Mirage

Setup

The whole package is contained in the Mirage folder resulting from the import but you can move it wherever you want.

What is an impostor

An impostor or imposter is a dynamically rendered billboard texture map used to give an illusion of 3D geometry in the distance.

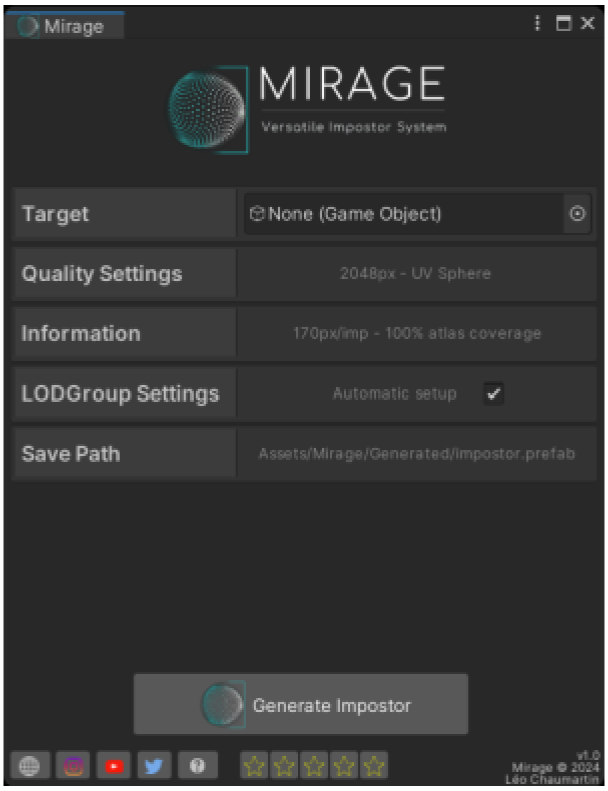
In the Mirage context, you can have as many objects as you want in a single impostor, and all the the points of view are stored into an abedo, normal and metallic maps. Smoothness or occlusion are not baked into textures yet but are still customizable on post baking options.

How to bake your first impostor

`Mirage` consists in a single editor window. You can open it via `Window->Mirage->Impostors`. For a simple one-click impostor approach, drag and drop the common ancestor of all the objects you want to bake into target area and click on the `Generate Impostor` button.

You can also customize a lot of settings for a more advanced usage. There are 5 category foldouts that can expanded or hidden by clicking on their title. Even when hidden, they still provide basic information or controls on the right of the title.

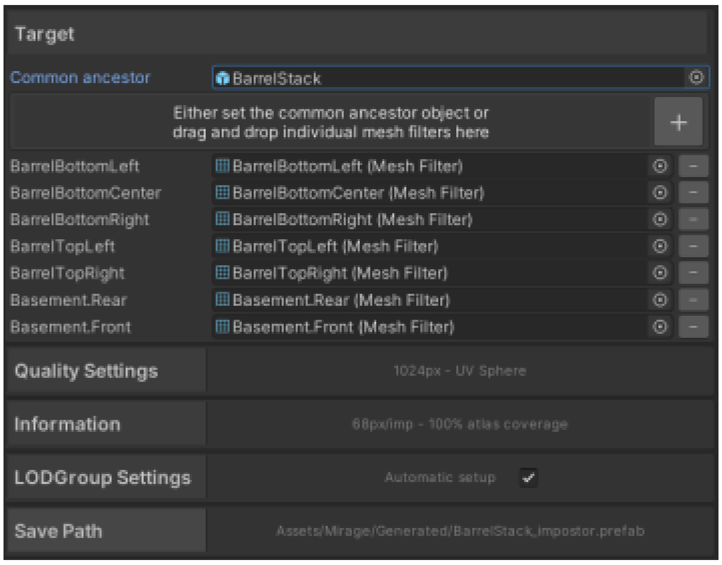
The following sections will provide detailed explaination of each category.



