Poly Few | Mesh Simplifier and Auto LOD Generator

Generated by Doxygen 1.8.17

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:

Chapter 4

Namespace Documentation

4.1 BrainFailProductions Namespace Reference

4.2 BrainFailProductions.PolyFewRuntime Namespace Reference

Classes

- class PolyfewRuntime

Chapter 5

Class Documentation

5.1 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.Custom↩ MeshActionStructure Class Reference

This class represents a custom data structure that holds reference to a MeshRendererPair, the GameObject from which the MeshRendererPair was constructed and an Action object used to execute some code.

Public Member Functions

- **CustomMeshActionStructure** (MeshRendererPair meshRendererPair, GameObject gameObject, Action action)

Public Attributes

- MeshRendererPair meshRendererPair

  The MeshRendererPair constructed for the referenced GameObject. This contains the mesh associated with the GameObject if any and some other info about the mesh.
- GameObject gameObject

  The GameObject with which this data structure is associated with.
- Action action

  An action object that can hold some custom code to execute.

5.1.1 Detailed Description

This class represents a custom data structure that holds reference to a MeshRendererPair, the GameObject from which the MeshRendererPair was constructed and an Action object used to execute some code.

5.1.2 Member Data Documentation

5.1.2.1 action

```
Action BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CustomMeshActionStructure.action
```

An action object that can hold some custom code to execute.

5.1.2.2 gameObject

```
GameObject BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CustomMeshActionStructure.game↩
Object
```

The GameObject with which this data structure is associated with.

5.1.2.3 meshRendererPair

```
MeshRendererPair BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CustomMeshActionStructure.↩
meshRendererPair
```

The MeshRendererPair constructed for the referenced GameObject. This contains the mesh associated with the GameObject if any and some other info about the mesh.

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.2 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.Material↩ Properties Class Reference

Public Member Functions

- **MaterialProperties** (int texArrIndex, int matIndex, string materialName, Material originalMaterial, Color albedoTint, Vector4 uvTileOffset, float normalIntensity, float occlusionIntensity, float smoothnessIntensity, float glossMapScale, float metalIntensity, Color emissionColor, Vector4 detailUVTileOffset, float alphaCutoff, Color specularColor, float detailNormalScale, float heightIntensity, float uvSec)
- void **BurnAttrToImg** (ref Texture2D burnOn, int index, int textureArrayIndex)

Public Attributes

- readonly int texArrIndex

  The index in the texture array this material points to
- readonly int matIndex

  The index in the materials array this material points to
- readonly string materialName

  The name of the original material which was merged
- readonly Material originalMaterial

  A reference to the original material which was combined
- Color albedoTint

  The albedo tint color. Please note that the alpha value means nothing and can't be changed
- Vector4 uvTileOffset = new Vector4(1, 1, 0, 0)

  The UV tiling
- float normalIntensity = 1

  The normal intensity
- float occlusionIntensity = 1

  The occlusion intensity
- float smoothnessIntensity = 1

  The smoothness intensity
- float glossMapScale = 1

  The scale of the specular/gloss map
- float metalIntensity = 1

  The metal intensity
- Color emissionColor = Color.black

  The color of the emissive channel
- Vector4 detailUVTileOffset = new Vector4(1, 1, 0, 0)

  The uv tiling for detailed maps
- float alphaCutoff = 0.5f

  The alpha cutoff value
- Color specularColor = Color.black

  The specular channel color
- float detailNormalScale = 1

  The scale of the detailed normal map
- float heightIntensity = 0.05f

  The height intensity
- readonly float uvSec = 0

  Leave this alone

5.2.1 Member Data Documentation

5.2.1.1 albedoTint

```
Color BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.albedoTint
```

The albedo tint color. Please note that the alpha value means nothing and can't be changed

5.2.1.2 alphaCutoff

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.alphaCutoff = 0.5f
```

The alpha cutoff value

5.2.1.3 detailNormalScale

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.detailNormalScale
= 1
```

The scale of the detailed normal map

5.2.1.4 detailUVTileOffset

```
Vector4 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.detailUVTile↩
Offset = new Vector4(1, 1, 0, 0)
```

The uv tiling for detailed maps

5.2.1.5 emissionColor

```
Color BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.emissionColor =
Color.black
```

The color of the emissive channel

5.2.1.6 glossMapScale

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.glossMapScale = 1
```

The scale of the specular/gloss map

5.2.1.7 heightIntensity

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.heightIntensity =
0.05f
```

The height intensity

5.2.1.8 materialName

```
readonly string BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.material↩
Name
```

The name of the original material which was merged

5.2.1.9 matIndex

```
readonly int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.matIndex
```

The index in the materials array this material points to

5.2.1.10 metalIntensity

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.metalIntensity = 1
```

The metal intensity

5.2.1.11 normalIntensity

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.normalIntensity =
1
```

The normal intensity

5.2.1.12 occlusionIntensity

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.occlusionIntensity
= 1
```

The occlusion intensity

5.2.1.13 originalMaterial

```
readonly Material BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.↩
originalMaterial
```

A reference to the original material which was combined

5.2.1.14 smoothnessIntensity

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.smoothnessIntensity
= 1
```

