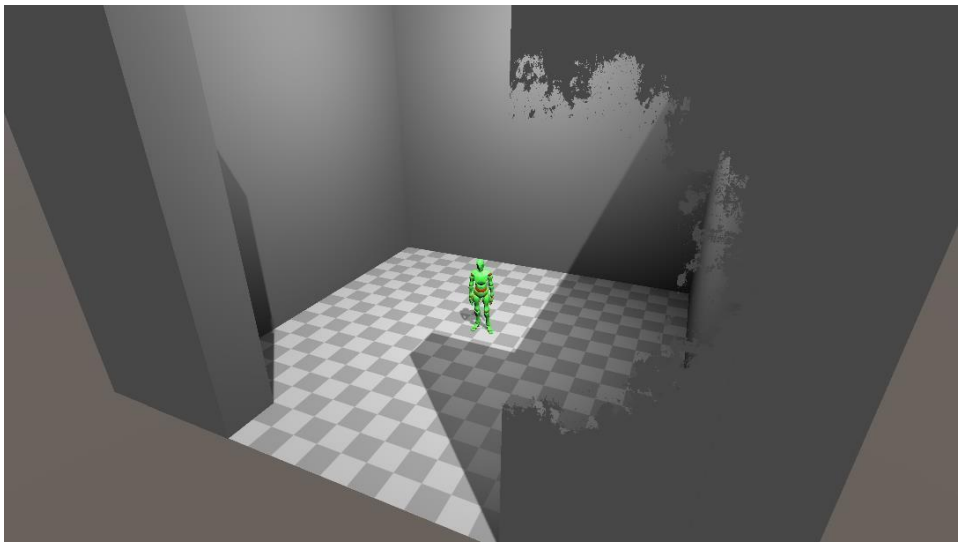
PixelPulse – Documentation



Dynamic Occlusion Cutout System (DOCS)

Contents

Script

This script is used to **calculate** and **update** the Dynamic Occlusion Cutout System (DOCS) **Materials**.

It has the following parameters:

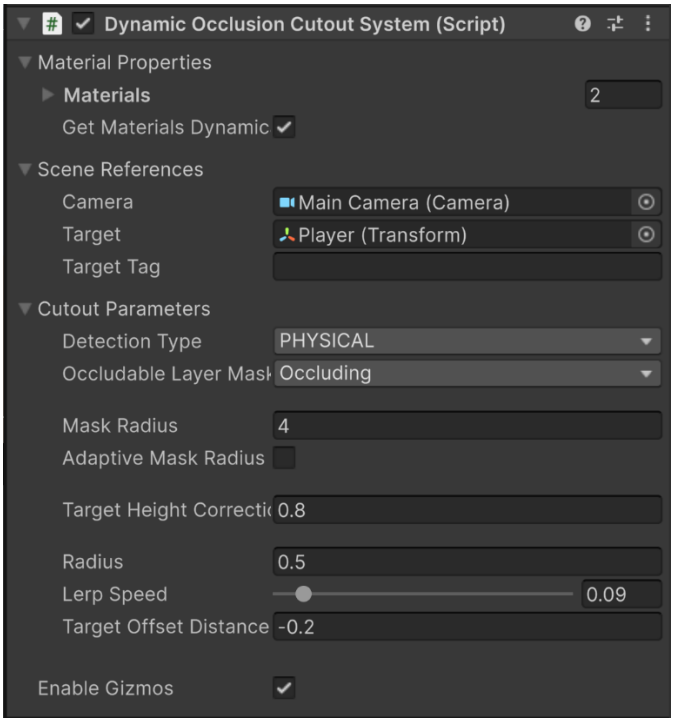


Fig. 1 – DOS script in Unity's Inspector view

Material Properties

- **Materials** are the references to the DOCS materials to update.
- **Get Materials Dynamically** is used to automatically get materials to update in runtime from the scene (has a performance cost ⚠️).

Scene References

- **Camera** is the Camera on which the occlusion will be active
- **Target** is the aim of the camera; it should stand behind the objects that will be cut out.
- **Target Tag** (optional string) is the Tag the script will look for in Awake(), leave it empty if the target is manually assigned.

Cutout Parameters

Detection Type & Layer Mask

- **Detection Type** offers 3 modes of cutout activation.

  - **Physical** will deliver the most precise results but it requires the use of objects that have physical components on them (Collider or Trigger).

  - **Graphical** mode does not rely on physical objects or colliders.
    Instead, it uses a computed capsule to determine which renderers should be considered for visual cutout updates.

    ⚠️ In scenes with a very large number of renderers, Graphical mode may increase per-frame CPU usage due to renderer iteration.
    Using the Occludable Layer Mask or explicitly limiting which renderers are considered greatly improves performance.

  - **Constant** is the less costly mode but will deliver the less precise results as well.
    This mode will activate the DOCS system constantly with no camera-target check, resulting in a radial cutout around the target (meaning also behind).

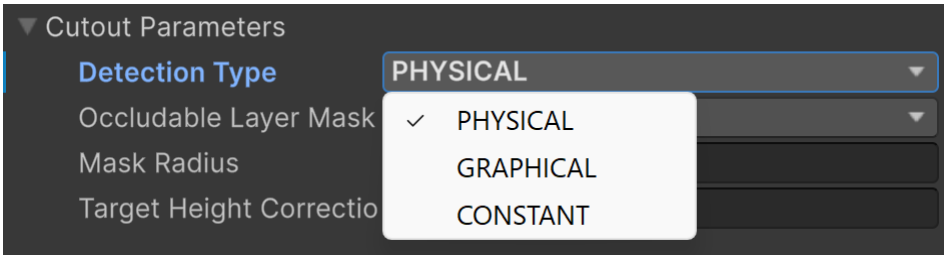


Fig. 2 – DOCS script displaying the Detection Types in Cutout Parameters

Comparison Table

Feature	Physical	Graphical	Constant
Uses colliders	✅	❌	❌
Uses renderers	❌	✅	❌
Occlusion-aware	✅	✅	❌
Physics-based	✅	❌	❌
Adaptive radius support	✅	✅	✅
Performance cost	Low	Medium–High	Very Low
Visual-only scenes	❌	✅	✅
Deterministic output	⚠️ *	✅	✅
Unity Terrain (foliage & trees)	❌	✅	❌

**Dependent of Unity's Physics parameters configuration*

- **Occludable Layer Mask** gives the option to filter out which objects are considered as potential occluders by the cutout system. When set, only renderers (Graphical mode) or colliders (Physical mode) belonging to the selected layers are evaluated.
  Using this mask:
    - Reduces unnecessary detection checks
    - Lowers memory usage when caching renderers
    - Improves runtime performance in scenes with many objects
    - Prevents unintended objects (VFX, characters, UI proxies, etc.) from triggering cutouts

  ⚠️ It is **<u>strongly recommended</u>** to configure this mask explicitly instead of using *Everything*, especially in **Graphical** mode. ⚠️

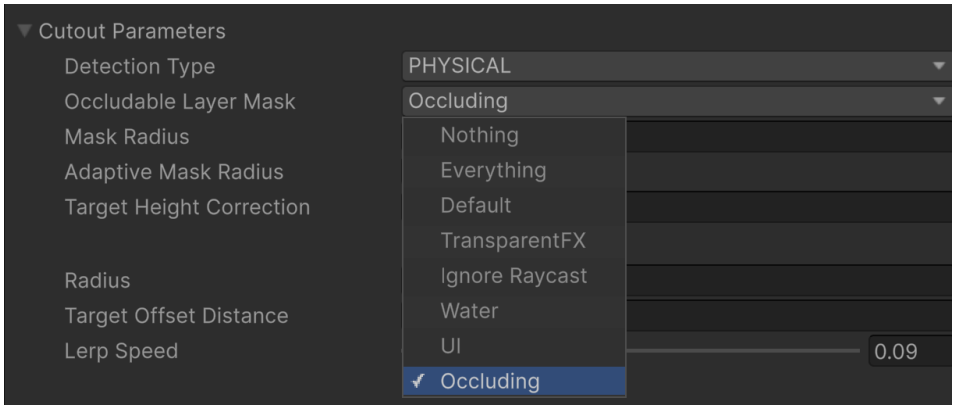


Fig. 3 - DOCS script displaying the Occludable Layer Mask in Cutout Parameters

Mask Radius & Adaptive Settings

- **Mask Radius** is the radius of the cutout mask.

