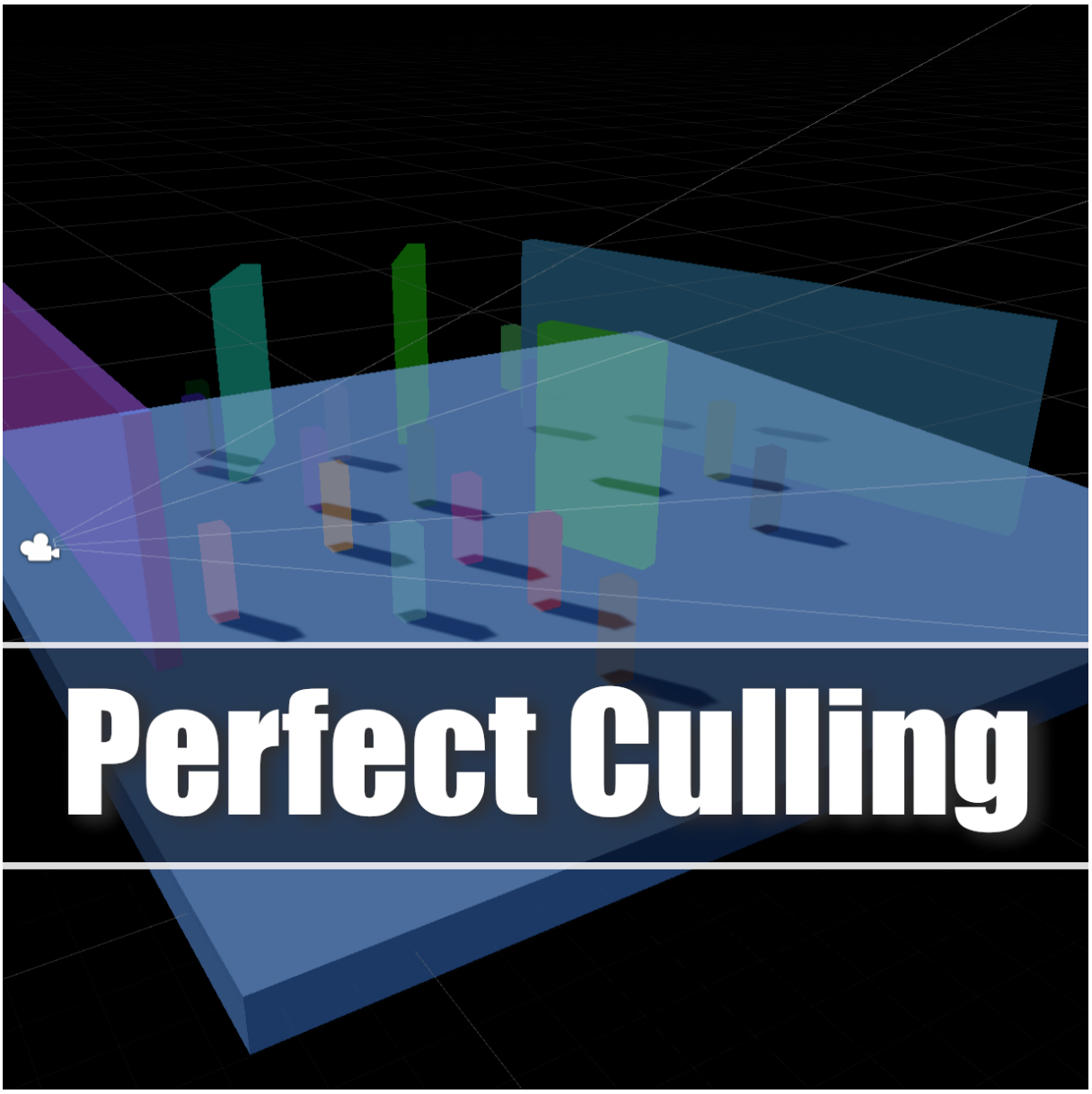


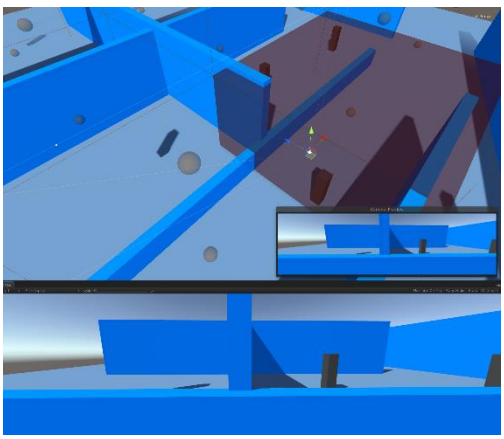
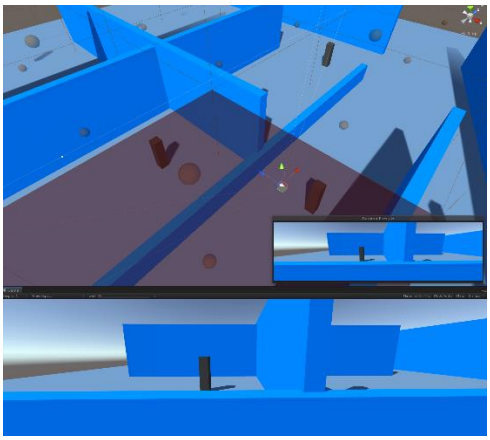
Troubleshooting

The first and very important thing to understand is that Perfect Culling only culls objects that it could not “see” during the baking process. Concluding that the asset is always right and you just need to understand how it came to this conclusion so you can fix it. Of course, this assumes that there is no bug somewhere else but the asset should be pretty stable at this point.