The smoothness intensity

5.2.1.15 specularColor

```
Color BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.specularColor =
Color.black
```

The specular channel color

5.2.1.16 texArrIndex

```
readonly int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.texArrIndex
```

The index in the texture array this material points to

5.2.1.17 uvSec

```
readonly float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.uvSec = 0
```

Leave this alone

5.2.1.18 uvTileOffset

```
Vector4 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MaterialProperties.uvTileOffset =
new Vector4(1, 1, 0, 0)
```

The UV tiling

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.3 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MeshRendererPair Class Reference

This class represents a simple data structure that holds reference to a mesh and whether that mesh is part of a MeshRenderer (Attached to MeshFilter) or SkinnedMeshRenderer. This structure is used thoroughly in various mesh simplification operations.

Public Member Functions

- **MeshRendererPair** (bool attachedToMeshFilter, Mesh mesh)
- void **Destruct** ()

Public Attributes

- bool attachedToMeshFilter

  Whether mesh is part of a MeshRenderer (Attached to MeshFilter) or SkinnedMeshRenderer.
- Mesh mesh

  A reference to a mesh

5.3.1 Detailed Description

This class represents a simple data structure that holds reference to a mesh and whether that mesh is part of a MeshRenderer (Attached to MeshFilter) or SkinnedMeshRenderer. This structure is used thoroughly in various mesh simplification operations.

5.3.2 Member Data Documentation

5.3.2.1 attachedToMeshFilter

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MeshRendererPair.attachedToMeshFilter
```

Whether mesh is part of a MeshRenderer (Attached to MeshFilter) or SkinnedMeshRenderer.

5.3.2.2 mesh

```
Mesh BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MeshRendererPair.mesh
```

A reference to a mesh

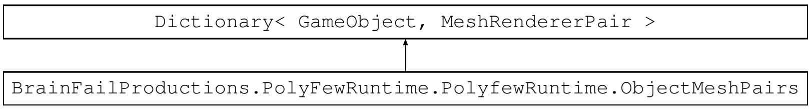
The documentation for this class was generated from the following file:

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_runtime_api_docs/PolyfewRuntime.cs

5.4 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ObjectMeshPairs Class Reference

A Dictionary that holds a GameObject as key and the associated MeshRendererPair as value

Inheritance diagram for BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ObjectMeshPairs:


Dictionary< GameObject, MeshRendererPair >
BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ObjectMeshPairs

5.4.1 Detailed Description

A Dictionary that holds a GameObject as key and the associated MeshRendererPair as value

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_runtime_api_docs/PolyfewRuntime.cs

5.5 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJExportOptions Class Reference

Options that define how the a GameObject will be exported to wavefront OBJ.

Public Member Functions

- **OBJExportOptions** (bool applyPosition, bool applyRotation, bool applyScale, bool generateMaterials, bool exportTextures)

Public Attributes

- readonly bool applyPosition = true

  When checked, the position of models will be taken into account on export.
- readonly bool applyRotation = true

  When checked, the rotation of models will be taken into account on export.
- readonly bool applyScale = true

  When checked, the scale of models will be taken into account on export.
- readonly bool generateMaterials = true

  Should the materials associated with the GameObject to export also be exported as .MTL files.
- readonly bool exportTextures = true

  Should the textures associated with the materials also be exported.

5.5.1 Detailed Description

Options that define how the a GameObject will be exported to wavefront OBJ.

5.5.2 Member Data Documentation

5.5.2.1 applyPosition

```
readonly bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJExportOptions.apply↩
Position = true
```

When checked, the position of models will be taken into account on export.

5.5.2.2 applyRotation

```
readonly bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJExportOptions.apply↩
Rotation = true
```

When checked, the rotation of models will be taken into account on export.

5.5.2.3 applyScale

```
readonly bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJExportOptions.applyScale =
true
```

When checked, the scale of models will be taken into account on export.

5.5.2.4 exportTextures

```
readonly bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJExportOptions.export↩
Textures = true
```

Should the textures associated with the materials also be exported.

5.5.2.5 generateMaterials