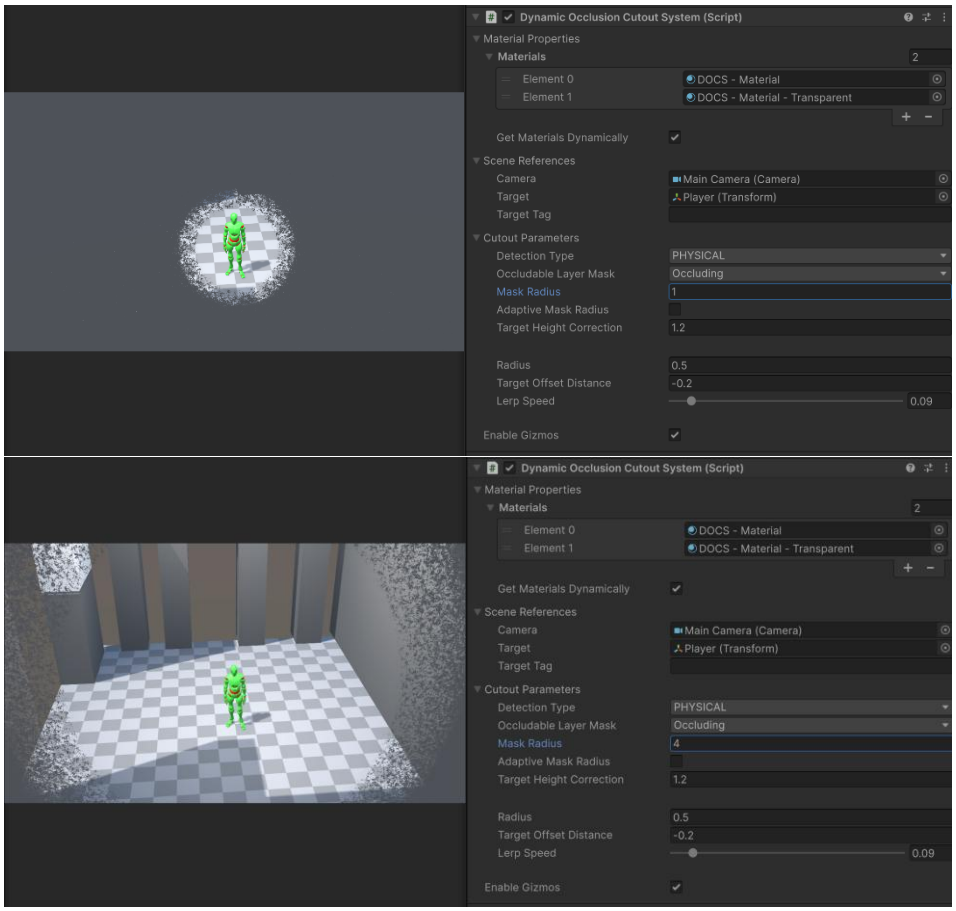


Fig. 4 – Mask Radius set at 1.0f and 4.0f

- **Adaptive Mask Radius** the mask radius automatically adjusts to maintain a visually consistent size relative to the camera view.
    - For perspective cameras, the radius scales based on the camera–target distance (zoom in/out) and the camera's Field of View (FOV).
    - For orthographic cameras, the radius scales according to the camera's orthographic size, independent of distance.

  The resulting radius is clamped between Minimum Mask Radius and Maximum Mask Radius to ensure stable and predictable behavior.

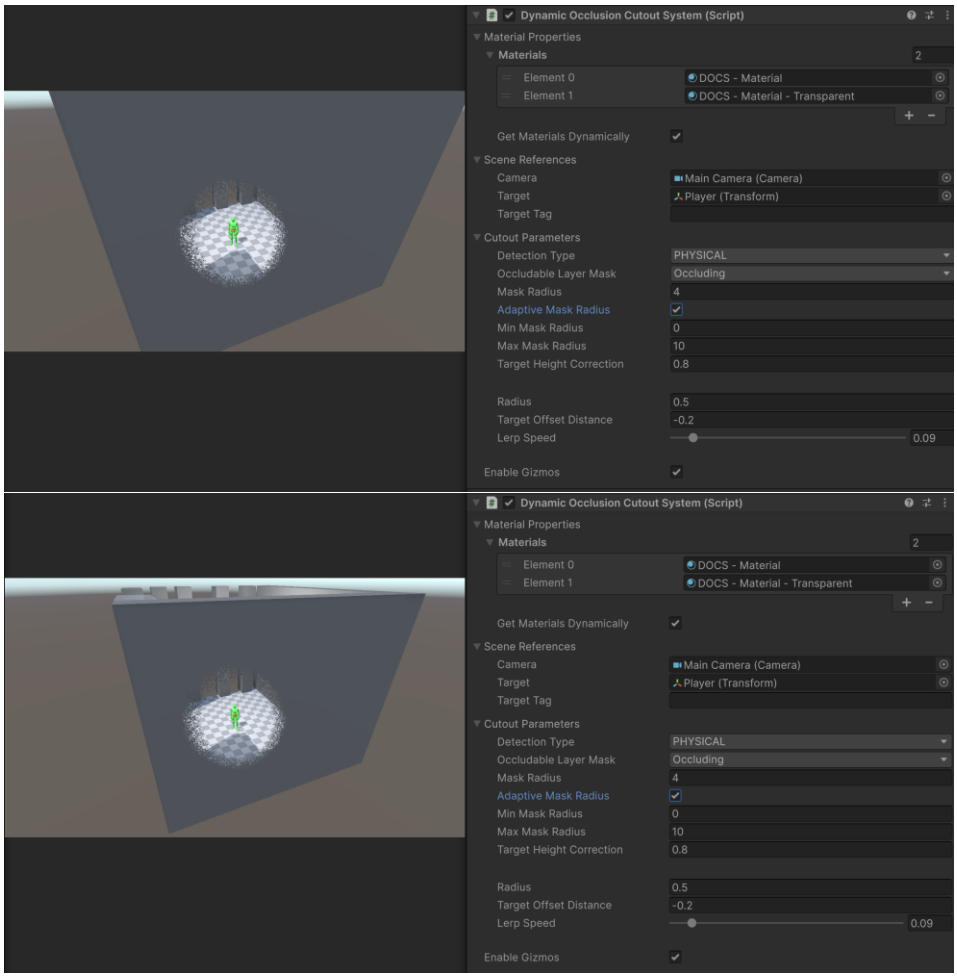


Fig. 5 – Display of adaptive mask radius at different distances form target (perspective camera)

Detection Behavior

- **Target Height Correction** gives the possibility to change the position of the cutout on the Y Axis.

