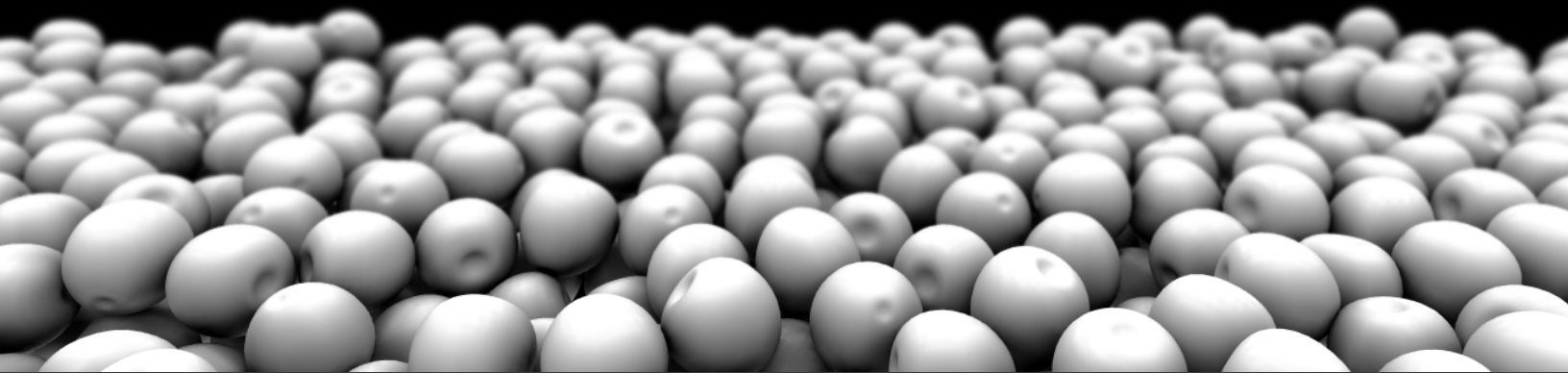


Per-Vertex
Ambient Occlusion &
Indirect Lighting
Generator



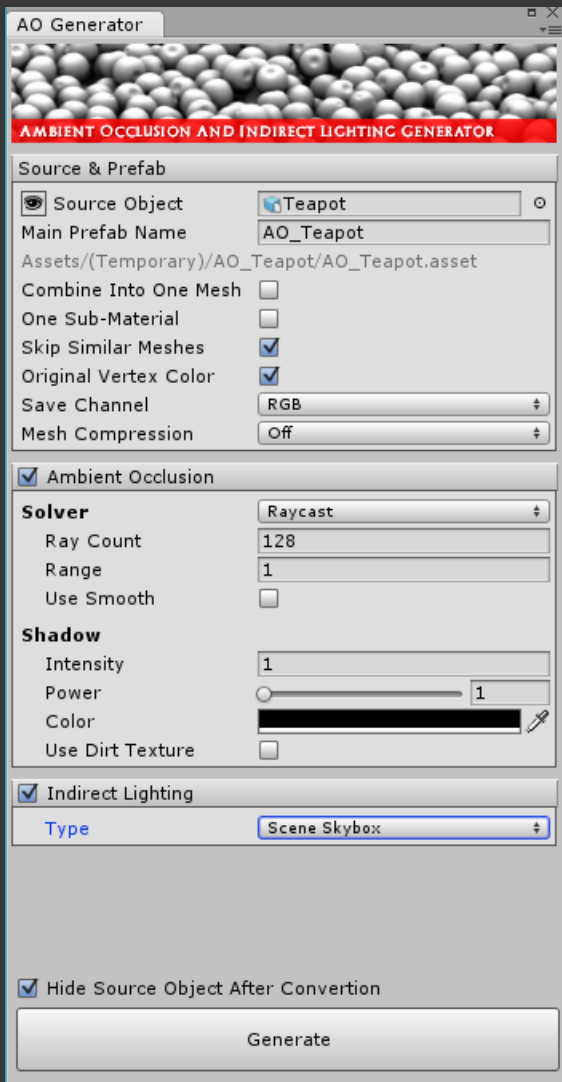
Thanks for purchasing **Per-Vertex Ambient Occlusion and Indirect Lighting Generator** package.

Please consider writing a review or just rate the [asset](#)



For any question or help use [forum](#)

Editor tool

Menu/Window/VacuumShaders/Per-Vertex Ambient Occlusion Generator



Source & Prefab tab

- **Source Object** – Gameobject for which is generating ambient occlusion (AO) and indirect Lighting (IL). Source object can be a single or contain hierarchy (children objects), AO and IL will be generated for all of them.
-   On/Off icon controls visibility of the source object. It may be helpful for making source object visible/invisible after conversion.
- **Main Prefab Name** – After generating AO and/or IL will be created prefab with these name and instantiated in the scene. Prefab is created in *Assets/(Temporary)* folder. Prefab will be copy of source object but with new meshes and VertexColor shader materials (for displaying per-vertex baked data).

