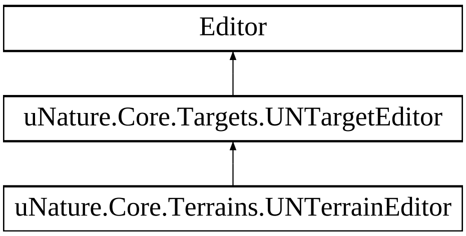
Will the system call a multi-threaded task for making the checks ?

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Targets/UNTarget.cs

5.100 uNature.Core.Targets.UNTargetEditor Class Reference

Inheritance diagram for uNature.Core.Targets.UNTargetEditor:


Editor
uNature.Core.Targets.UNTargetEditor
uNature.Core.Terrains.UNTerrainEditor

Public Member Functions

- override void **OnInspectorGUI** ()

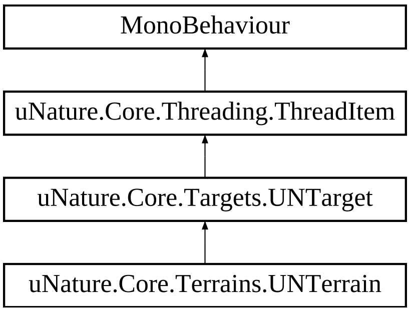
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNTargetEditor.cs

5.101 uNature.Core.Terrains.UNTerrain Class Reference

A class that needs to be on each terrain that needs to be taken into account when managing the system.

Inheritance diagram for uNature.Core.Terrains.UNTerrain:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.Targets.UNTarget
uNature.Core.Terrains.UNTerrain

Public Member Functions

- virtual UNTerrainSector GenerateSector (int sectorResolution, bool multiThread)

  Generate a sector and assign it to the UNTerrain.
- virtual UNTerrainSector GenerateSector (int sectorResolution)

  Generate a sector and assign it to the UNTerrain.
- override void Check (Seekers.UNSeeker seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)

  Check for seeker on terrain.
- override bool InDistance (Seekers.UNSeeker seeker)

  Check if the seeker is in range of the terrain.
- override void CreatePool (System.Type PoolItemType)

  Fill up our Pool.
- override Vector3 FixPosition (Vector3 position)

  Return the position that can be used with the chunks.

Public Attributes

- const float removedTreeInstanceHeight = 0f

  The height which "destroyed" tree instances will get. Don't change if not needed.
- const int collidersPoolItemInstanceIncrease = 10000

  The height which "destroyed" tree instances will get. Don't change if not needed.
- float distanceOffset

  By how much will the terrain distance be still considered?
- Vector3 **lastSceneViewPosition** = Vector3.zero
- IEnumerator verifyTreeInstancesChangeRoutine

  An routine that is used from the editor to perform realtime tree instances updates.

Static Public Attributes

- static List< UNTerrain > terrains = new List<UNTerrain>()

  All of the current existing terrains in the current scene.

Protected Member Functions

- override void Awake ()

  Initiate startup variables
- virtual void OnTerrainChanged (int changedFlags)

  On terrain changed.
- virtual void CheckForTreeInstancesRespawns ()

  This method will check every set amount of time the trees in the terrain and restore them if needed.
- override void OnEnable ()

  Add this terrain to the terrains Pool
- override void OnDisable ()

  Remove this terrain to the terrains Pool
- override void OnPositionChanged (Vector3 newPosition)

  On terrain position changed

Properties

- Terrain **terrain** `[get, set]`
- UNTerrainData **terrainData** `[get]`
- UNTerrainSector **sector** `[get, set]`
- int sectorResolution `[get, set]`

  How much times will the terrain be divided? the more => the slower creation but higher performance on runtime.. the less => faster creation but lower performance on runtime.
- bool manageGrass `[get, set]`

  Will the system try to optimize your grass?
- bool **updateGrassOnHeightsChange** `[get, set]`
- bool manageTrees `[get, set]`

  Will the system try to optimize your trees?

Additional Inherited Members

5.101.1 Detailed Description

A class that needs to be on each terrain that needs to be taken into account when managing the system.

5.101.2 Member Function Documentation

5.101.2.1 override void uNature.Core.Terrains.UNTerrain.Awake ( ) `[protected], [virtual]`

Initiate startup variables

Reimplemented from uNature.Core.Targets.UNTarget.