```
readonly bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJExportOptions.generate↩
Materials = true
```

Should the materials associated with the GameObject to export also be exported as .MTL files.

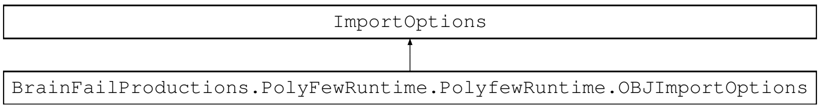
The documentation for this class was generated from the following file:

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.6 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJImport↩ Options Class Reference

Options that define how the model will be loaded and imported.

Inheritance diagram for BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJImportOptions:


ImportOptions
BrainFailProductions.PolyFewRuntime.PolyfewRuntime.OBJImportOptions

5.6.1 Detailed Description

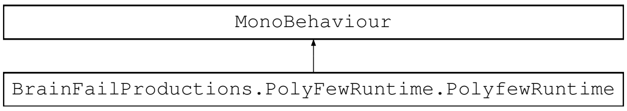
Options that define how the model will be loaded and imported.

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.7 BrainFailProductions.PolyFewRuntime.PolyfewRuntime Class Reference

Inheritance diagram for BrainFailProductions.PolyFewRuntime.PolyfewRuntime:



Classes

- class CustomMeshActionStructure

  This class represents a custom data structure that holds reference to a MeshRendererPair, the GameObject from which the MeshRendererPair was constructed and an Action object used to execute some code.
- class MaterialProperties
- class MeshRendererPair

  This class represents a simple data structure that holds reference to a mesh and whether that mesh is part of a MeshRenderer (Attached to MeshFilter) or SkinnedMeshRenderer. This structure is used thoroughly in various mesh simplification operations.
- class ObjectMeshPairs

  A Dictionary that holds a GameObject as key and the associated MeshRendererPair as value
- class OBJExportOptions

  Options that define how the a GameObject will be exported to wavefront OBJ.
- class OBJImportOptions

  Options that define how the model will be loaded and imported.
- class PreservationSphere

  This class is used to represent a preservation sphere. A preservation sphere retains the original quality of the mesh area enclosed within it while simplifying all other areas of the mesh. Please note that mesh simplification with preservation spheres might get slow.
- class ReferencedNumeric

  A wrapper class that holds a primitive numeric type and fakes them to act as reference types.
- class SimplificationOptions

  This class holds all the available options for mesh simplification. An object of this class is needed by many of the Mesh Simplification methods for controlling the mesh simplification process.

Public Types

- enum MeshCombineTarget { **SkinnedAndStatic**, **StaticOnly**, **SkinnedOnly** }

  An enum that is used to specify what kind of meshes to combine

Static Public Member Functions

- static int SimplifyObjectDeep (GameObject toSimplify, SimplificationOptions simplificationOptions, Action< GameObject, MeshRendererPair > OnEachMeshSimplified)

  Simplifies the provided gameobject include the full nested children hierarchy with the settings provided. Any errors are thrown as exceptions with relevant information. Please note that the method won't simplify the object if the simplification strength provided in the SimplificationOptions is close to 0.
- static ObjectMeshPairs SimplifyObjectDeep (GameObject toSimplify, SimplificationOptions simplificationOptions)

  Simplifies the meshes nested under the given gameobject(including itself) including the full nested children hierarchy with the settings provided. Retuns back a specialized data structure with the simplified meshes. Any errors are thrown as exceptions with relevant information. Please note that the method won't simplify the object if the simplification strength provided in the SimplificationOptions is close to 0.
- static int SimplifyObjectDeep (ObjectMeshPairs objectMeshPairs, SimplificationOptions simplificationOptions, Action< GameObject, MeshRendererPair > OnEachMeshSimplified)

  Simplifies the meshes provided in the "objectMeshPairs" argument and assigns the simplified meshes to the corresponding objects. Any errors are thrown as exceptions with relevant information. Please note that the method won't simplify the object if the simplification strength provided in the SimplificationOptions is close to 0.
- static List< Mesh > SimplifyMeshes (List< Mesh > meshesToSimplify, SimplificationOptions simplificationOptions, Action< Mesh > OnEachMeshSimplified)

Simplifies the meshes provided in the "meshesToSimplify" argument and returns the simplified meshes in a new list. Any errors are thrown as exceptions with relevant information.Please note that the returned list of simplified meshes doesn't guarantee the same order of meshes as supplied in the "meshesToSimplify" list. Please note that preservation spheres don't work with this method.

- static ObjectMeshPairs GetObjectMeshPairs (GameObject forObject, bool includeInactive)

  This method returns a specialized DataStructure for the provided object. The key is a reference to a GameObject and the value is a MeshRendererPair which contains a reference to the mesh attached to the GameObject (key) and the type of mesh (Skinned or static).

- static void CombineMeshesInGameObject (GameObject forObject, bool skipInactiveRenderers, Action< string, string > OnError, MeshCombineTarget combineTarget=MeshCombineTarget.SkinnedAndStatic)

  Tries to combine meshes nested under the provided GameObject. Please note that the method modifies the provided gameobject and it's children hierarchy.

- static GameObject CombineMeshesFromRenderers (Transform rootTransform, MeshRenderer[ ] original←MeshRenderers, SkinnedMeshRenderer[ ] originalSkinnedMeshRenderers, Action< string, string > OnError)

  Tries to combine the static and skinned meshes provided in the arguments.

- static void ConvertSkinnedMeshesInGameObject (GameObject forObject, bool skipInactiveRenderers, Action< string, string > OnError)

  Converts all skinned meshes in the provided GameObject to non skinned/static meshes and also changes the corresponding renderer components. Please note that this method modifies the original GameObject and it's child hierarchy.

- static Tuple< SkinnedMeshRenderer, MeshRenderer, Mesh >[ ] ConvertSkinnedMeshesFromRenderers (SkinnedMeshRenderer[ ] renderersToConvert, Action< string, string > OnError)

  Converts all Skinned Mesh Renderers in the provided list to non simple Mesh Renderers and also changes the corresponding meshes. Pleaes note that this method doesn't modify the actualy GameObject(s) from which the provided SkinnedMeshRenderes are extracted.

- static async void ImportOBJFromFileSystem (string objAbsolutePath, string texturesFolderPath, string materialsFolderPath, Action< GameObject > OnSuccess, Action< Exception > OnError, OBJImportOptions importOptions=null)

  Imports a wavefront obj file provided by the absolute path. Please note that this method doesn't work on WebGL builds and will safely return.

- static async void ImportOBJFromNetwork (string objURL, string objName, string diffuseTexURL, string bumpTexURL, string specularTexURL, string opacityTexURL, string materialURL, ReferencedNumeric< float > downloadProgress, Action< GameObject > OnSuccess, Action< Exception > OnError, OBJImportOptions importOptions=null)

  Downloads a wavefront obj file from the direct URl passed and imports it. You can also specify the URL for different textures associated with the model and also the URL to the linked material file. This function also works on WebGL builds.

- static async void ExportGameObjectToOBJ (GameObject toExport, string exportPath, Action OnSuccess, Action< Exception > OnError, OBJExportOptions exportOptions=null)

  Exports the provided GameObject to wavefront OBJ format with support for saving textures and materials. Please note that the method won't work on WebGL builds and will safely return.

- static int CountTriangles (bool countDeep, GameObject forObject)

  Counts the number of triangles in the provided GameObject. If "countDeep" is true then the method counts all the triangles considering all the nested meshes in the children hierarchies of the given GameObject.

- static int CountTriangles (List< Mesh > toCount)

  Counts the number of triangles in the provided meshes list.

- static List< MaterialProperties > GetMaterialsProperties (GameObject forObject)

  Get a list of MaterialProperties objects associated with this GameObject. These objects can be used to change the properties of individual materials on this GameObject that were combined with Batch Few

- static void ChangeMaterialProperties (MaterialProperties changeTo, GameObject forObject)

  Change the properties of a merged material associated with this GameObject

5.7.1 Member Enumeration Documentation

5.7.1.1 MeshCombineTarget