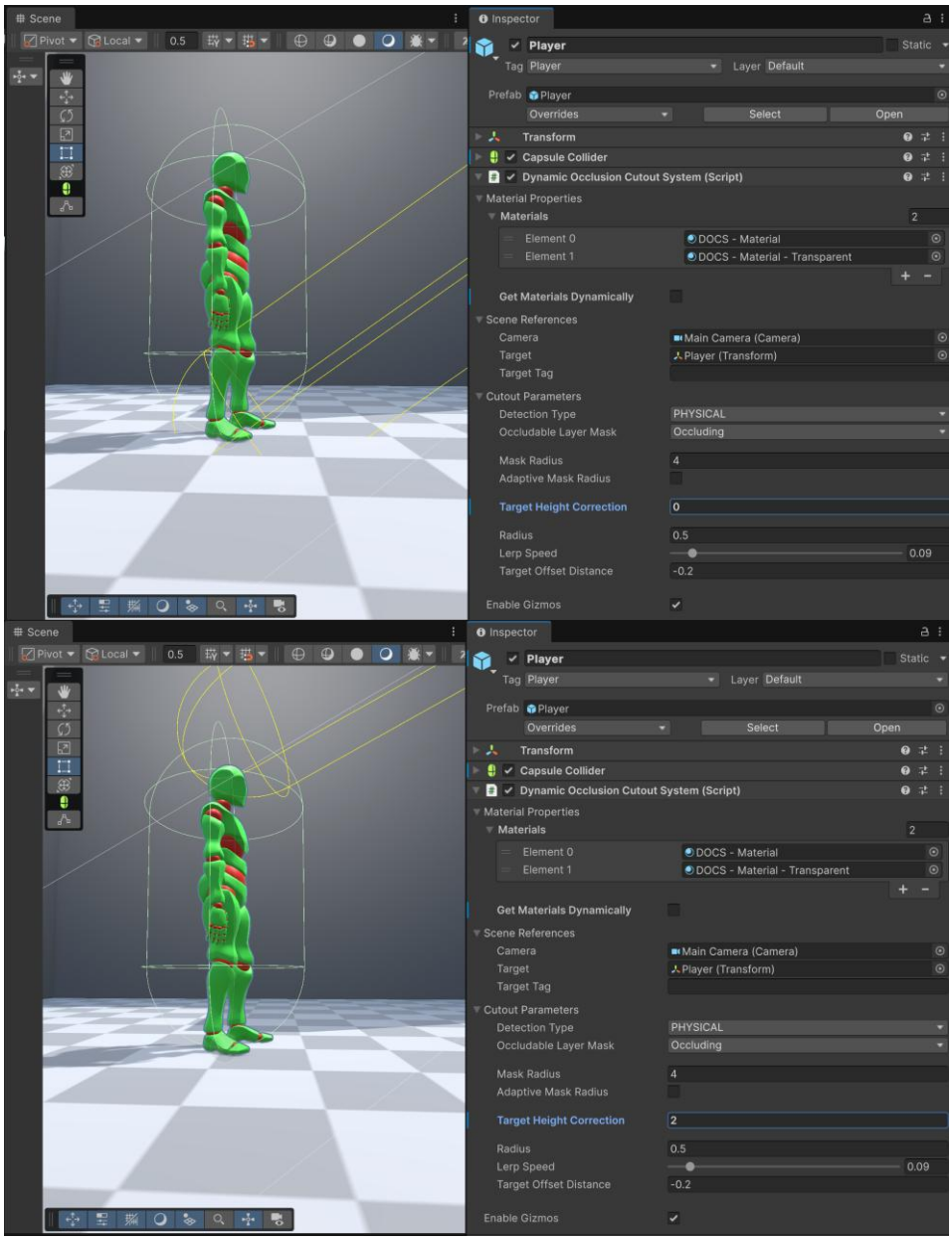


Fig. 6 - Target Height Correction set at 0.0f and 2.0f (exaggerated)

- **Radius (Physical Mode)** will change the size of the SphereCast used for object detection.

  The contact point of the SphereCast will show a red WireSphere and object name (Requires enabling Gizmos)

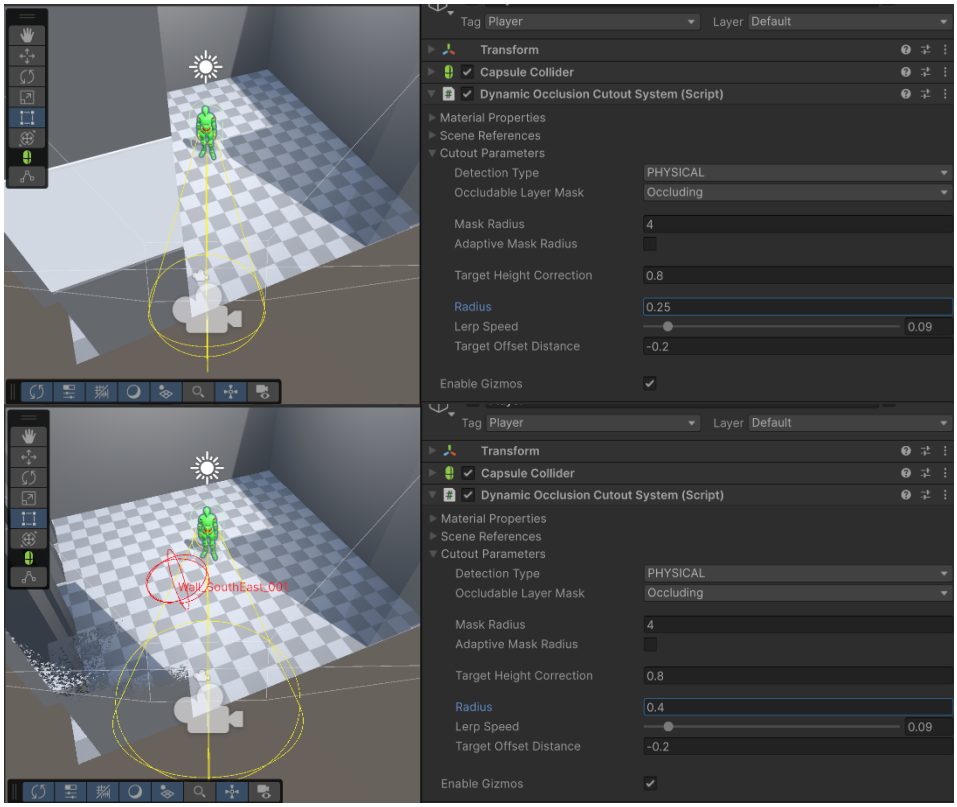


Fig. 7 - Radius set at 0.25f and 0.4f – Detection of Object in the capsule's area

- **Graphical Mode** will change the way the object detection is made.

  Detecting objects will display their renderers in green, their center position on a green label and a red WireSphere and label *"Graphical Hit"* when the object hits the detection capsule. Renderers that are cached but not detected will be rendered in orange.

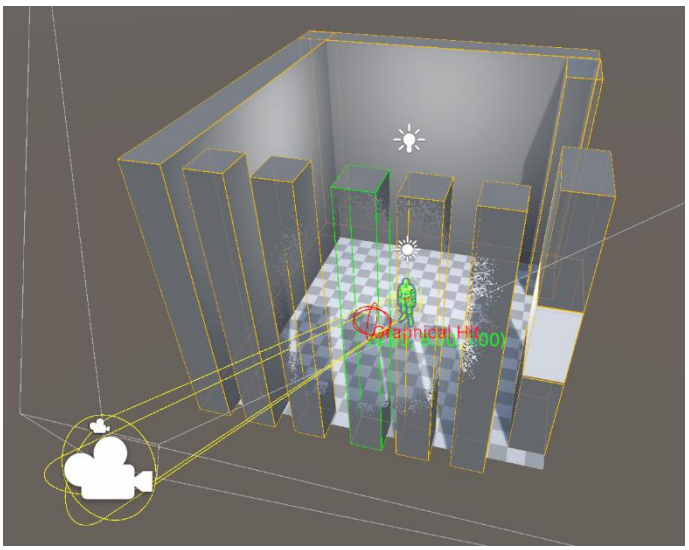


Fig. 8 - Detection of Objects in Graphical mode, orange renderers, green when hit

- **Lerp Speed** will define the speed at which the cutout will update.
  0.01 will update slowly - 1 will be instantaneous.