5.101.2.2 override void uNature.Core.Terrains.UNTerrain.Check (Seekers.UNSeeker *seeker,* Vector3 *seekerPos,* float *seekingDistance,* bool *isPlaying*)

Check for seeker on terrain.

Parameters

seeker	Our seeker.
seekerPos	the seeker pos

5.101.2.3 virtual void uNature.Core.Terrains.UNTerrain.CheckForTreeInstancesRespawns ( ) [protected], [virtual]

This method will check every set amount of time the trees in the terrain and restore them if needed.

5.101.2.4 override void uNature.Core.Terrains.UNTerrain.CreatePool (System.Type *PoolItemType*) [virtual]

Fill up our Pool.

Reimplemented from uNature.Core.Targets.UNTarget.

5.101.2.5 override Vector3 uNature.Core.Terrains.UNTerrain.FixPosition (Vector3 *position*) [virtual]

Return the position that can be used with the chunks.

Parameters

position	the original position

Returns

position that can be used in local space with the terrain

Reimplemented from uNature.Core.Targets.UNTarget.

5.101.2.6 virtual UNTerrainSector uNature.Core.Terrains.UNTerrain.GenerateSector (int *sectorResolution*, bool *multiThread*) [virtual]

Generate a sector and assign it to the UNTerrain.

Parameters

sectorResolution	How many pieces will the terrain be divided to? the bigger it is the more pieces.

5.101.2.7 virtual UNTerrainSector uNature.Core.Terrains.UNTerrain.GenerateSector (int *sectorResolution*) [virtual]

Generate a sector and assign it to the UNTerrain.

Parameters

sectorResolution	How many pieces will the terrain be divided to? the bigger it is the more pieces.

5.101.2.8 override bool uNature.Core.Terrains.UNTerrain.InDistance (Seekers.UNSeeker *seeker*)

Check if the seeker is in range of the terrain.

Parameters

seeker	Seeker

Returns

in range?

5.101.2.9 override void uNature.Core.Terrains.UNTerrain.OnDisable ( ) [protected], [virtual]

Remove this terrain to the terrains Pool

Reimplemented from uNature.Core.Targets.UNTarget.

5.101.2.10 override void uNature.Core.Terrains.UNTerrain.OnEnable ( ) [protected], [virtual]

Add this terrain to the terrains Pool

Reimplemented from uNature.Core.Targets.UNTarget.

5.101.2.11 override void uNature.Core.Terrains.UNTerrain.OnPositionChanged (Vector3 *newPosition*) [protected], [virtual]

On terrain position changed

Parameters

newPosition	

Reimplemented from uNature.Core.Threading.ThreadItem.

5.101.2.12 virtual void uNature.Core.Terrains.UNTerrain.OnTerrainChanged (int *changedFlags*) [protected], [virtual]

On terrain changed.

Parameters

changedFlags	

5.101.3 Member Data Documentation

5.101.3.1 const int uNature.Core.Terrains.UNTerrain.collidersPoolItemInstanceIncrease = 10000

The height which "destroyed" tree instances will get. Don't change if not needed.

5.101.3.2 float uNature.Core.Terrains.UNTerrain.distanceOffset

By how much will the terrain distance be still considered?

5.101.3.3 const float uNature.Core.Terrains.UNTerrain.removedTreeInstanceHeight = 0f

The height which "destroyed" tree instances will get. Don't change if not needed.

5.101.3.4 List<UNTerrain> uNature.Core.Terrains.UNTerrain.terrains = new List<UNTerrain>() `[static]`

All of the current existing terrains in the current scene.

5.101.3.5 IEnumerator uNature.Core.Terrains.UNTerrain.verifyTreeInstancesChangeRoutine

An routine that is used from the editor to perform realtime tree instances updates.

5.101.4 Property Documentation

5.101.4.1 bool uNature.Core.Terrains.UNTerrain.manageGrass `[get], [set]`

Will the system try to optimize your grass?

Also, make sure to design the grass LODs if the grass doesnt work as you'd like (Window/uNature/Settings).

5.101.4.2 bool uNature.Core.Terrains.UNTerrain.manageTrees `[get], [set]`

Will the system try to optimize your trees?

5.101.4.3 int uNature.Core.Terrains.UNTerrain.sectorResolution [get], [set]

How much times will the terrain be divided? the more => the slower creation but higher performance on runtime.. the less => faster creation but lower performance on runtime.