```
enum BrainFailProductions.PolyFewRuntime.PolyfewRuntime.MeshCombineTarget  [strong]
```

An enum that is used to specify what kind of meshes to combine

5.7.2 Member Function Documentation

5.7.2.1 ChangeMaterialProperties()

```
static void BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ChangeMaterialProperties (
            MaterialProperties changeTo,
            GameObject forObject )  [static]
```

Change the properties of a merged material associated with this GameObject

Parameters

changeTo	The new material properties
forObject	The GameObject whose merged material to change properties for

5.7.2.2 CombineMeshesFromRenderers()

```
static GameObject BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CombineMeshesFromRenderers
(
            Transform rootTransform,
            MeshRenderer[ ] originalMeshRenderers,
            SkinnedMeshRenderer[ ] originalSkinnedMeshRenderers,
            Action< string, string > OnError )  [static]
```

Tries to combine the static and skinned meshes provided in the arguments.

Parameters

rootTransform	The root transform to create the combined meshes based from, essentially the origin of the new mesh.
originalMeshRenderers	The list of MeshRenderer components whose corresponding meshes to combine.
OnError	The method to invoke when an error occurs. The method is passed the error title and the description of the error.
originalSkinnedMeshRenderers	The list of SkinnedMeshRenderer components whose corresponding meshes to combine.

Returns

A new GameObject with the combined meshes, or returns null in case of any problem.

5.7.2.3 CombineMeshesInGameObject()

```
static void BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CombineMeshesInGameObject (
             GameObject forObject,
             bool skipInactiveRenderers,
             Action< string, string > OnError,
             MeshCombineTarget combineTarget = MeshCombineTarget.SkinnedAndStatic )  [static]
```

Tries to combine meshes nested under the provided GameObject. Please note that the method modifies the provided gameobject and it's children hierarchy.

Parameters

forObject	The object under which all the Static and Skinned meshes will be merged.
skipInactiveRenderers	Whether the child renderers of the provided objects be skipped if they are inactive.
OnError	The method to invoke when an error occurs. The method is passed the error title and the description of the error.
combineTarget	Indicates what kind of meshes to combine.

5.7.2.4 ConvertSkinnedMeshesFromRenderers()

```
static Tuple<SkinnedMeshRenderer, MeshRenderer, Mesh> [] BrainFailProductions.PolyFewRuntime.↩
PolyfewRuntime.ConvertSkinnedMeshesFromRenderers (
             SkinnedMeshRenderer[] renderersToConvert,
             Action< string, string > OnError )  [static]
```

Converts all Skinned Mesh Renderers in the provided list to non simple Mesh Renderers and also changes the corresponding meshes. Pleaes note that this method doesn't modify the actualy GameObject(s) from which the provided SkinnedMeshRenderes are extracted.

Parameters

renderersToConvert	The list of SkinnedMeshRenderers to convert.
OnError	The method to invoke when an error occurs. The method is passed the error title and the description of the error.

Returns

An array of tuples that contain the original SkinnedMeshRenderer, the corresponding converted Mesh↩ Renderer and the Mesh. Returns null if the operation failed.

5.7.2.5 ConvertSkinnedMeshesInGameObject()