Usually after you figured out what the asset saw you probably would agree that the asset made the right decision and you just need to tweak some settings to give it more information so it can make a better decision that works for your level.

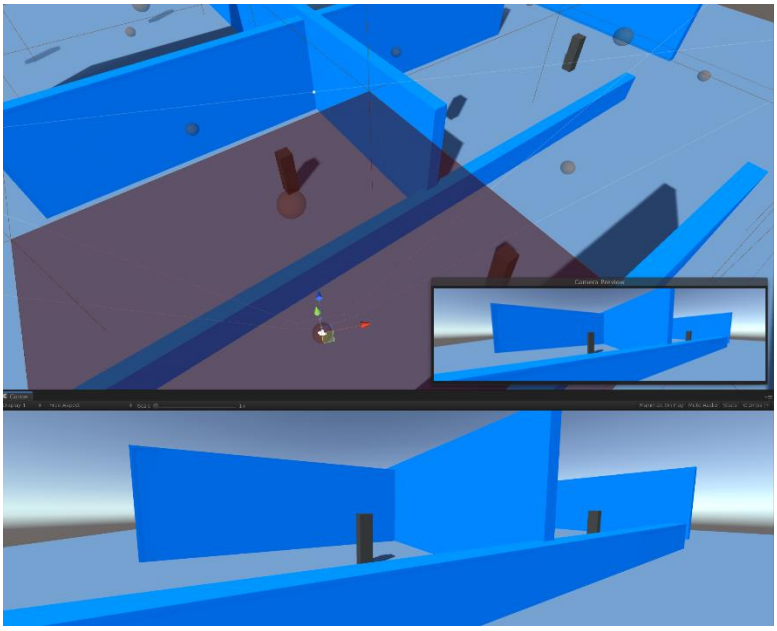
This document will describe some common issues and how to diagnose and fix them.

Asset popping

In this example you'd expect both pillars to render but only one is visible at a time and they pop in and out very visibly.