Target	None (Game Object)
Quality Settings	2048px - UV Sphere
Information	170px/imp - 100% atlas coverage
LODGroup Settings	Automatic setup ✓
Save Path	Assets/Mirage/Generated/impostor.prefab

Setup the target(s)

This category holds all the source object(s) information. Note that only one impostor can be baked at a time, so all the target will all be part of the same impostor. For a simple approach, drag and drop the common ancestor in the dedicated field. Note that if you want to bake only one object the common ancestor can be the object itself or its parent. When the common ancestor is set, Mirage will automatically detect all the mesh filters in its children, recursively. You can check which mesh filters are included by expanding the section as following. Null or invalid objects will be ignored.


Target

Common ancestor: BarrelStack

Either set the common ancestor object or
drag and drop individual mesh filters here

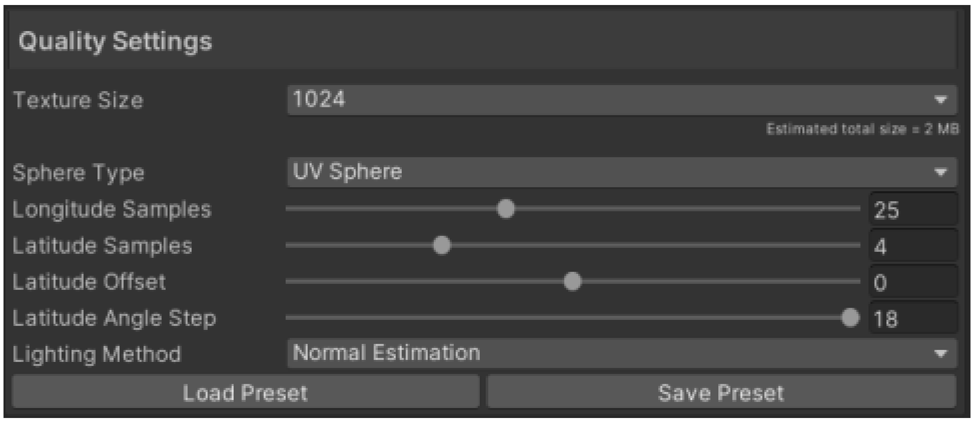
BarrelBottomLeft	BarrelBottomLeft (Mesh Filter)
BarrelBottomCenter	BarrelBottomCenter (Mesh Filter)
BarrelBottomRight	BarrelBottomRight (Mesh Filter)
BarrelTopLeft	BarrelTopLeft (Mesh Filter)
BarrelTopRight	BarrelTopRight (Mesh Filter)
Basement.Rear	Basement.Rear (Mesh Filter)
Basement.Front	Basement.Front (Mesh Filter)

Quality Settings	1024px - UV Sphere
Information	68px/imp - 100% atlas coverage
LODGroup Settings	Automatic setup ✓
Save Path	Assets/Mirage/Generated/BarrelStack_impostor.prefab

You can also exclude some meshes or add some manually by dropping them into the area. If you change the mesh filters manually, the common ancestor will be updated accordingly. Note that **a common ancestor is needed to bake an impostor**.

Quality Settings

This category is crucial and will have a direct impact on the visual quality of the impostor as it provides full control over the baking topology. This section should be paired with the `Information` category that provides real-time visualization for an excellent and fast control over the baking topology.