```
static void BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ConvertSkinnedMeshesInGame↩
Object (
            GameObject forObject,
            bool skipInactiveRenderers,
            Action< string, string > OnError )  [static]
```

Converts all skinned meshes in the provided GameObject to non skinned/static meshes and also changes the corresponding renderer components. Please note that this method modifies the original GameObject and it's child hierarchy.

Parameters

forObject	The game object under which all SkinnedMeshRenderers will be converted.
skipInactiveRenderers	Whether the child renderers of the provided objects be skipped if they are inactive.
OnError	The method to invoke when an error occurs. The method is passed the error title and the description of the error.

5.7.2.6 CountTriangles() [1/2]

```
static int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CountTriangles (
            bool countDeep,
            GameObject forObject )  [static]
```

Counts the number of triangles in the provided GameObject. If "countDeep" is true then the method counts all the triangles considering all the nested meshes in the children hierarchies of the given GameObject.

Parameters

countDeep	If true the method also counts and considers the triangles of the nested children hierarchies for the given GameObject.
forObject	The GameObject for which to count the triangles.

Returns

The total traingles summing the triangles count of all the meshes nested under the provided GameObject.

5.7.2.7 CountTriangles() [2/2]

```
static int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.CountTriangles (
            List< Mesh > toCount )  [static]
```

Counts the number of triangles in the provided meshes list.

Parameters

toCount	The list of meshes whose triangles will be counted.

Returns

The total triangles summing the triangles count of all the meshes in the provided list. WIll return 0 if there are no meshes in the list

5.7.2.8 ExportGameObjectToOBJ()

```
static async void BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ExportGameObjectToOBJ (
            GameObject toExport,
            string exportPath,
            Action OnSuccess,
            Action< Exception > OnError,
            OBJExportOptions exportOptions = null )  [static]
```

Exports the provided GameObject to wavefront OBJ format with support for saving textures and materials. Please note that the method won't work on WebGL builds and will safely return.

Parameters

toExport	The GameObject that will be exported.
exportPath	The path to the folder where the file will be written.
exportOptions	Some additional export options for customizing the export.
OnSuccess	The callback to be invoked on successful export.
OnError	The callback method that will be invoked when the import was not successful. The method is passed in an exception that made the task unsuccessful.

5.7.2.9 GetMaterialsProperties()

```
static List<MaterialProperties> BrainFailProductions.PolyFewRuntime.PolyfewRuntime.GetMaterials←↩
Properties (
            GameObject forObject )  [static]
```

Get a list of MaterialProperties objects associated with this GameObject. These objects can be used to change the properties of individual materials on this GameObject that were combined with Batch Few

Parameters

forObject	The GameObject whose material properties objects to fetch

Returns

A list of MaterialProperties objects associated with this GameObject

5.7.2.10 GetObjectMeshPairs()

```
static ObjectMeshPairs BrainFailProductions.PolyFewRuntime.PolyfewRuntime.GetObjectMeshPairs (
             GameObject forObject,
             bool includeInactive )  [static]
```

This method returns a specialized DataStructure for the provided object. The key is a reference to a GameObject and the value is a MeshRendererPair which contains a reference to the mesh attached to the GameObject (key) and the type of mesh (Skinned or static).

Parameters

forObject	The object for which the ObjectMeshPairs is constructed.
includeInactive	If this is true then the method also considers the nested inactive children of the GameObject provided, otherwise it only considers the active nested children.

Returns

A specialized data structure that contains information about all the meshes nested under the provided GameObject.

5.7.2.11 ImportOBJFromFileSystem()

```
static async void BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ImportOBJFromFileSystem (
             string objAbsolutePath,
             string texturesFolderPath,
             string materialsFolderPath,
             Action< GameObject > OnSuccess,
             Action< Exception > OnError,
             OBJImportOptions importOptions = null )  [static]
```

Imports a wavefront obj file provided by the absolute path. Please note that this method doesn't work on WebGL builds and will safely return.

Parameters

objAbsolutePath	The absolute path to the obj file.
texturesFolderPath	The absolute path to the folder containing the texture files associated with the model to load. If you don't want to load the associated textures or there are none then you can pass an empty or null to this argument.
materialsFolderPath	The absolute path to the folder containing the material files assoicated with the model to load. If you don't want to load the associated material or there is none then you can pass an empty or null to this argument.
OnSuccess	The callback method that will be invoked when the import was successful. The method is passed in the imported GameObject as the argument.
OnError	The callback method that will be invoked when the import was not successful. The method is passed in an exception that made the task unsuccessful.
importOptions	Specify additional import options for custom importing.

5.7.2.12 ImportOBJFromNetwork()