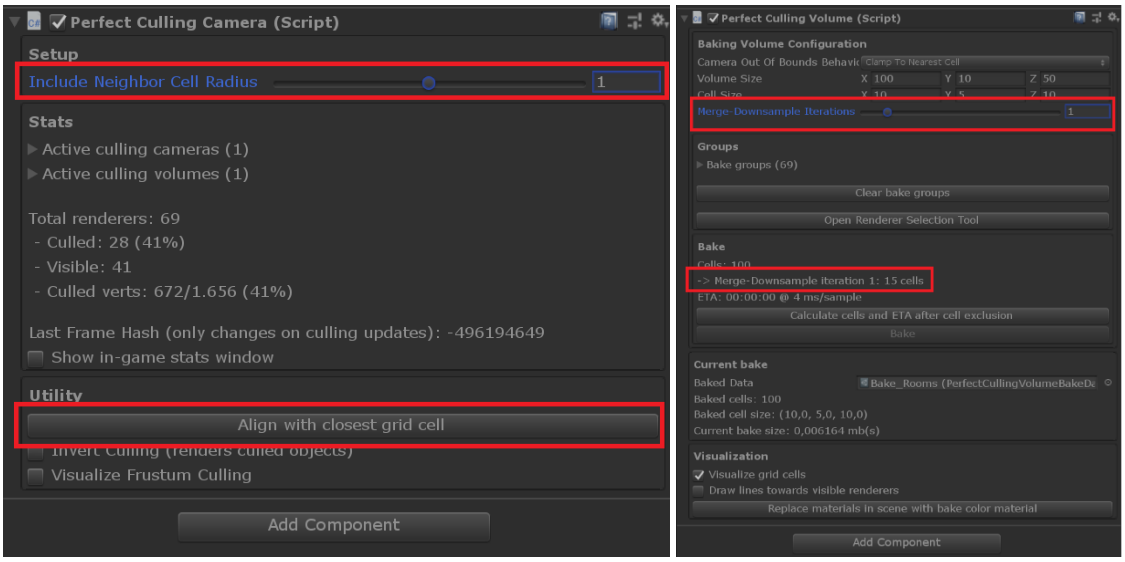
The first step should be to confirm what the asset saw during the baking process. So, select your camera and press **Align Camera with closest Grid Cell.** The camera will snap to the sampling location of the currently active cell.



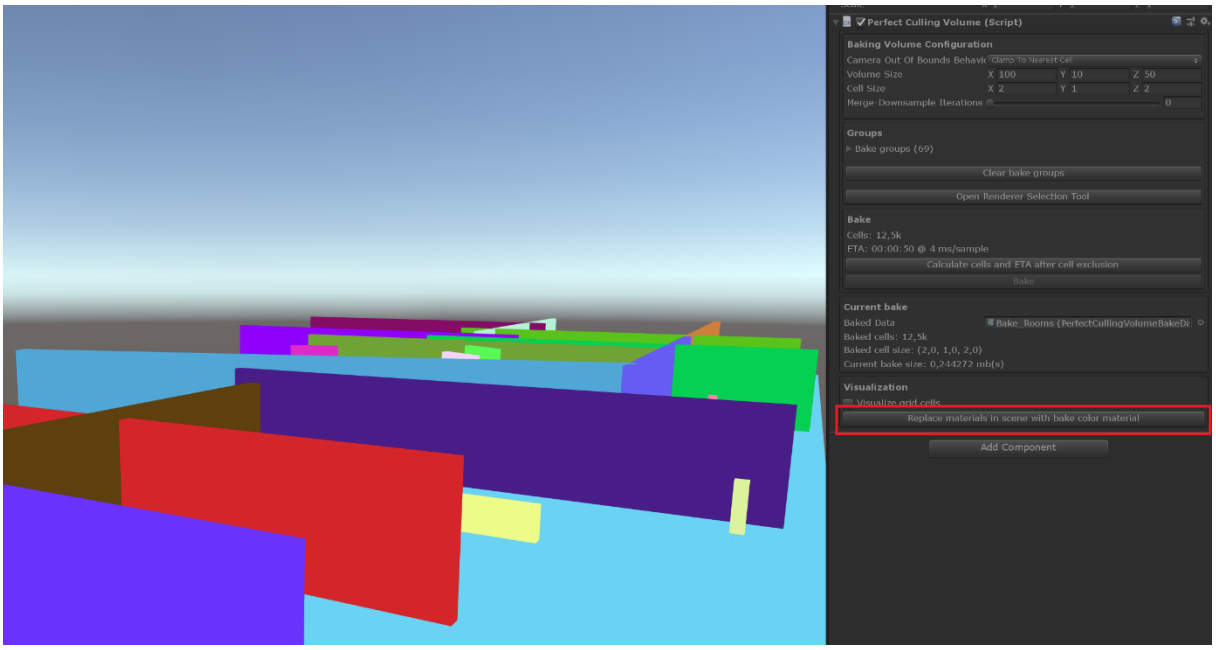
Aligning the camera to the closest grid cell suddenly makes us understand why the asset culled the other renderer: It simply was not visible from the sampling point.