Quality Settings

Texture Size: 1024
Estimated total size = 2 MB
Sphere Type: UV Sphere
Longitude Samples: 25
Latitude Samples: 4
Latitude Offset: 0
Latitude Angle Step: 18
Lighting Method: Normal Estimation

Texture Size

As previously mentionned, Color, Normal and Metallic Maps are stored into square POT atlases. This field controls the resolution of these atlases, from 128x128 to 4096x4096. The `Estimated total size` label provides the size of the whole impostor prefab after baking (with DXT compression)

Sphere Type

- **UV sphere**: The default choice. The UV sphere is based on a fixed number of longitude and latitude samples. The coverage is denser at the poles.The UV Sphere is a solid choice in many use cases, with more versatility regarding partial sphere baking.
- **Pseudo-Fibonacci sphere**: at an experimental stage. Provides a homogeneous coverage. Not compatible with latitude interpolation in the impostor shader. Despite being homogeneous, the output impostor will look smooth near the equator, with more noticeable snaps or blur around the poles. Full sphere baking only.

Longitude Samples

UV Sphere only. Holds the number of longitudinal samples. The higher the better for smooth horizontal rotation, at a cost of higher POV number thus lower resolution/POV in the atlas. All samples are equally spread around the 360 degrees.

Latitude Samples

UV Sphere only. Holds half of the number of latitudinal samples (equator excluded). This means that a 0 value will just bake around the equator. 1 will bake an additionnal sample on each side of the equator, and so on.

The higher the better for smooth vertical rotation, at a cost of higher POV number thus lower resolution/POV in the atlas.

Latitude Offset

UV Sphere only. Shifts the equator sample. Useful for partial spheres.

Latitude Step

UV Sphere only. Holds angular offset between two latitudinal samples. A small value will increase smoothness of vertical rotation, but may limit the global latitudinal range.

Density

Pseudo-Fibonacci sphere only. Controls the density of the sphere. The higher the better for smooth rotations in any direction, at a cost of higher POV number thus lower resolution/POV in the atlas.

Lighting Method

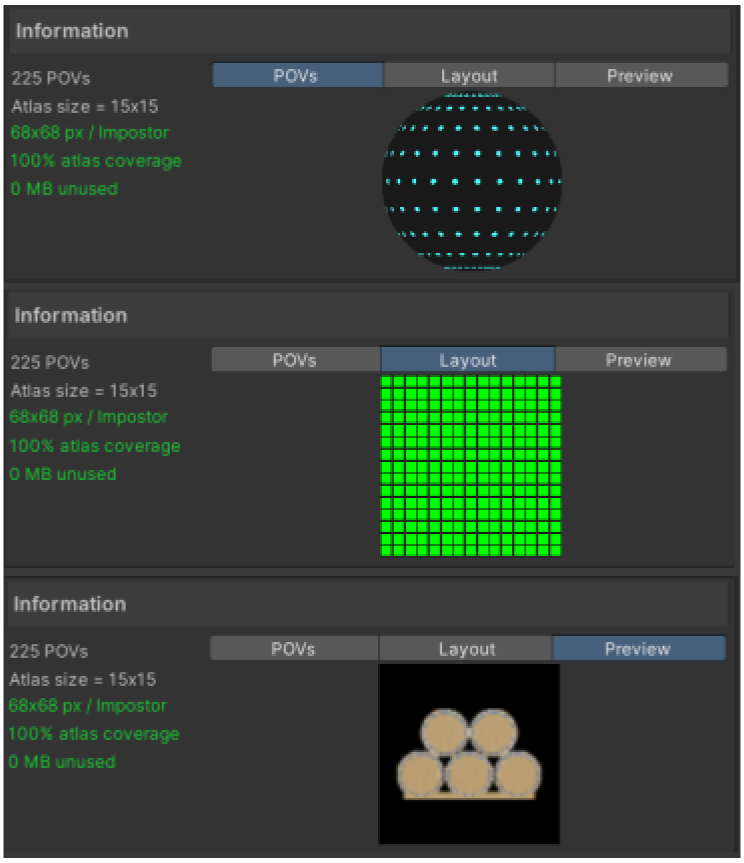
1. **Surface Estimation:** Computes dedicated normal and metallic map atlases to get physically correct light reflections on the impostor. The recommended method. The baked impostor will use a lit surface shader.
2. **Use Scene Sun Source:** Adds the sun source of the opened scene in the baking context. Can provide better results on single lit scenes, especially with intricate smoothness/occlusion materials. Uses significantly less memory per impostor. The baked impostor will use an unlit shader.