```
static async void BrainFailProductions.PolyFewRuntime.PolyfewRuntime.ImportOBJFromNetwork (
             string objURL,
             string objName,
             string diffuseTexURL,
             string bumpTexURL,
             string specularTexURL,
             string opacityTexURL,
             string materialURL,
             ReferencedNumeric< float > downloadProgress,
             Action< GameObject > OnSuccess,
             Action< Exception > OnError,
             OBJImportOptions importOptions = null )  [static]
```

Downloads a wavefront obj file from the direct URl passed and imports it. You can also specify the URL for different textures associated with the model and also the URL to the linked material file. This function also works on WebGL builds.

Parameters

objURL	The direct URL to the obj file.
objName	The name for the GameObject that will represent the imported obj.
diffuseTexURL	The absolute URL to the associated Diffuse texture (Main texture). If the model has no diffuse texture on the material then you can pass in null or empty string to this parameter.
bumpTexURL	The absolute URL to the associated Bump texture (Bump map). If the model has no bump map then you can pass in null or empty string to this parameter.
specularTexURL	The absolute URL to the associated Specular texture (Reflection map). If the model has no reflection map then you can pass in null or empty string to this parameter.
opacityTexURL	The absolute URL to the associated Opacity texture (Transparency map). If the model has no transparency map then you can pass in null or empty string to this parameter.
materialURL	If the model has an associated material file (.mtl) then pass in the absolute URL to that otherwise pass a null or empty string.
downloadProgress	The object of type ReferencedNumeric of type float that is updated with the download progress percentage.
OnSuccess	The callback method that will be invoked when the import was successful. The method is passed in the imported GameObject as the argument..
OnError	The callback method that will be invoked when the import was not successful. The method is passed in an exception that made the task unsuccessful.
importOptions	Specify additional import options for custom importing.

5.7.2.13 SimplifyMeshes()

```
static List<Mesh> BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplifyMeshes (
             List< Mesh > meshesToSimplify,
             SimplificationOptions simplificationOptions,
             Action< Mesh > OnEachMeshSimplified )  [static]
```

Simplifies the meshes provided in the "meshesToSimplify" argument and returns the simplified meshes in a new list. Any errors are thrown as exceptions with relevant information.Please note that the returned list of simplified meshes doesn't guarantee the same order of meshes as supplied in the "meshesToSimplify" list. Please note that preservation spheres don't work with this method.

Parameters

meshesToSimplify	The list of meshes to simplify.
simplificationOptions	Provide a SimplificationOptions object which contains different parameters and rules for simplifying the meshes. Please note that preservationSphere won't work for this method.
OnEachMeshSimplified	This method will be called when a mesh is simplified. The method will be passed the original mesh that was simplified.

Returns

The list of simplified meshes.

5.7.2.14 SimplifyObjectDeep() [1/3]

```
static ObjectMeshPairs BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplifyObjectDeep (
            GameObject toSimplify,
            SimplificationOptions simplificationOptions ) [static]
```

Simplifies the meshes nested under the given gameobject(including itself) including the full nested children hierarchy with the settings provided. Retuns back a specialized data structure with the simplified meshes. Any errors are thrown as exceptions with relevant information. Please note that the method won't simplify the object if the simplification strength provided in the SimplificationOptions is close to 0.

Parameters

toSimplify	The gameobject to simplify.
simplificationOptions	Provide a SimplificationOptions object which contains different parameters and rules for simplifying the meshes.
OnEachMeshSimplified	This method will be called when a mesh is simplified. The method will be passed a gameobject whose mesh is simplified and some information about the original unsimplified mesh.

Returns

A specialized data structure that holds information about all the simplified meshes and their information and the GameObjects with which they are associated. Please note that in case the simplificationStrength was near 0 the method doesn't simplify any meshes and returns null.

5.7.2.15 SimplifyObjectDeep() [2/3]

```
static int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplifyObjectDeep (
            GameObject toSimplify,
            SimplificationOptions simplificationOptions,
            Action< GameObject, MeshRendererPair > OnEachMeshSimplified )  [static]
```

Simplifies the provided gameobject include the full nested children hierarchy with the settings provided. Any errors are thrown as exceptions with relevant information. Please note that the method won't simplify the object if the simplification strength provided in the SimplificationOptions is close to 0.

Parameters

toSimplify	The gameobject to simplify.
simplificationOptions	Provide a SimplificationOptions object which contains different parameters and rules for simplifying the meshes.
OnEachMeshSimplified	This method will be called when a mesh is simplified. The method will be passed a gameobject whose mesh is simplified and some information about the original unsimplified mesh. If you donot want to receive this callback then you can pass null as an argument here.

Returns

The total number of triangles after simplifying the provided gameobject inlcuding the nested children hierarchies. Please note that the method returns -1 if the method doesn't simplify the object.

5.7.2.16 SimplifyObjectDeep() [3/3]