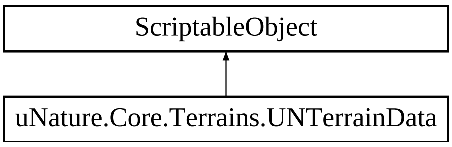
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/UNTerrain.cs

5.102 uNature.Core.Terrains.UNTerrainData Class Reference

The terrain data class which is used by uNature. Can be accesed by "UNTerrain.terrainData".

Inheritance diagram for uNature.Core.Terrains.UNTerrainData:



Public Member Functions

- UNTreePrototype GetPrototype (TreePrototype prototype)

  Get a terrain tree prototype (UNTerrainData) from a unity's tree prototype.
- void UpdateMultithreadedVariables ()

  Update the current multi-threaded variables on this terrain so it can be used on a different thread.
- void Backup ()

  Backup the terrain data.
- void ApplyBackup (Terrain terrain)

  Apply the current backup.
- void DeleteBackup ()

  Delete the current backup.
- void Initialize ()

  Initialize the terrain data.

Static Public Member Functions

- static string **GetBackUpPath** (string backUpName)
- static UNTerrainData GetInstance (TerrainData terrainData)

  Get the UNTerrainData instance from providing a terrainData.

Public Attributes

- Vector2 **FoliageRelativeMultiplier**
- int **multiThreaded_detailResolution**
- int **multiThreaded_detailWidth**
- int **multiThreaded_detailHeight**
- Vector3[,] **multiThreaded_terrainSampleNormals**
- DetailPrototype[ ] **multiThreaded_detailPrototypes**
- float **multiThreaded_heightMapWidth**
- float **multiThreaded_heightMapHeight**

Protected Member Functions

- virtual void CheckForTreePrototypesChange ()

  Checks and updates dirty terrain tree prototypes.
- void SendUpdateEventToLinkedTerrains (TerrainChangedFlags flag)

  Send an TerrainData changed event to all linked terrains.

Properties

- static string **persistentPath** `[get]`
- string **objectName** `[get, set]`
- TerrainData terrainData `[get, protected set]`

  The terrain data which this object resembles
- TerrainData **backedUpTerrainData** `[get]`
- List< UNTreePrototype > **treePrototypes** `[get]`
- float[,] heights `[get]`

  Get the heights of the terrain (used in multi-threading).
- bool **isDirty** `[get]`
- Vector3 **multiThreaded_terrainDataSize** `[get, set]`
- float[,] **multiThreaded_terrainHeights** `[get, set]`

5.102.1 Detailed Description

The terrain data class which is used by uNature. Can be accesed by "UNTerrain.terrainData".

5.102.2 Member Function Documentation

5.102.2.1 void uNature.Core.Terrains.UNTerrainData.ApplyBackup (Terrain *terrain*)

Apply the current backup.

5.102.2.2 void uNature.Core.Terrains.UNTerrainData.Backup ( )

Backup the terrain data.

5.102.2.3 virtual void uNature.Core.Terrains.UNTerrainData.CheckForTreePrototypesChange ( ) `[protected]`, `[virtual]`

Checks and updates dirty terrain tree prototypes.

5.102.2.4 void uNature.Core.Terrains.UNTerrainData.DeleteBackup ( )

Delete the current backup.

5.102.2.5 static UNTerrainData uNature.Core.Terrains.UNTerrainData.GetInstance (TerrainData *terrainData*) `[static]`

Get the UNTerrainData instance from providing a terrainData.

Parameters

terrainData	the terrain data which this UNTerrainData belongs to.

Returns

the UNTerrainData instance for the provided terrain data.

5.102.2.6 UNTreePrototype uNature.Core.Terrains.UNTerrainData.GetPrototype (TreePrototype *prototype*)

Get a terrain tree prototype (UNTerrainData) from a unity's tree prototype.

Parameters

prototype	

Returns

5.102.2.7 void uNature.Core.Terrains.UNTerrainData.Initialize ( )

Initialize the terrain data.

5.102.2.8 void uNature.Core.Terrains.UNTerrainData.SendUpdateEventToLinkedTerrains (TerrainChangedFlags *flag*) [protected]

Send an TerrainData changed event to all linked terrains.

5.102.2.9 void uNature.Core.Terrains.UNTerrainData.UpdateMultithreadedVariables ( )

Update the current multi-threaded variables on this terrain so it can be used on a different thread.

5.102.3 Property Documentation

5.102.3.1 float [,] uNature.Core.Terrains.UNTerrainData.heights [get]

Get the heights of the terrain (used in multi-threading).