There are multiple ways how you can fix it:

- Reduce the cell size so you end up with more cells and thus more precise occlusion data
- Bake in what neighbor cells see using the **Merge-Downsample** feature on the **PerfectCullingVolume.** This also reduces your memory usage and got no overhead at run-time.
- Make use of the **Include Neighbor Cell Radius** on your camera to take into account what neighbor cells see.



Always prefer to bake it in (image on the right). The feature on the camera (image on the left) is meant as a quick way to confirm that the issue can be fixed by considering neighbor cells.



If you still cannot understand why renderers disappear enter Play Mode and press **Replace materials in scene with bake color material.** This will swap out all materials in your scene to make them look like the asset saw them during the baking process. You might find a transparent object that blocks the view because it was rendered opaque. You can force a material to render transparent by adding **pc_trans** to its name. You can also force a material to render opaque by adding **pc_opaque** to its name.

Some renderers such as foliage might make use of special shaders that render double-sided. You can add the *PerfectCullingRendererTag* component and check the double-sided checkbox to make them double sided in the baking process as well.

Also, if a renderer did not change to the bake material you maybe missed to add it to the Perfect Culling Volume or the renderer type is not supported. For instance, Unity Terrain is not supported (though you could convert it into a mesh to make it work with Perfect Culling).

Most of my level is not culled

A very common reason for this is baking cells that are below the level. You will find that many meshes are hollow from the inside because they are backface culled. That means if you sample from the other side you can see through them. You might want to look into the **Exclude cells - Sampling Providers** documentation to exclude such cells. This saves memory and reduces your bake time and also avoids that your entire level could become visible in a very elegant way.

Another option could also be to add the **PerfectCullingRendererTag** and check the render double-sided checkbox. That might be useful in other places, too. However if you could just exclude the sampling position from the bake always prefer that as it reduces the baking time and memory usage.

Large chunks of my level disappear in some places

This is most likely caused by a sampling a position where the view is completely blocked and thus no other renderers have been visible. You probably just need to include neighbor cell data to fix this.

The bake size is pretty large

The bake size depends on the number of cells and the number of renderers and that's why you need to reduce some of these factors.

A very simple way to half the number of cells is to make use of the **Merge-Downsample** feature on the **PerfectCullingVolume.**

You also should make sure that you only bake cells that really matter. Please see the **Exclude cells - Sampling Providers** documentation.

You could also try:

- Split your level up into multiple smaller volumes
- Increase your cell size to reduce the number of cells
- Reduce the size of your volume (only places the camera can reach should be contained)

Finally, you can also combine meshes that are very close to each other into a custom bake group. That way the asset only needs to reference a single bake group thus reducing memory usage. Please see the **Custom Scene Groups** documentation.

Some assets disappear but I cannot figure out why

Does the asset make use of LODGroup? If you are not using the native renderer (default on Windows) but Unity for baking and a renderer is culled by Unity because of the LODGroup setup it would not be visible to Perfect Culling and thus be culled by Perfect Culling as well. In this case make sure you are using very conservative LOD quality settings during the baking process.

If you are using the native renderer (default on Windows) the asset would always use LOD0 for rendering.

Just some additional information for your investigation.

Support

If you are running into issues, need assistance, got some feedback, please feel free to get in touch.

Mail: info@koenigz.com