- **Target Offset Distance** gives the possibility to change the starting point of the Raycast. This parameter can prevent the system from stopping if the target (player) is too close to a wall.

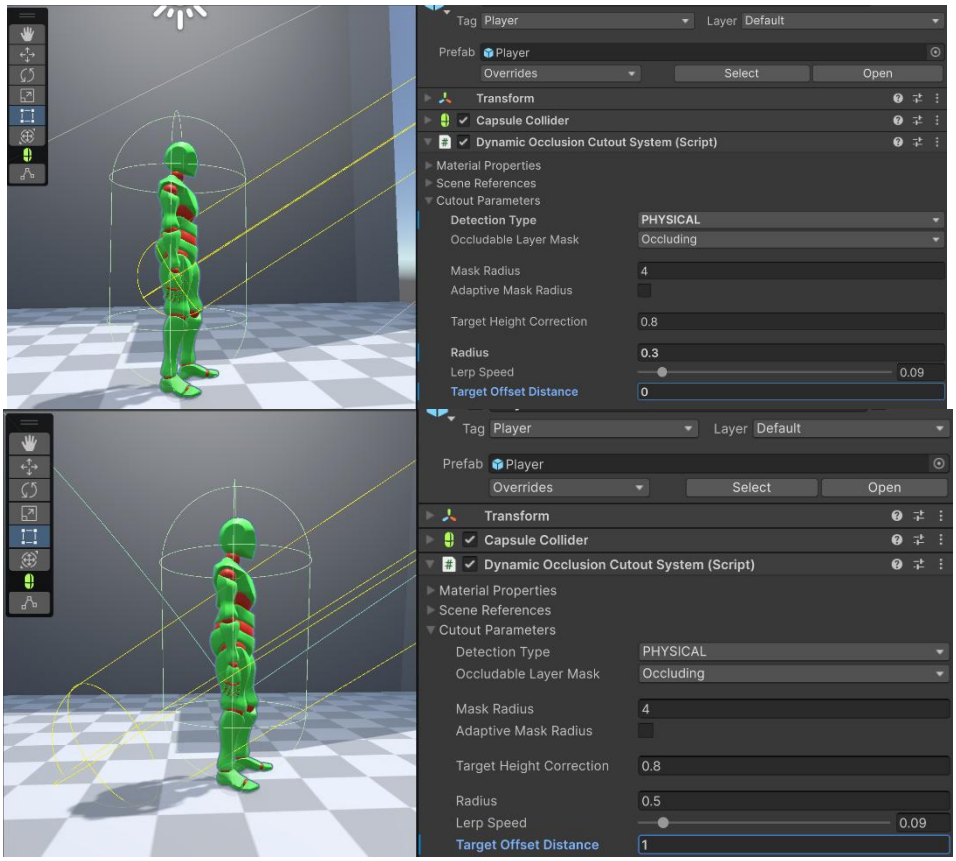
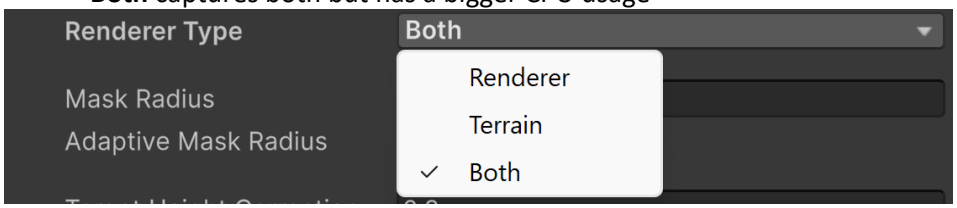


Fig. 9 – Target Offset Distance set to 0.0f and -1.0f (exaggerated)

- **Render Type** gives the possibility to change what the Graphical detection captures.
  - **Renderer** only captures renderers placed in the scene(s)
  - **Terrain** only captures Terrain objects (trees)
  - **Both** captures both but has a bigger CPU usage



Shader

The shader enables the creation of DOCS Materials.

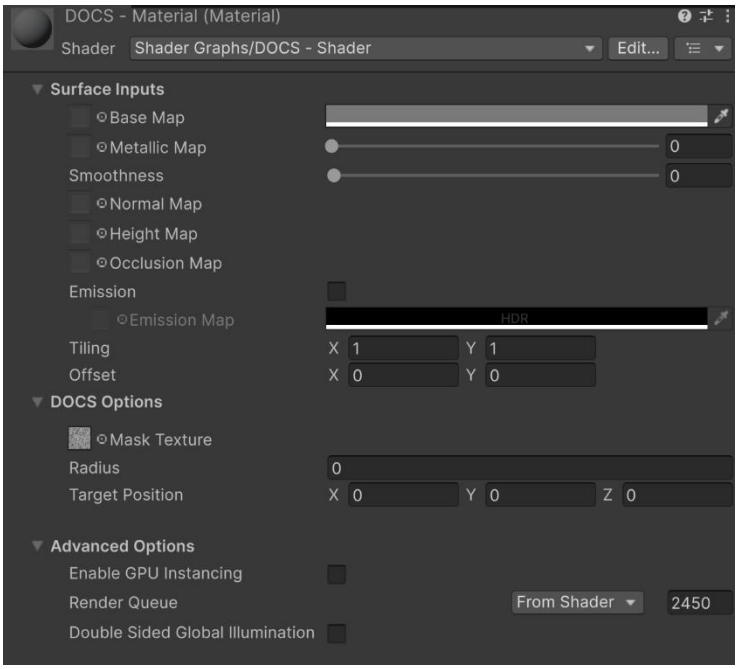


Fig. 10 – DOCS Material displayed in inspector

This material has the original surface shader canals:

- **Base Map** & **Base Colour**
- **Metallic Map** & **Metalness**
- **Smoothness**
- **Noraml Map & Normal Strenght**
- **Height Map & Height Strenght**
- **Occlusion Map / Occlusion Strenght**
- **Emission Map** & **Emissive Colour**
- **Tiling & Offset**

And the DOCS parameters:

- **Target Position** is updated by the script (can be changed manually for debugging in editor)
- **Radius** is updated by the script (can be changed manually for debugging in editor)
- **Mask Texture** is the texture used for the occlusion cutout.

⚠ ⚠ ⚠

MaskTextures use Triplanar UVs, this setting can be changed in the shader but first be sure YOU KNOW WHAT YOU ARE DOING if you want to change its functionality

⚠ ⚠ ⚠

How to use it?