Presets

You can add/load presets easily with the dedicated buttons. Gives an additional degree of freedom to match your workflow. A few relevant presets are already included such as 1-POV-billboard or different hemispheres. Importing a preset and modifying values won't modify the preset unless you explicitly export and replace it.

Information

The whole information foldout is readonly. It provides useful visual feedback to help tweaking the previous category fields.



225 POVs
Atlas size = 15x15
68x68 px / Impostor
100% atlas coverage
0 MB unused

225 POVs
Atlas size = 15x15
68x68 px / Impostor
100% atlas coverage
0 MB unused

225 POVs
Atlas size = 15x15
68x68 px / Impostor
100% atlas coverage
0 MB unused

On the left column, some metrics are displayed. The values will appear in green is considered valid, orange when considered too low/high. Please note that these colors are purely informative and won't inhibit the impostor generation.

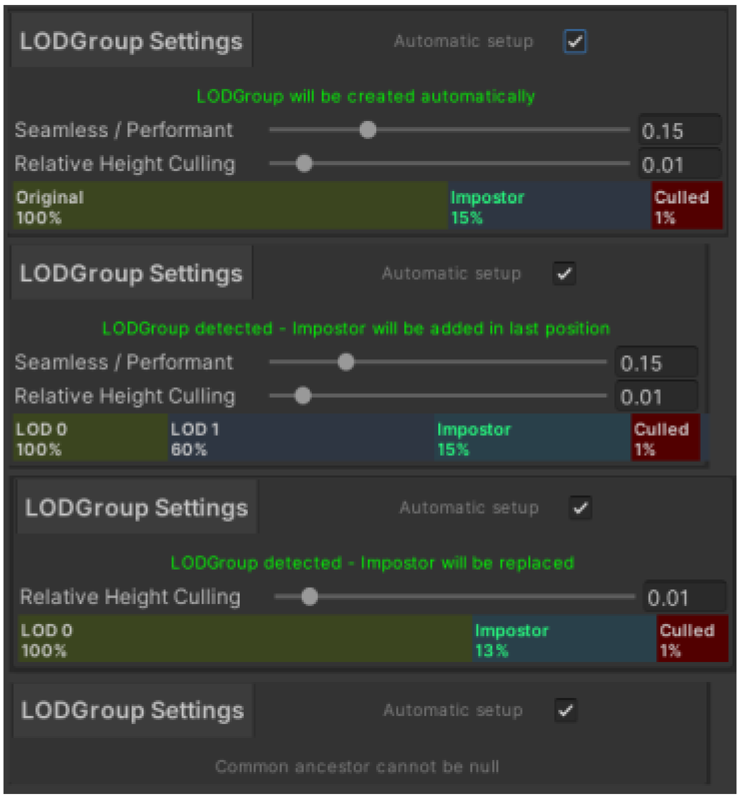
- The total number of POV.
  - For UV Sphere Baking : $N_{POV} = S_{Long} * S_{Lat}$
  - For Pseudo-Fibonacci Sphere baking : $N_{POV} = \sum_{l=-S_{Lat}/2}^{S_{Lat}/2} cos(\frac{2\pi.l}{S_{Lat}}). S_{Long}$
- The Atlas POV size. 15x15 means there are 15 POVs per line and per column.
- Single POV resolution
- The atlas coverage : if $\sqrt{N_{POV}}$ is an integer, the coverage will be 100% otherwise there will be blank slots in the atlas. 100% is better but not necessary.
- Unused memory : The estimated memory held by the blank slots in the grid. Will be 0 if the coverage is 100%

On the right area stand 3 tabs for more visual information.