5.102.3.2 TerrainData uNature.Core.Terrains.UNTerrainData.terrainData [get], [protected set]

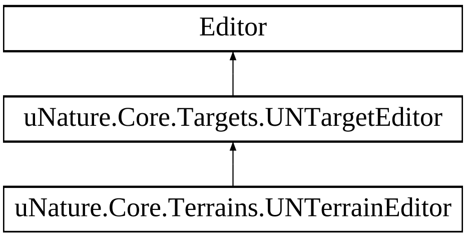
The terrain data which this object resembles

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/TerrainData/UNTerrainData.cs

5.103 uNature.Core.Terrains.UNTerrainEditor Class Reference

Inheritance diagram for uNature.Core.Terrains.UNTerrainEditor:


Editor
uNature.Core.Targets.UNTargetEditor
uNature.Core.Terrains.UNTerrainEditor

Public Member Functions

- override void **OnInspectorGUI** ()

Public Attributes

- List< UNTreePrototype > **selectedPrototypes** = new List<UNTreePrototype>()
- TerrainTabs **currentTab** = TerrainTabs.Grids

Properties

- List< SelectionBoxItems< System.Type > > **PoolCache** `[get]`

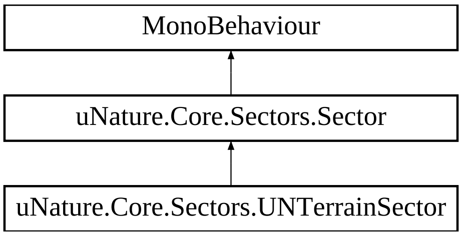
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNTerrainEditor.cs

5.104 uNature.Core.Sectors.UNTerrainSector Class Reference

An sector class dedicated only for terrains.

Inheritance diagram for uNature.Core.Sectors.UNTerrainSector:


MonoBehaviour
uNature.Core.Sectors.Sector
uNature.Core.Sectors.UNTerrainSector

Public Member Functions

- override void OnCreated (Transform owner, int resolution)

  Called when the object is created.

Parameters

terrain	*The terrain we belong to.*

- override void Awake ()

  Called on awake.
- void FetchTreeInstances (bool useUNThread, System.Action OnFinish)

  Get all the terrain tree instances into chunks

Parameters

useUNThread	*Do you want to use the uNature thread to reduce performance issues ?*

- override void ApplicationQuit ()

  Called when the application has quit.

Public Attributes

- UNTerrain **unTerrain**
- Terrain **_terrain**
- TreeInstance[ ] **originalTreeInstances**

Protected Member Functions

- override void OnChunkCreated (Chunk chunk)

  Called when a chunk is created to allow custom logic on the inherited sectors.
- override void OnStartCreatingChunks ()

  Called right before starting to create the chunks.

Properties

- Terrain **terrain** `[get, set]`
- int **treeInstancesCount** `[get, set]`
- List< TIChunk > **treeInstancesChunks** `[get]`
- bool **restoreComplete** `[get]`

Additional Inherited Members

5.104.1 Detailed Description

An sector class dedicated only for terrains.

5.104.2 Member Function Documentation

5.104.2.1 override void uNature.Core.Sectors.UNTerrainSector.ApplicationQuit ( ) `[virtual]`

Called when the application has quit.

Reimplemented from uNature.Core.Sectors.Sector.

5.104.2.2 override void uNature.Core.Sectors.UNTerrainSector.Awake ( ) [virtual]

Called on awake.

Reimplemented from uNature.Core.Sectors.Sector.

5.104.2.3 void uNature.Core.Sectors.UNTerrainSector.FetchTreeInstances (bool *useUNThread*, System.Action *OnFinish*)

Get all the terrain tree instances into chunks

Parameters

useUNThread	Do you want to use the uNature thread to reduce performance issues ?

5.104.2.4 override void uNature.Core.Sectors.UNTerrainSector.OnChunkCreated (Chunk *chunk*) [protected], [virtual]

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

chunk	

Reimplemented from uNature.Core.Sectors.Sector.

5.104.2.5 override void uNature.Core.Sectors.UNTerrainSector.OnCreated (Transform *owner*, int *resolution*) [virtual]

Called when the object is created.

Parameters

terrain	The terrain we belong to.

Reimplemented from uNature.Core.Sectors.Sector.