```
static int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplifyObjectDeep (
            ObjectMeshPairs objectMeshPairs,
            SimplificationOptions simplificationOptions,
            Action< GameObject, MeshRendererPair > OnEachMeshSimplified )  [static]
```

Simplifies the meshes provided in the "objectMeshPairs" argument and assigns the simplified meshes to the corresponding objects. Any errors are thrown as exceptions with relevant information. Please note that the method won't simplify the object if the simplification strength provided in the SimplificationOptions is close to 0.

Parameters

objectMeshPairs	The ObjectMeshPairs data structure which holds relationship between objects and the corresponding meshes which will be simplified. You can get this structure by calling "GetObjectMeshPairs(GameObject forObject, bool includeInactive)" method.
simplificationOptions	Provide a SimplificationOptions object which contains different parameters and rules for simplifying the meshes.
OnEachMeshSimplified	This method will be called when a mesh is simplified. The method will be passed a gameobject whose mesh is simplified and some information about the original unsimplified mesh. If you donot want to receive this callback then you can pass null as an argument here.

Returns

The total number of triangles after simplifying the provided gameobject inlcuding the nested children hierarchies. Please note that the method returns -1 is the method doesn't simplify the object.

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.8 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.↩ PreservationSphere Class Reference

This class is used to represent a preservation sphere. A preservation sphere retains the original quality of the mesh area enclosed within it while simplifying all other areas of the mesh. Please note that mesh simplification with preservation spheres might get slow.

Public Member Functions

- **PreservationSphere** (Vector3 worldPosition, float diameter, float preservationStrength)

Public Attributes

- Vector3 worldPosition

  The position of this preservation sphere in world coordinates. Please note that this position should accurately represent the center point of the sphere.
- float diameter

  The diameter of this preservation sphere.
- float preservationStrength = 100

  The percentage of triangles to preserve in the region enclosed by this preservation sphere.

5.8.1 Detailed Description

This class is used to represent a preservation sphere. A preservation sphere retains the original quality of the mesh area enclosed within it while simplifying all other areas of the mesh. Please note that mesh simplification with preservation spheres might get slow.

5.8.2 Member Data Documentation

5.8.2.1 diameter

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.PreservationSphere.diameter
```

The diameter of this preservation sphere.

5.8.2.2 preservationStrength

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.PreservationSphere.preservation↩
Strength = 100
```

The percentage of triangles to preserve in the region enclosed by this preservation sphere.

5.8.2.3 worldPosition

```
Vector3 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.PreservationSphere.worldPosition
```

The position of this preservation sphere in world coordinates. Please note that this position should accurately represent the center point of the sphere.

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.9 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.↩ ReferencedNumeric< T > Class Template Reference

A wrapper class that holds a primitive numeric type and fakes them to act as reference types.

Public Member Functions

- **ReferencedNumeric** (T value)

Properties

- T **Value** `[get, set]`

5.9.1 Detailed Description

A wrapper class that holds a primitive numeric type and fakes them to act as reference types.

Template Parameters

T	Any primitive numeric type. Int, float, double, byte etc

Type Constraints

T* : *struct

T* : *IComparable

***T* : *IComparable*<*T*>**

T* : *IConvertible

***T* : *IEquatable*<*T*>**

T* : *IFormattable

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

5.10 BrainFailProductions.PolyFewRuntime.PolyfewRuntime.↩ SimplificationOptions Class Reference

This class holds all the available options for mesh simplification. An object of this class is needed by many of the Mesh Simplification methods for controlling the mesh simplification process.

Public Member Functions

- **SimplificationOptions** (float simplificationStrength, bool simplifyOptimal, bool enableSmartlink, bool recalculateNormals, bool preserveUVSeamEdges, bool preserveUVFoldoverEdges, bool preserveBorderEdges, bool regardToleranceSphere, List< PreservationSphere > preservationSpheres, bool regardCurvature, int maxIterations, float aggressiveness, bool useEdgeSort)

Public Attributes

- float simplificationStrength

  The strength with which to reduce the polygons by. Greater strength results in fewer polygons but lower quality. The acceptable values are between [0-100] inclusive.
- bool simplifyMeshLossless = false

  If set to true the mesh is simplified without loosing too much quality. Please note that simplify lossless cannot guarantee optimal triangle count after simplification. It's best that you specify the simplificationStrength manually and leave this to false. Also in case if this is true then the "simplificationStrength" attribute will be disregarded.
- bool enableSmartlinking = true

  Smart linking links vertices that are very close to each other. This helps in the mesh simplification process where holes or other serious issues could arise. Disabling this (where not needed) can cause a minor performance gain.
- bool recalculateNormals = false

  Recalculate mesh normals after simplification. Use this option if you see incorrect lighting or dark regions on the simplified mesh(es). This also recalculates the tangents afterwards.
- bool preserveUVSeamEdges = false

  This option (if set to true) preserves the mesh areas where the UV seams are made. These are the areas where different UV islands are formed (usually the shallow polygon conjested areas).
- bool preserveUVFoldoverEdges = false

  This option (if set to true) preserves UV foldover areas. Usually these are the areas where sharp edges, corners or dents are formed in the mesh or simply the areas where the mesh folds over.

- bool preserveBorderEdges = false

  This option (if set to true) preserves border edges of the mesh. Border edges are the edges that are unconnected and open. Preserving border edges might lead to lesser polygon reduction but can be helpful where you see serious mesh and texture distortions.
- bool regardPreservationSpheres = false

  This option (if set to true) will take into account the preservation spheres (If specified in the SimplificationOptions). A preservation sphere retains the original quality of the mesh area enclosed within it while simplifying all other areas of the mesh. Please note that mesh simplification with preservation spheres might get slow.
- List< PreservationSphere > preservationSpheres = new List<PreservationSphere>()

  The list of preservation spheres that dictate which areas of the mesh to preserve during simplification. This list will only be regarded if "regardPreservationSphere" option is set to true.
- bool regardCurvature = false

  This option (if set to true) will take into account the discrete curvature of mesh surface during simplification. Taking surface curvature into account can result in very good quality mesh simplification, but it can slow the simplification process significantly.
- int maxIterations = 100

  The maximum passes the reduction algorithm does. Higher number is more expensive but can bring you closer to your target quality. 100 is the lowest allowed value. The default value of 100 works best for most of the meshes and should not be changed.
- float aggressiveness = 7

  The agressiveness of the reduction algorithm to use for this LOD level. Higher number equals higher quality, but more expensive to run. Lowest value is 7. The default value of 7 works best for most of the meshes and should not be changed.
- bool useEdgeSort = false

  Using edge sort can result in very good quality mesh simplification in some cases but can be a little slow to run.

5.10.1 Detailed Description

This class holds all the available options for mesh simplification. An object of this class is needed by many of the Mesh Simplification methods for controlling the mesh simplification process.

5.10.2 Member Data Documentation

5.10.2.1 aggressiveness

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.aggressiveness
= 7
```