- POVs : Visualize the Longitude / Latitude coverage. A great helper when tweaking the Quality Settings. You can orbit around with the mouse to see clearer.
- Layout : Visualize the Atlas layout. Filled atlas slots will be green, blank will be red.
- Preview : Visualize how pixelated your each single POV will be. Displays the actual objects entered in the Target foldout. You can orbit around with the mouse as well.

LODGroup Settings

This category is not crucial since you can always modify it afterwards, but it can make you save a lot a time. If you just want to bake the impostor to a prefab without adding it to the scene, you can untick the field Automatic Setup.


LODGroup Settings — Automatic setup ✓
LODGroup will be created automatically
Seamless / Performant 0.15
Relative Height Culling 0.01
Original 100% | Impostor 15% | Culled 1%

LODGroup Settings — Automatic setup ✓
LODGroup detected - Impostor will be added in last position
Seamless / Performant 0.15
Relative Height Culling 0.01
LOD 0 100% | LOD 1 60% | Impostor 15% | Culled 1%

LODGroup Settings — Automatic setup ✓
LODGroup detected - Impostor will be replaced
Relative Height Culling 0.01
LOD 0 100% | Impostor 13% | Culled 1%

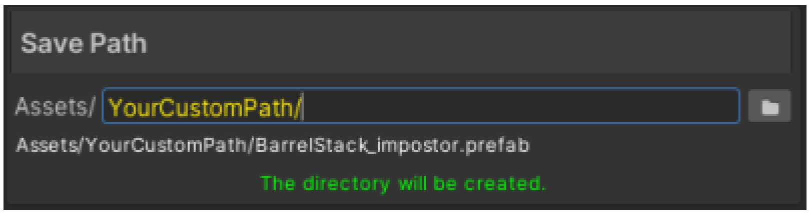
LODGroup Settings — Automatic setup ✓
Common ancestor cannot be null

There are 4 possible scenarios for this area.

- None of the input meshes are part of a LODGroup : A LODGroup will be added on the common ancestor. all the source renderers will be added in LOD0 and the impostor will be added in LOD1 following the given settings. A cross fading is added for smooth transitions.
- There is already a LODGroup but no impostors : The impostor will be added in an additional LOD level following the given parameters. Note that all the source renderers MUST belong to either no LOD or the same LOD level from the same LODGroup to avoid nested LODGroup errors.
- There is already a LODGroup with an impostor : The impostor will be replaced using the same relative height transition. Note that impostors are detected thanks to the ImpostorReference script attached to the LODGroup. If you want to have several impostors in the same LODGroup, just delete safely the ImpostorReference component before baking. Also, the previous impostor's prefab will not be deleted, only the instance in the scene will be replaced.
- There isn't any common ancestor for the input objects. Baking won't be possible.

Save Path

Impostors are saved as packed prefabs, embedding the quad mesh, the atlas(es) and the material. You can customize this path in this section.


Save Path

Assets/ YourCustomPath/

Assets/YourCustomPath/BarrelStack_impostor.prefab

The directory will be created.

You can either type the path relative to the Assets folder or use the file browser with the dedicated button.