To use this system:

- **Create a material** derived from the **DOCS – Shader**

Fig. 11 – DOCS Shader displayed in Project Folder

- **Right click** on the shader > **Create** > **Material**
  In *Assets > Dynamic Occlusion Cutout System (DOCS) > Shader*

- **Place the Material** on the GameObjects you want to see through (ex. Walls)
  Inspector > Mesh Renderer > Materials > Element (n)

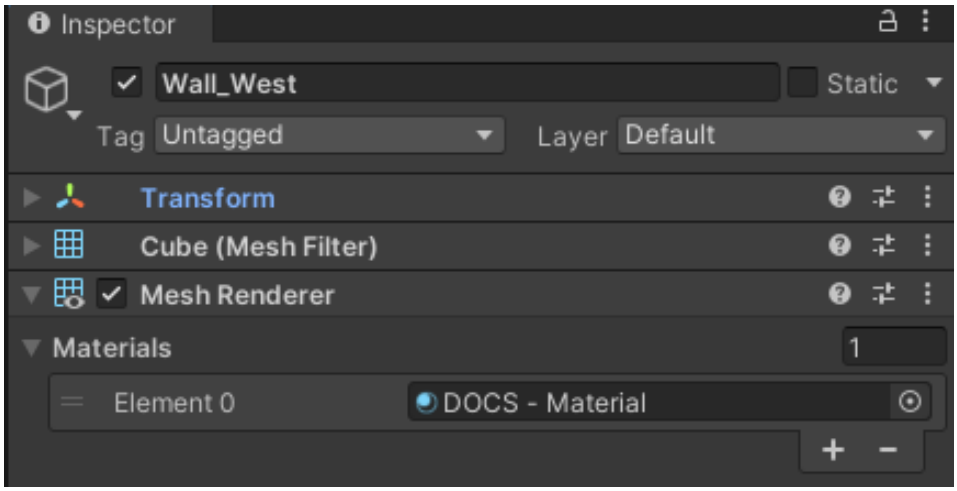


Fig. 12 – Wall object with DOCS – Material set in its Mesh Renderer

- **Assign** the **Dynamic Occlusion Cutout System - Script** on **<u>any object</u>** in your scene
  The most relevant object to hold the system is generally the character or camera (any object can hold it, it should still work).

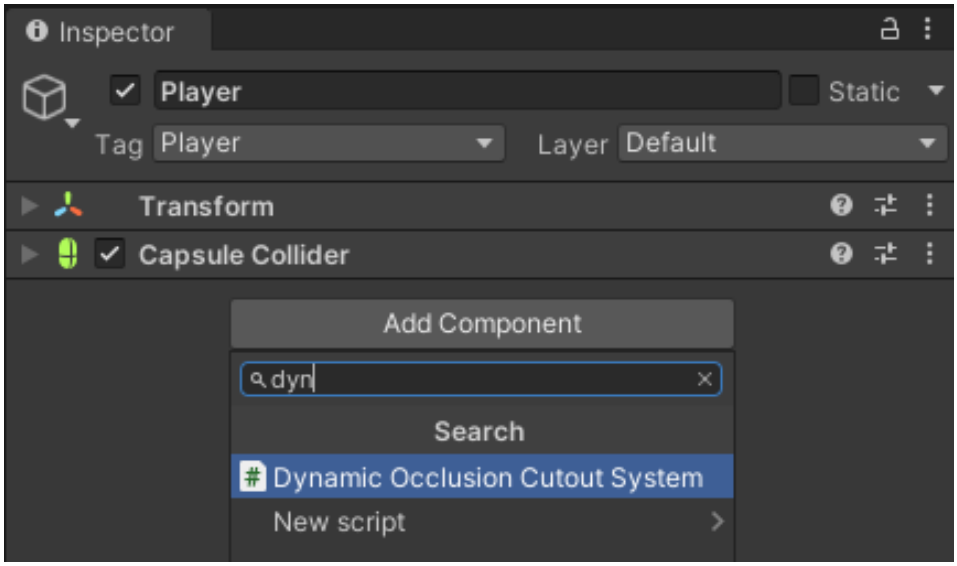


Fig. 13 – Addition of DOCS script to a scene object (Add Component button)

- **Assign** the newly created **material** in the script for it to be updated
  Get Materials Dynamically will prevent this step but comes with a CPU cost!

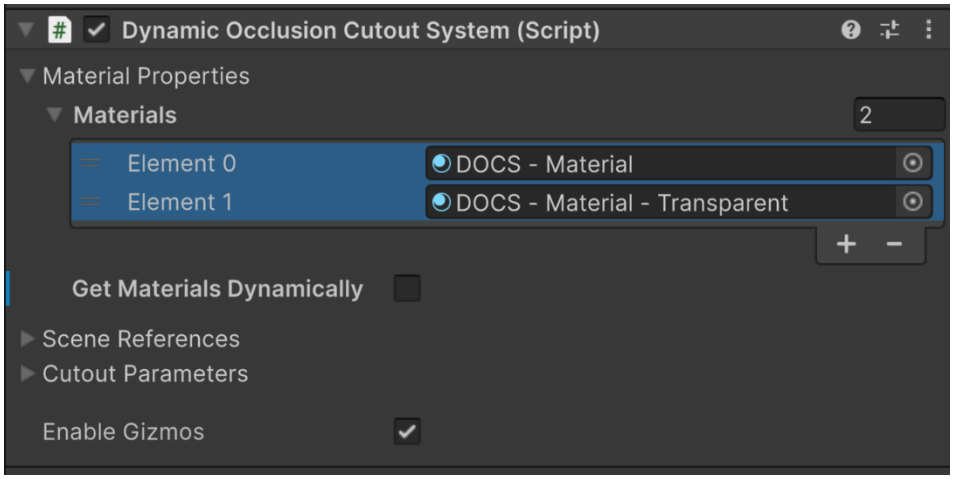


Fig. 14 – Reference of the DOCS materials in the script (drag and drop from the folder)

- **Enter** the **Target Tag** *(optional)* or **Assign** the **Target**
  Entering a tag will automatically search for the target with that tag in the scene

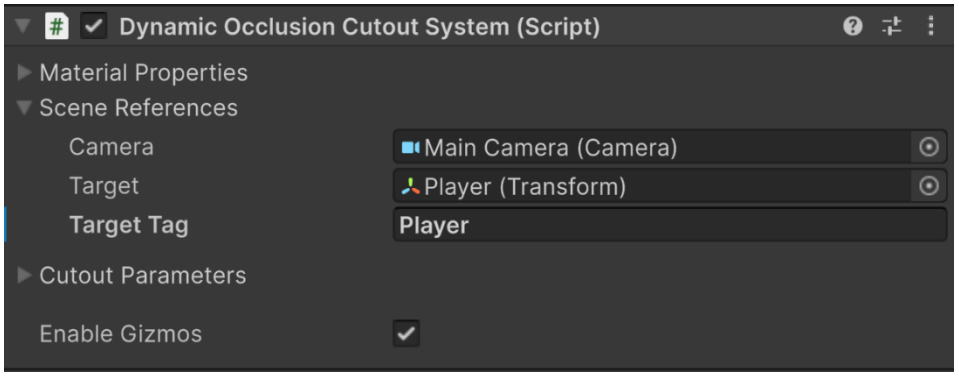


Fig. 15 – Display of the Target Tag parameter set with "Player" as a tag

- **Select** the **Detection Type** you want to use
  Refer to the previous documentation and comparison table

- *(Optional)* **Set** the **Occlusable Layer Mask** in the script and on objects that must be occluded
  ⚠️ *It is **Highly recommended to set** it for optimization purposes!* ⚠️

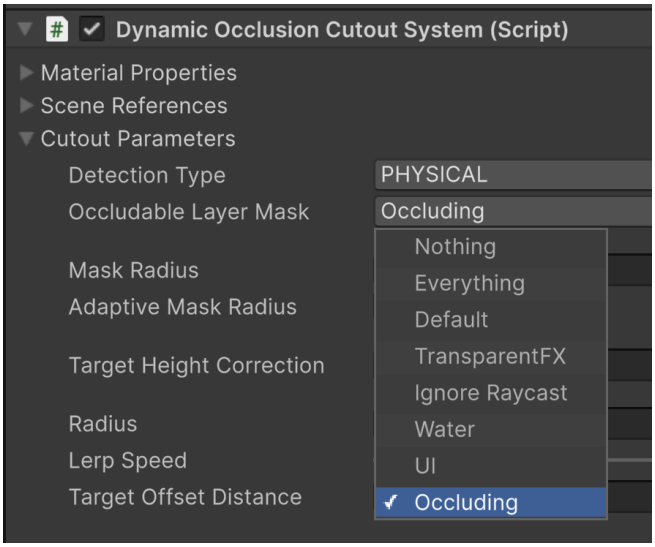


Fig. 15 – Display of the Target Tag parameter set with "Player" as a tag

- To set a layer, go to the inspector and Add a Layer if not already created for this purpose *(Inspector > Layer > Add Layer)*

