Per-Vertex Ambient Occlusion and Indirect Lighting Generator API

To get access to the API add statement to a script file:

C# - `using VacuumShaders.AmbientOcclusionGenerator;`

Javascript - `import VacuumShaders.AmbientOcclusionGenerator;`

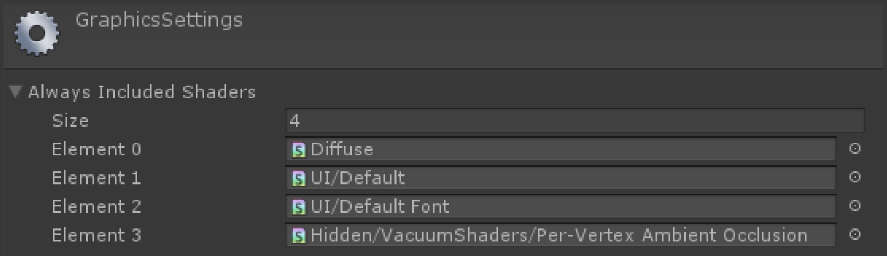
Static public class **Generator** contains:

1. Function for calculating Ambient Occlusion and Indirect Lighting

```
static public Color[] Generate(GameObject _gameObject,
                               AmbientOcclusion _ambientOcclusion,
                               IndirectLighting _indirectLighting);
```

- _gameObject – Gameobject for which is generating ambient occlusion (AO) and indirect Lighting (IL). Must contain mesh data through MeshFilter or SkinnedMeshRenderer components, otherwise will be returned **- NULL**.
- _ambientOcclusion – Contains data for AO calculation.
- _indirectLighting – Contains data for IL calculation.

Note, build project must contain **Hidden/VacuumShaders/Per-Vertex Ambient Occlusion.shader** in GraphicsSettings Always Include Shaders array, otherwise will be returned **- NULL**.



GraphicsSettings

▼ Always Included Shaders

Size	4
Element 0	Diffuse
Element 1	UI/Default
Element 2	UI/Default Font
Element 3	Hidden/VacuumShaders/Per-Vertex Ambient Occlusion

2. Function for multiple mesh combining

```
static public COMBINE_INFO CombineMeshes(Transform _parent, out Mesh _combinedMesh);
```

- _parent – Transform which hierarchy (children) meshes should be combined into one.
- _combinedMesh – Combined resultant mesh.

Note, cannot combine skinned meshes with final skin and function will not combine meshes if:

I. _parent == null
II. There is no mesh data
III. Any MeshFilter or SkinnedMeshRenderer component has no mesh data.

3. Function for combining submeshes

```
static public void CombineSubMeshes(Mesh _origMesh, out Mesh _combinedMesh);
```

Check example scenes and scripts.