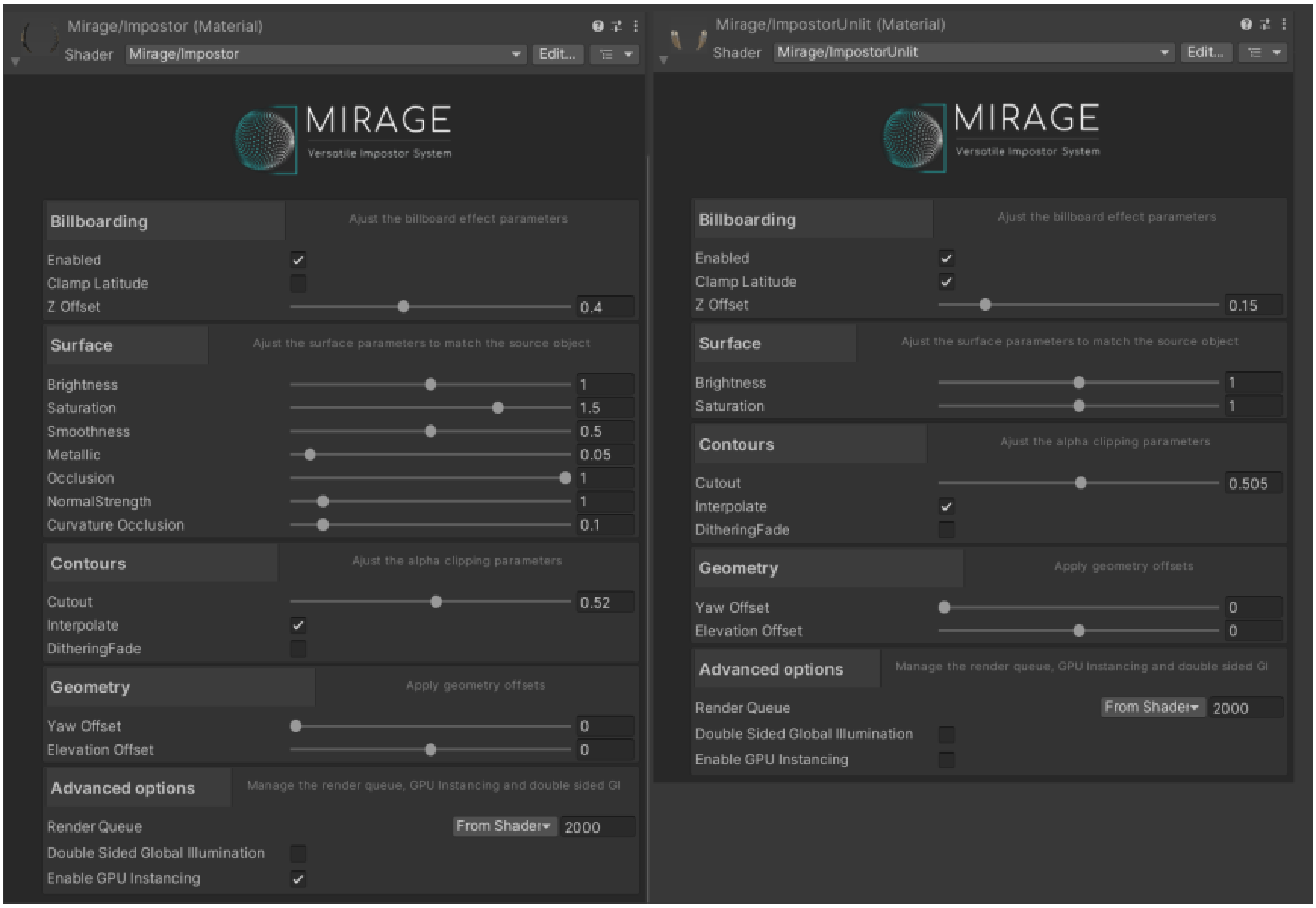
The label underneath the text field hold the actual, unique path that will be used for the impostor.

If the path contains one or more non-existing directories, they will be created automatically at baking time.

Note that no file nor directory is created before baking.

Post-baking adjustments

Both the Lit and Unlit impostor materials have a custom Shader GUI where you can perform advanced post-baking adjustments.