The agressiveness of the reduction algorithm to use for this LOD level. Higher number equals higher quality, but more expensive to run. Lowest value is 7. The default value of 7 works best for most of the meshes and should not be changed.

5.10.2.2 enableSmartlinking

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.enableSmartlinking
= true
```

Smart linking links vertices that are very close to each other. This helps in the mesh simplification process where holes or other serious issues could arise. Disabling this (where not needed) can cause a minor performance gain.

5.10.2.3 maxIterations

```
int BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.maxIterations =
100
```

The maximum passes the reduction algorithm does. Higher number is more expensive but can bring you closer to your target quality. 100 is the lowest allowed value. The default value of 100 works best for most of the meshes and should not be changed.

5.10.2.4 preservationSpheres

```
List<PreservationSphere> BrainFailProductions.PolyFewRuntime.PolyfewRuntime.Simplification↩
Options.preservationSpheres = new List<PreservationSphere>()
```

The list of preservation spheres that dictate which areas of the mesh to preserve during simplification. This list will only be regarded if "regardPreservationSphere" option is set to true.

5.10.2.5 preserveBorderEdges

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.preserveBorder↩
Edges = false
```

This option (if set to true) preserves border edges of the mesh. Border edges are the edges that are unconnected and open. Preserving border edges might lead to lesser polygon reduction but can be helpful where you see serious mesh and texture distortions.

5.10.2.6 preserveUVFoldoverEdges

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.preserveUV↩
FoldoverEdges = false
```

This option (if set to true) preserves UV foldover areas. Usually these are the areas where sharp edges, corners or dents are formed in the mesh or simply the areas where the mesh folds over.

5.10.2.7 preserveUVSeamEdges

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.preserveUVSeam↩
Edges = false
```

This option (if set to true) preserves the mesh areas where the UV seams are made. These are the areas where different UV islands are formed (usually the shallow polygon conjested areas).

5.10.2.8 recalculateNormals

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.recalculate←↩
Normals = false
```

Recalculate mesh normals after simplification. Use this option if you see incorrect lighting or dark regions on the simplified mesh(es). This also recalculates the tangents afterwards.

5.10.2.9 regardCurvature

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.regardCurvature
= false
```

This option (if set to true) will take into account the discrete curvature of mesh surface during simplification. Taking surface curvature into account can result in very good quality mesh simplification, but it can slow the simplification process significantly.

5.10.2.10 regardPreservationSpheres

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.regardPreservation←↩
Spheres = false
```

This option (if set to true) will take into account the preservation spheres (If specified in the SimplificationOptions). A preservation sphere retains the original quality of the mesh area enclosed within it while simplifying all other areas of the mesh. Please note that mesh simplification with preservation spheres might get slow.

5.10.2.11 simplificationStrength

```
float BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.simplification←↩
Strength
```

The strength with which to reduce the polygons by. Greater strength results in fewer polygons but lower quality. The acceptable values are between [0-100] inclusive.

5.10.2.12 simplifyMeshLossless

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.simplifyMesh←↩
Lossless = false
```

If set to true the mesh is simplified without loosing too much quality. Please note that simplify lossless cannot guarantee optimal triangle count after simplification. It's best that you specify the simplificationStrength manually and leave this to false. Also in case if this is true then the "simplificationStrength" attribute will be disregarded.

5.10.2.13 useEdgeSort

```
bool BrainFailProductions.PolyFewRuntime.PolyfewRuntime.SimplificationOptions.useEdgeSort =
false
```

Using edge sort can result in very good quality mesh simplification in some cases but can be a little slow to run.

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

- D:/LOST STUFF/LOST UNITY/UNITY ASSET STORE PACKAGES/Poly Few/Documentation/polyfew_↩ runtime_api_docs/PolyfewRuntime.cs

Index