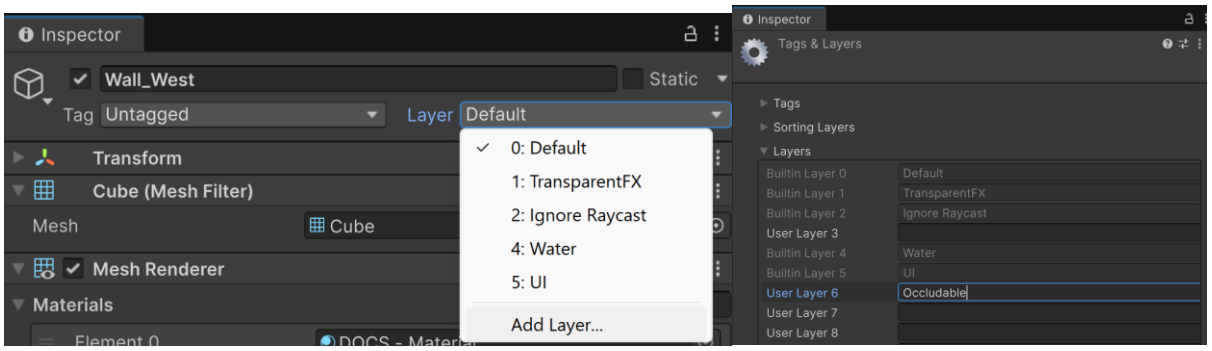


Fig. 16 – Unity Inspector showing the addition of a Layer

- **Set** the **objects** to be cut out **to that layer** *(In scene objects or prefabs)*

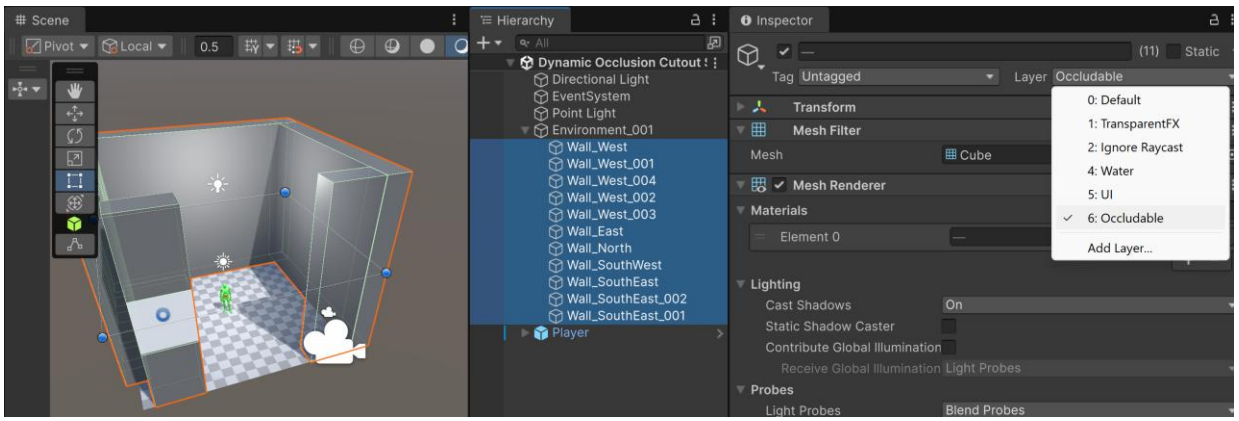
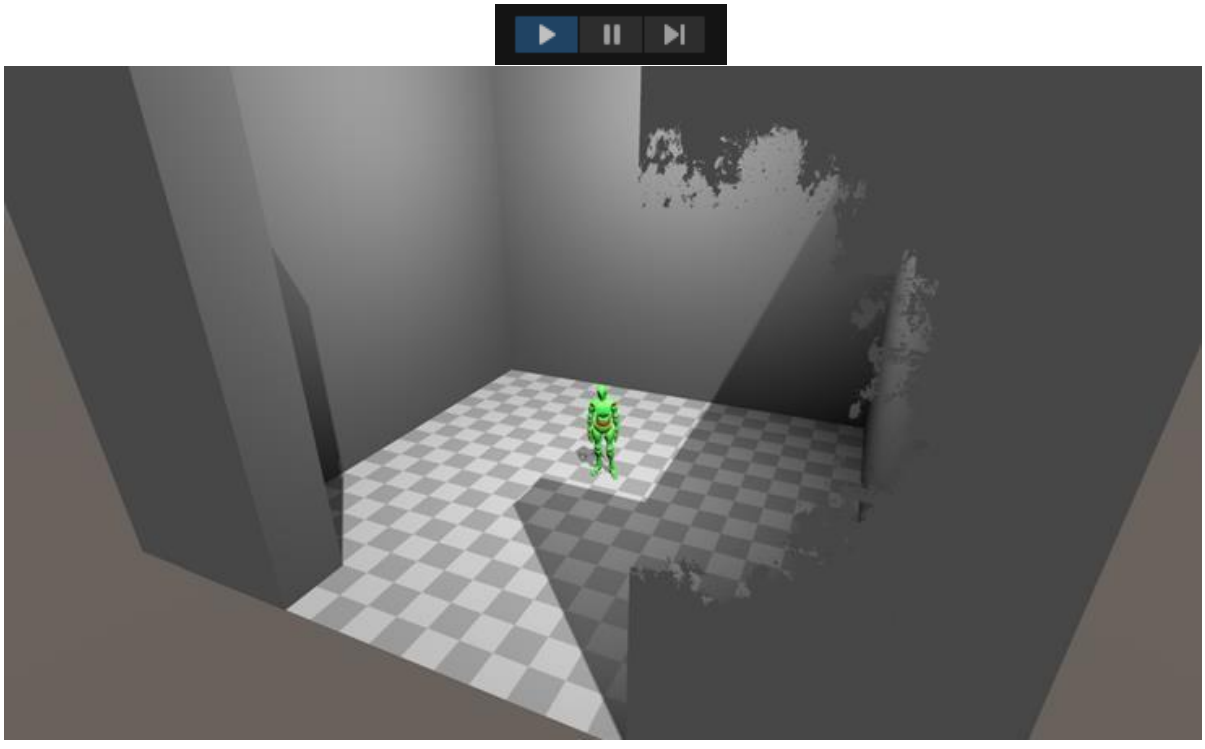


Fig. 17 – Unity Hierarchy with selection of Wall object and setting of their Layer to "Occludable"

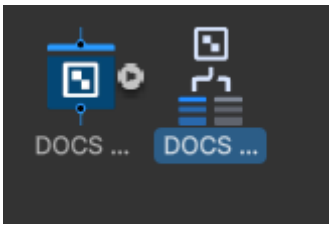
The following settings are project dependent, refer to the documentation above for the details of each cutout setting.