5.104.2.6 override void uNature.Core.Sectors.UNTerrainSector.OnStartCreatingChunks ( ) [protected], [virtual]

Called right before starting to create the chunks.

Reimplemented from uNature.Core.Sectors.Sector.

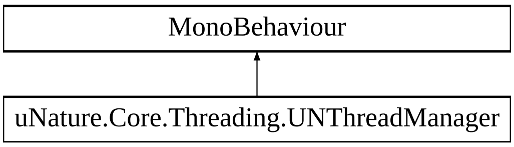
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/UNTerrainSector.cs

5.105 uNature.Core.Threading.UNThreadManager Class Reference

This class handles the multi-threading mechanics.

Inheritance diagram for uNature.Core.Threading.UNThreadManager:



Public Member Functions

- void UpdateThreadItems ()

  Updates the thread items in the scene.
- void RunOnUnityThread (IThreadTask action)

  Add an action to the unity thread
- void RunOnThread (IThreadTask action)

  Add an action to the UN thread
- void DelayActionSeconds (IThreadTask task, float time)

  Run any action with a specific delay of seconds.
- void DelayActionFrames (int frames, IThreadTask task)

  Run any action after 1 frame

Static Public Member Functions

- static void **InitializeIfNotAvailable** ()

Static Public Attributes

- static float updateThreadItemsTime = 0.1f

  How often will the thread manager update the thread items.

Protected Member Functions

- void OnThreadProcess (System.Object processObject)

  Called when the thread needs to process the task.

Properties

- static UNThreadManager **instance** `[get]`
- bool threadEnabled `[get]`

  Is the multi-thread option enabled?
- int threadWorkersCount `[get]`

  Thread workers count
- static bool **inUnityThread** `[get]`

5.105.1 Detailed Description

This class handles the multi-threading mechanics.

5.105.2 Member Function Documentation

5.105.2.1 void uNature.Core.Threading.UNThreadManager.DelayActionFrames (int *frames,* **IThreadTask** *task*)

Run any action after 1 frame

Parameters

task	the task you want to run after 1 frame

5.105.2.2 void uNature.Core.Threading.UNThreadManager.DelayActionSeconds (**IThreadTask** *task,* float *time*)

Run any action with a specific delay of seconds.

Parameters

task	the task you want to run after the specific amount of seconds
time	the specific amount of seconds to wait

5.105.2.3 void uNature.Core.Threading.UNThreadManager.OnThreadProcess (System.Object *processObject*) [protected]

Called when the thread needs to process the task.

Parameters

processObject	

5.105.2.4 void uNature.Core.Threading.UNThreadManager.RunOnThread (**IThreadTask** *action*)

Add an action to the UN thread

Parameters

action	the action

5.105.2.5 void uNature.Core.Threading.UNThreadManager.RunOnUnityThread (**IThreadTask** *action*)

Add an action to the unity thread

Parameters

action	the action

5.105.2.6 void uNature.Core.Threading.UNThreadManager.UpdateThreadItems ( )

Updates the thread items in the scene.

5.105.3 Member Data Documentation

5.105.3.1 float uNature.Core.Threading.UNThreadManager.updateThreadItemsTime = 0.1f [static]

How often will the thread manager update the thread items.

5.105.4 Property Documentation

5.105.4.1 bool uNature.Core.Threading.UNThreadManager.threadEnabled [get]

Is the multi-thread option enabled?

5.105.4.2 int uNature.Core.Threading.UNThreadManager.threadWorkersCount [get]

Thread workers count

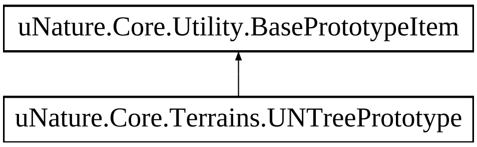
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.106 uNature.Core.Terrains.UNTreePrototype Class Reference

A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.

Inheritance diagram for uNature.Core.Terrains.UNTreePrototype:


uNature.Core.Utility.BasePrototypeItem
uNature.Core.Terrains.UNTreePrototype

Public Member Functions

- UNTreePrototype (TreePrototype prototype)

  Create a new instance of the object
- override bool Equals (object obj)

  Custom equals operator to take into account treePrototypes.
- override int GetHashCode ()

  Override to avoid warnings.

Static Public Member Functions

- static void CheckForMissings (List< UNTreePrototype > items, TreePrototype[ ] prototypes)

  This method will check whether any of the items is missing.