Mirage/Impostor (Material)

Shader: Mirage/Impostor

Billboarding — Ajust the billboard effect parameters

Enabled	✓
Clamp Latitude	
Z Offset	0.4

Surface — Ajust the surface parameters to match the source object

Brightness	1
Saturation	1.5
Smoothness	0.5
Metallic	0.05
Occlusion	1
NormalStrength	1
Curvature Occlusion	0.1

Contours — Ajust the alpha clipping parameters

Cutout	0.52
Interpolate	✓
DitheringFade	

Geometry — Apply geometry offsets

Yaw Offset	0
Elevation Offset	0

Advanced options — Manage the render queue, GPU Instancing and double sided GI

Render Queue	From Shader 2000
Double Sided Global Illumination	
Enable GPU Instancing	✓

Mirage/ImpostorUnlit (Material)

Shader: Mirage/ImpostorUnlit

Billboarding — Ajust the billboard effect parameters

Enabled	✓
Clamp Latitude	✓
Z Offset	0.15

Surface — Ajust the surface parameters to match the source object

Brightness	1
Saturation	1

Contours — Ajust the alpha clipping parameters

Cutout	0.505
Interpolate	✓
DitheringFade	

Geometry — Apply geometry offsets

Yaw Offset	0
Elevation Offset	0

Advanced options — Manage the render queue, GPU Instancing and double sided GI

Render Queue	From Shader 2000
Double Sided Global Illumination	
Enable GPU Instancing	

Billboarding

The billboarding effect consists in making the quad face the camera at any time. This is performed in the vertex shader stage.

- You can disable it with the dedicated toggle.
- Clamp Latitude will stop the vertical billboard effect if the view angle gets out of the latitudinal range. The horizontal billboarding will remain.
- Z Offset can get the a impostor closer from the camera. Really useful to get the most out of the LODGroup's crossfading effect, or to adapt the size at closer range without altering longer ranges.

Surface

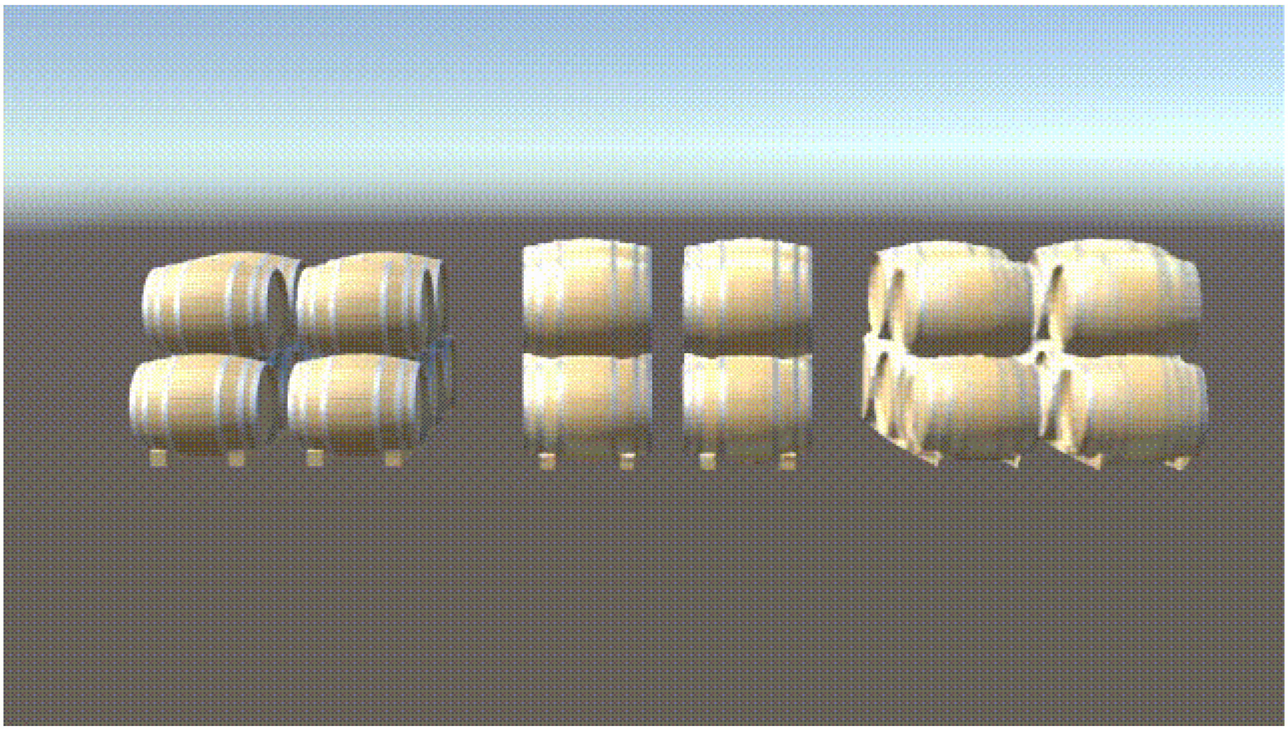
Due to the lack of smoothness, metallic and occlusion maps, it is sometimes required to tweak some surface parameters to match the source objects better.