- Finally click **Play** and **Enjoy** 😊

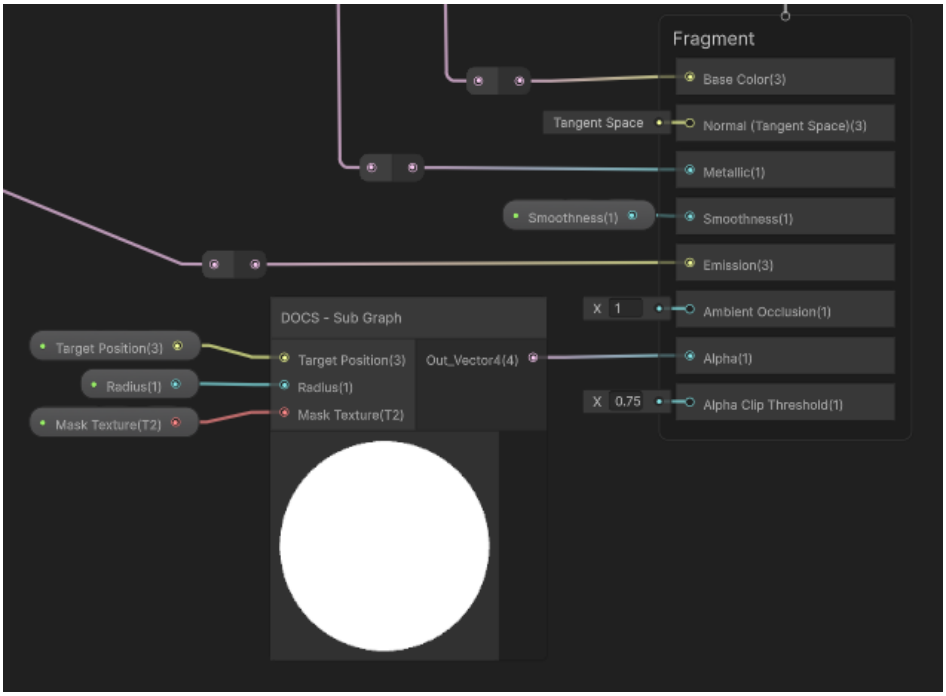


Integration with existing materials

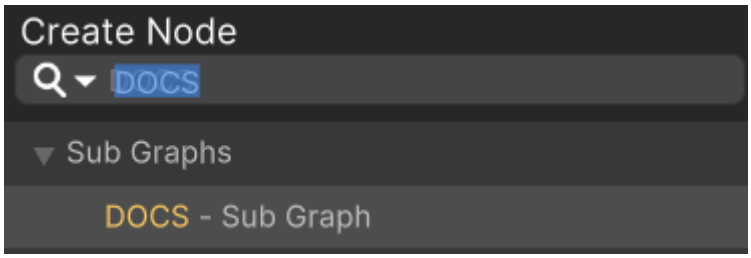
To facilitate the integration with already existing materials the version 1.0.2 introduced a Sub-Graph.



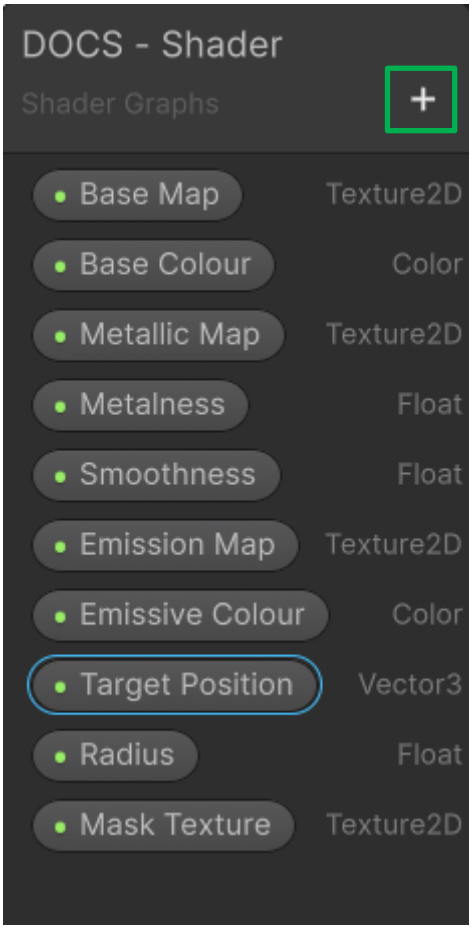
To use it, it must be integrated into the shader used for walls/elements you want to cut out



To do so, create a new node in the ShaderGraph (Right click) and search for "**DOCS**" and link it to the alpha channel.



Then, declare a Vector3 called “**Target Position**”, a float called “**Radius**” and link them to the node.

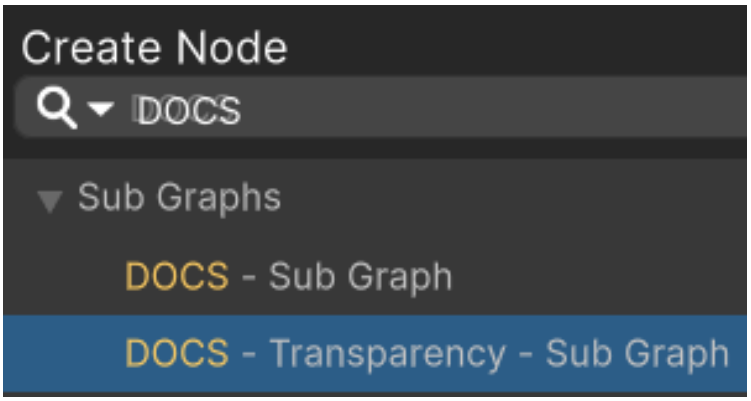


Optionally, you can declare and change the “**Mask Texture**” for a different style of cut out.

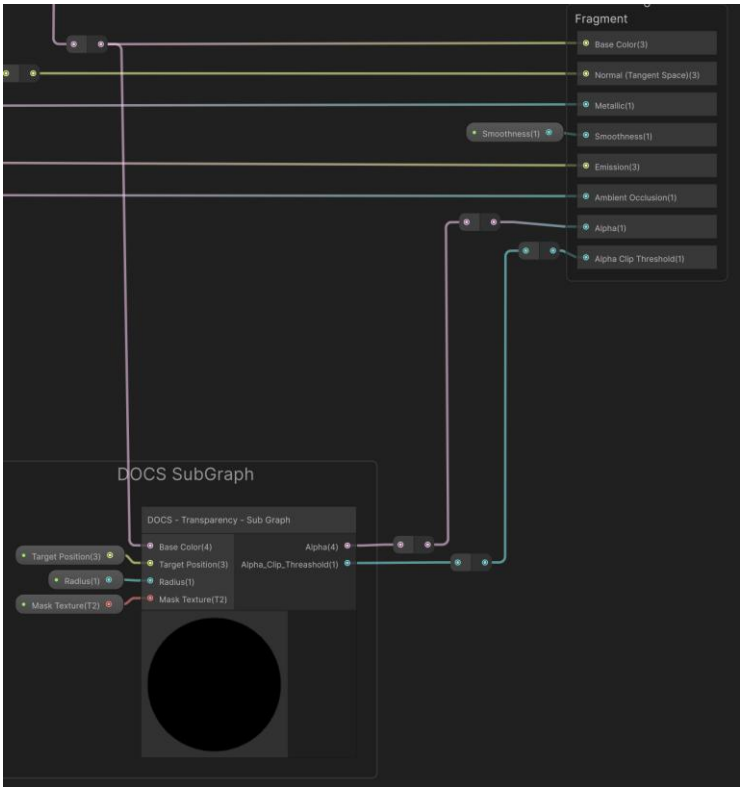
Once setup, the modified materials should behave like the DOCS materials.