Public Attributes

- GameObject prototypeObject

  the game object of the tree prototype.
- bool isMissing = false

  Is this prototype missing on the terrainData? if so, make sure to wait for it to "comeback" and meanwhile store its data.
- int **prototypeIndex**

Protected Member Functions

- override Texture2D GetPreview ()

  Get Item Preview

Properties

- override bool isEnabled `[get]`

  Is this prototype missing?
- bool **enabled** `[get, set]`
- bool **forcePoolCreation** `[get, set]`

5.106.1 Detailed Description

A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.

5.106.2 Constructor & Destructor Documentation

5.106.2.1 uNature.Core.Terrains.UNTreePrototype.UNTreePrototype (TreePrototype *prototype*)

Create a new instance of the object

Parameters

prototype	The tree prototype this instance is based on.

5.106.3 Member Function Documentation

5.106.3.1 static void uNature.Core.Terrains.UNTreePrototype.CheckForMissings (List< UNTreePrototype > *items*, TreePrototype[ ] *prototypes*) [static]

This method will check whether any of the items is missing.

Parameters

items	the items list
prototypes	the tree prototypes of the terrain data

5.106.3.2 override bool uNature.Core.Terrains.UNTreePrototype.Equals (object *obj*)

Custom equals operator to take into account treePrototypes.

Parameters

obj	

Returns

5.106.3.3 override int uNature.Core.Terrains.UNTreePrototype.GetHashCode ( )

Override to avoid warnings.

Returns

5.106.3.4 override Texture2D uNature.Core.Terrains.UNTreePrototype.GetPreview ( ) [protected], [virtual]

Get Item Preview

Returns

Reimplemented from uNature.Core.Utility.BasePrototypeItem.

5.106.4 Member Data Documentation

5.106.4.1 bool uNature.Core.Terrains.UNTreePrototype.isMissing = false

Is this prototype missing on the terrainData? if so, make sure to wait for it to "comeback" and meanwhile store its data.

5.106.4.2 GameObject uNature.Core.Terrains.UNTreePrototype.prototypeObject

the game object of the tree prototype.

5.106.5 Property Documentation

5.106.5.1 override bool uNature.Core.Terrains.UNTreePrototype.isEnabled [get]

Is this prototype missing?

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/TerrainData/UNTerrainData.cs

5.107 uNature.Core.Vector2i Struct Reference

Public Member Functions

- **Vector2i** (int x, int y)
- override bool **Equals** (object obj)
- override int **GetHashCode** ()
- override string **ToString** ()

Static Public Member Functions

- static Vector2i **operator+** (Vector2i a, Vector2i b)
- static Vector2i **operator-** (Vector2i a, Vector2i b)
- static Vector2i **operator**∗ (Vector2i a, Vector2i b)
- static Vector2i **operator**/ (Vector2i a, Vector2i b)
- static bool **operator==** (Vector2i a, Vector2i b)
- static bool **operator!=** (Vector2i a, Vector2i b)

Public Attributes

- int **x**
- int **y**

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMath.cs

5.108 uNature.Core.Utility.Vector3_XZ_FAST Struct Reference

Public Member Functions

- **Vector3_XZ_FAST** (Vector3 v3)
- **Vector3_XZ_FAST** (Vector2 v2)
- **Vector3_XZ_FAST** (float x, float z)

Public Attributes

- float **x**
- float **z**

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNStandaloneUtility.cs

5.109 uNature.Core.FoliageClasses.WindSettings Class Reference

Properties

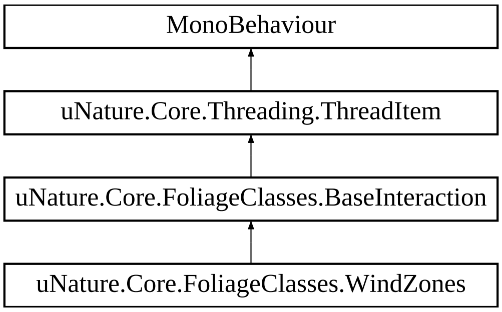
- float **windBending** `[get, set]`
- float **windSpeed** `[get, set]`

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageDB.cs

5.110 uNature.Core.FoliageClasses.WindZones Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.WindZones:


MonoBehaviour
uNature.Core.Threading.ThreadItem
uNature.Core.FoliageClasses.BaseInteraction
uNature.Core.FoliageClasses.WindZones

Additional Inherited Members

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

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/Interaction/WindZones.cs

Index