- Prefab generated folder *Assets/(Temporary)* is a working folder. During generating process some files can be created and deleted by editor. After achieving desired AO and/or IL results prefab and mesh assets can be moved to any project folder.
- **Combine Into One Mesh** – If source object contains several meshes than they can be combined into one mesh. One mesh asset will be created and main prefab will use it.
If this option is enabled main prefab will have no hierarchy.
Option is available only if sum of all mesh vertices in the source object hierarchy is ≤ 65.000
Combined mesh will have one sub-mesh only.
- **Skip Similar Meshes** – If *Source Object* contains multiple same meshes (mesh name defines *similarity*) there is no need to convert and save them all. It's enough to convert only one of them and in final prefab it will be used. But in some cases it is necessary to convert all meshes in the *Source Object* – in this case this option must be turned off.
- **One Sub Material** – Removes multiple material IDs from resultant mesh. Mesh with one sub mesh requires one draw call.
- **Original Vertex Color** – Generated AO and IL can be combined with source mesh's vertex color.
- **Save Channel** – Generated AO and IL values can be saved in any channel of vertex color.
Data saved inside R, G, B, or Alpha is grayscale. Full colored data must be saved inside RGB.
- **Mesh Compression** - Compressing meshes saves space in the built game, but more compression introduces more artifacts in vertex data (including color).

Ambient Occlusion tab

For generating AO tool contains two solvers: Raycast and Occlusion Render.

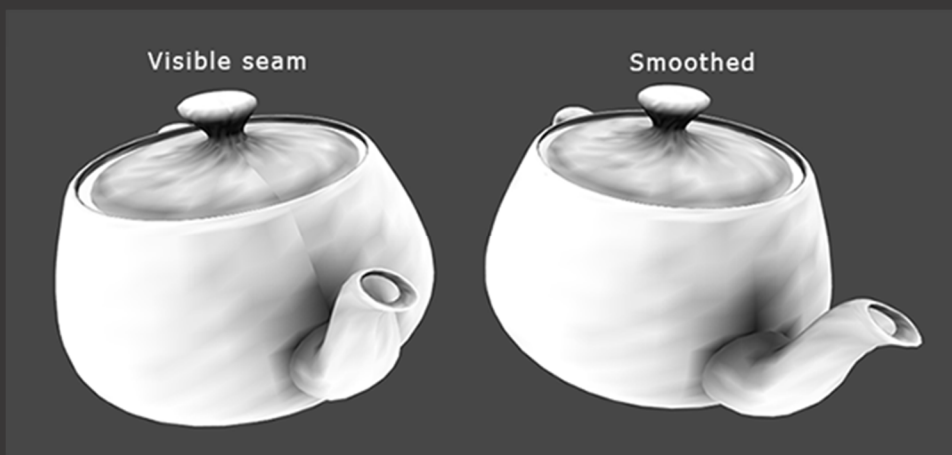
Raycast

Solver	Raycast
Ray Count	128
Range	1
Use Smooth	<input checked="" type="checkbox"/>
Radius	0

For AO calculating *throws* multiple rays from each mesh vertex.

Raycast solver can run on any platform and device, is multithreaded and produces physically correct AO result.

- **Ray Count** – Number of rays per vertex *thrown* for calculating AO. High value produces better results.
- **Range** – Ray length.
- **Use Smooth** – AO is calculated for each vertex individually and result is different for each one. If mesh has several vertices in the same position, resultant mesh will have visible seams (artifacts). This option fixes that, but at additional pass price.
- **Radius** – Smoothing radius. Mostly value of 0(zero) will fix visible seams.