Integration with transparent materials

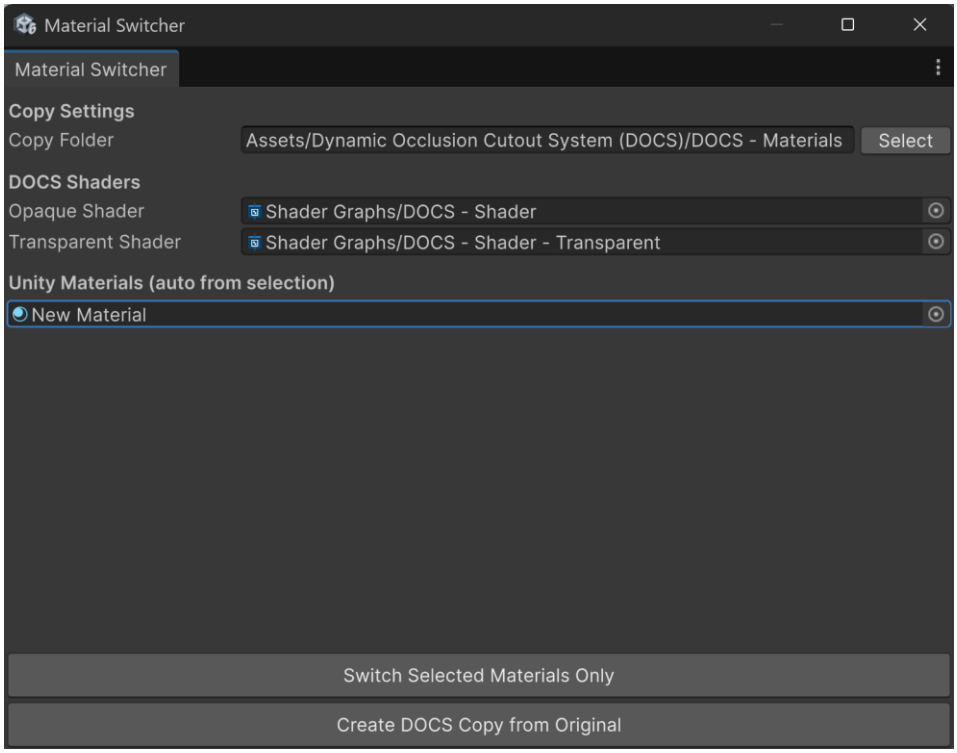
To integrate it into transparent materials, the procedure is the same; the only difference is that the DOCS node must be the *“DOCS – Transparency – SubGraph”*.



The Base Color must be linked to the node, and the alpha threshold output must be linked to the Fragment shader *(Master Node)*.

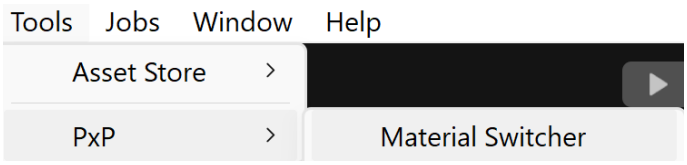


Material Switcher tool



The material switcher tool enables automatic material update for **Unity's Default Materials**.

To open the tool's window, go to the tab *Tool > PxP > Material Switcher*.



- **Copy Folder** is the folder where materials can be copied, you can enter the file path manually or use the "**Select"** button to define one using the file system.
- **Opaque Shader** is the reference to the "*Shader Graphs/DOCS - Shader*"
- **Transparent Shader** is the reference to the "*Shader Graphs/DOCS - Shader - Transparent*"

  These fields are to be set with the shaders (opaque and transparent) that the materials are going to be switched to (can be user defined / custom shaders).

- **Unity Materials** *(auto from selection)* is a list of materials that is set by user selection.

  If a material is selected, it should appear in that list. Selection of Folders will recursively get all materials that meet the requirements (being a Unity's Default Shader).

Two buttons are displayed below the tool:

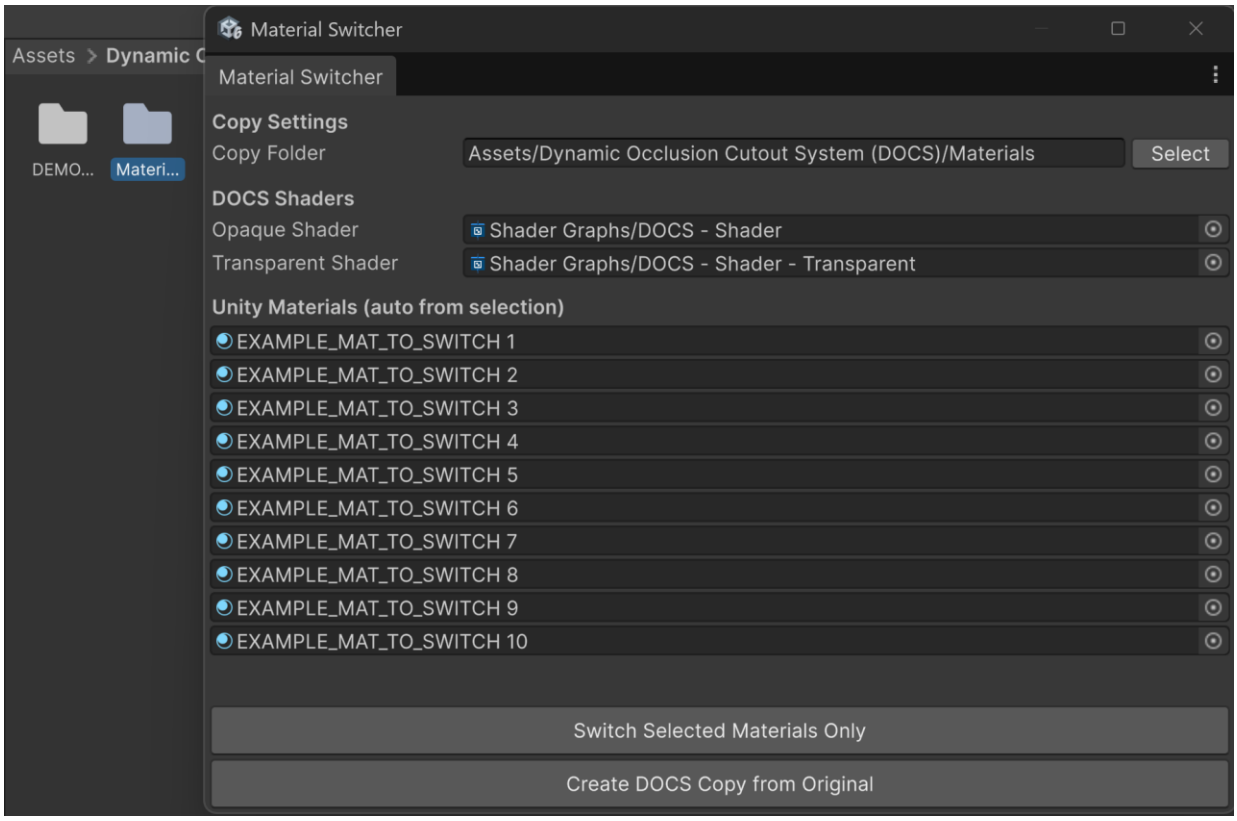
- **Switch Selected Materials Only** will update the materials from their current version to the corresponding DOCS shader (depending on whether it is transparent or not).

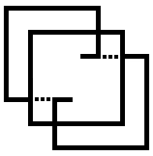
  ⚠️ This operation will **Override** the selected materials ⚠️

- **Create DOCS Copy from Original** will create a new version of the targeted materials at the Copy Folder location.

  ⚠️ Leaving the Copy Folder blank will disable the button ⚠️

When all parameters are set the result should be the following:





The PixelPulse team thanks you for purchasing

Dynamic Occlusion Cutout System - DOCS

and wishes you all the best for your project.

For any issues regarding the DOCS package, address a mail to PixelPulse directly or fill in a contact form on our website