- The Brightness slider can increase/decrease the albedo power.
- The Saturation slider can increase/decrease the color saturation.
- The Smoothness and Occlusion sliders work the same way than the default diffuse material.
- The Metallic slider is multiplied with the computed metallic map.
- The Curvature Occlusion slider applies an additionnal occlusion in rifts, and sharp angles, to estimate self-shades. Can create a gray outline on some objects.

Contours

Impostors use opaque materials. To get the contours right we use alpha clipping to discard pixels that are not representing the object on each POV. The contours are usually fine out-of-the-box, but sometimes it is required to adjust a few settings.

- The Cutout slider can increase/decrease the sharpness of the alpha clipping geometry. You should not go higher than 0.5 to avoid cropping some important features.
- The Interpolate toggle turns On/Off impostor bilinear interpolation smoothing. Interpolation removes the snapping effect, but adds blur. Note that interpolation adds shader complexity.
- The Dithering Fade toggle turns On/Off Dithering fading on the edges of the impostor. Independant from the LODGroup crossfade effect.



From left to right: The original model, interpolation disabled, interpolation enabled.

Geometry

These settings are theoritically not needed but give additionnal freedom to advanced users.

- The Yaw/Elevation offset sliders make the object virtually rotate on the Horizontal/Vertical axis respectively.

Advanced Options

You can control the Render Queue, Double sided Global Illumation and GPU instancing, like the default surface shader.

Compatible objects

Mirage Impostors are compatible with any object with a MeshFilter and a MeshRenderer attached.

Note that translations (on all axes), scaling (on all axes) and rotation (around the vertical axis only) are supported.

Limitations

Due to their nature, Mirage's impostors have their few limitations.

- Full rotation compatibility : as said just above, the impostors won't be rotated around X and Z axes even if their parent does.
- Lossy scale : Mirage uses the transforms lossyScale (https://docs.unity3d.com/ScriptReference/Transform-lossyScale.html) of the source objects. If the source objects are skewed by nested non uniform scaling a rotation, the impostor may be a appear close from but source but unskewed.

- Shadows : Mirage impostors are compatible with shadows but they are not physically correct since it only projects the alpha clipped quad instead of having a dedicated shadow pass. It means shadow will be correct if the light comes from the front or the back but not the sides of the impostor. This is a simplicity choice, since correct shadow would add a GPU-intensive shader pass and shadows are usually not even visible from the impostor nominal distance. Moreover, it keeps Mirage natively pipeline independant since shader graph is not yet designed to add custom passes.

Contact

Didn't find what you were searching for?

Feel free to contact me by email or on my discord channel.

support@leochaumartin.com (mailto:support@leochaumartin.com)

https://discord.gg/kYwzdvAt8q

References

https://www.gamedeveloper.com/programming/dynamic-2d-imposters-a-simple-efficient-directx-9-implementation

https://calcifer.org/kenneth-christiansen/ComputerScience/imposters.pdf

https://developer.nvidia.com/gpugems/gpugems3/part-iv-image-effects/chapter-21-true-impostors