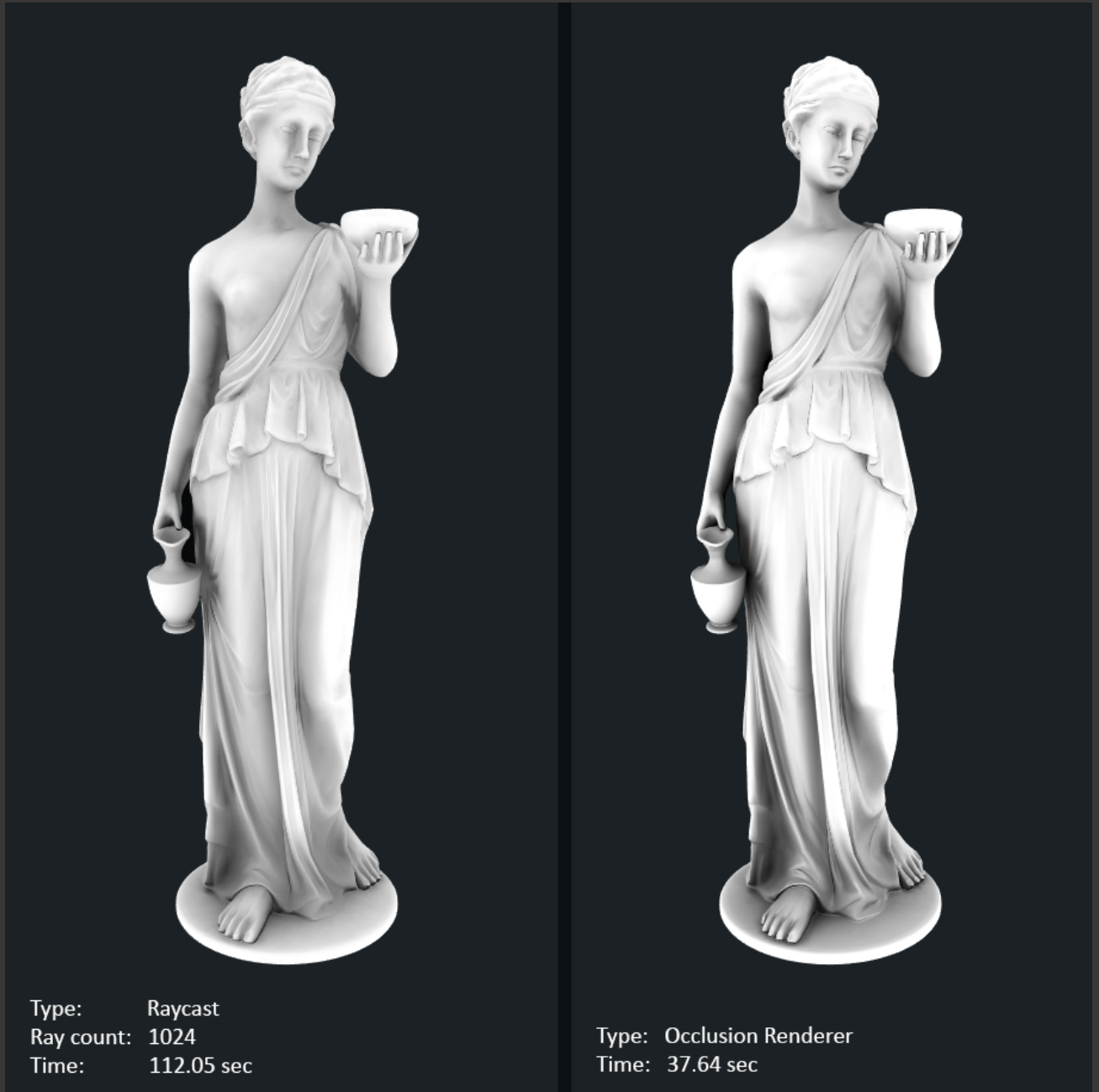


Occlusion Render

Solver	Occlusion Render
Occlusion Distance	1000
Layer Mask	Everything

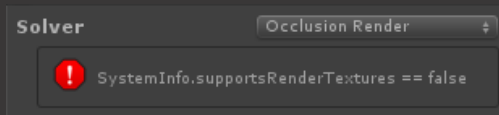
For AO calculating object is rendered from each vertex position and then determined average value.

Occlusion Render solver produces smoothest result (as infinity ray count in Raycast solver), is faster on high poly meshes, but requires device with RenderTexture support.



- **Occlusion Distance** – Visibility distance of vertex.
- **Layer mask** – Defines object layers that interact with AO solver. If it is set to 'Nothing' – nothing will be rendered, even selected object.

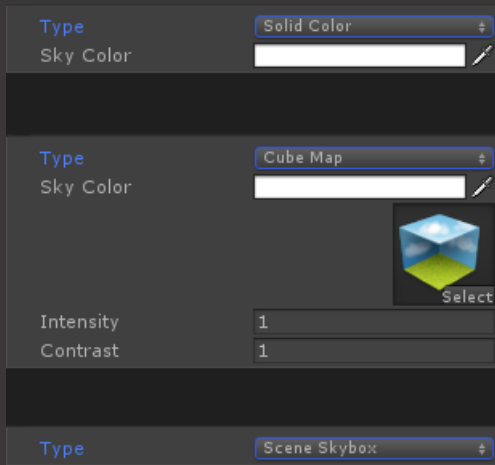
If target device does not support RenderTextures instead of Occlusion Render solver will be used Raycast solver with its default values (128 rays per vertex).



Editor window will also inform about not supporting of RenderTextures.

Indirect Lighting tab

Available 3 types of indirect lighting calculation:



- Solid Color – Simple and fastest
- Cube Map – IL is calculated from cubemap texture.
- Scene Skybox – IL is calculated from scene active skybox. Supports all type of skybox shaders, but requires RenderTextures support.