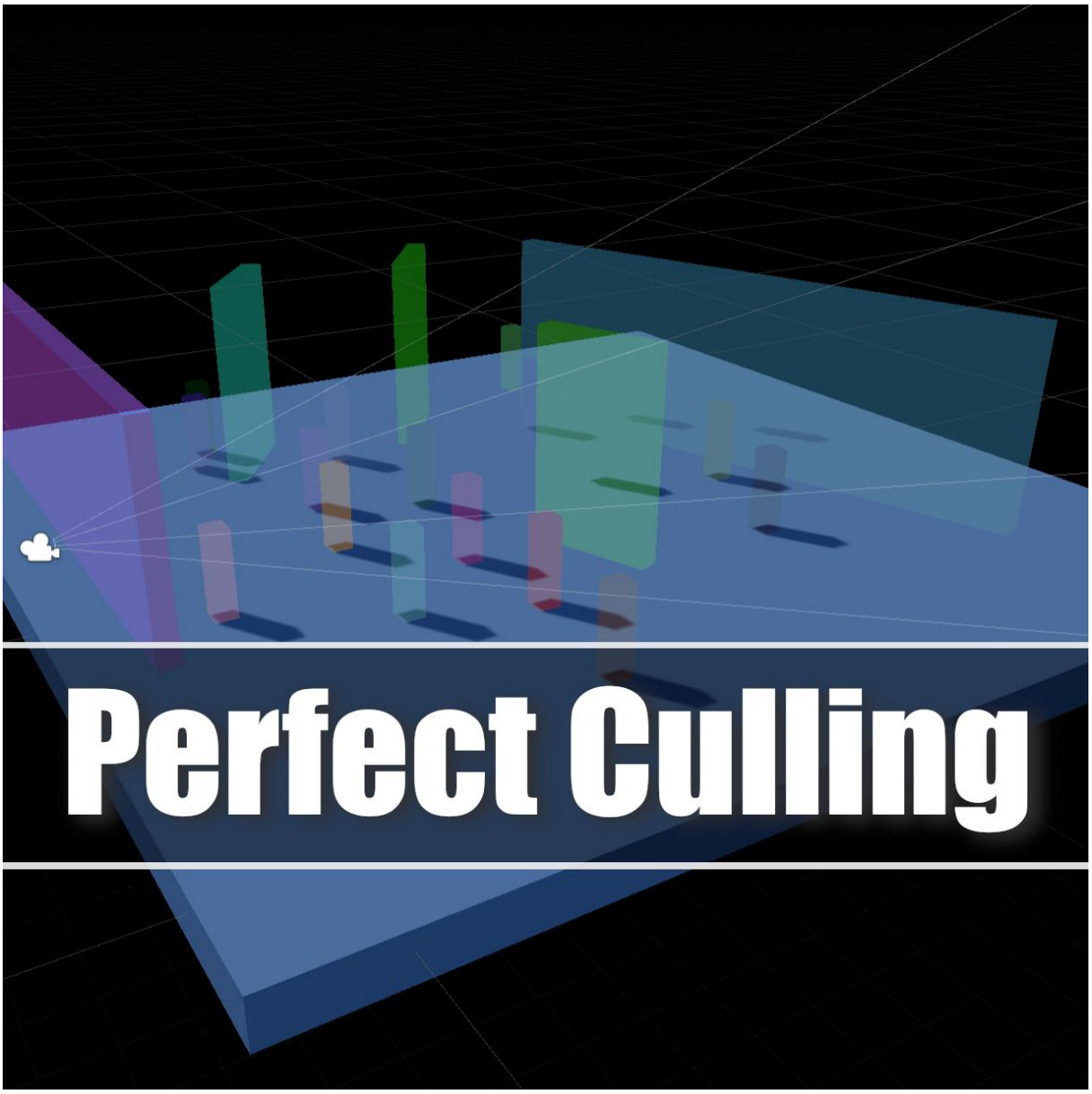
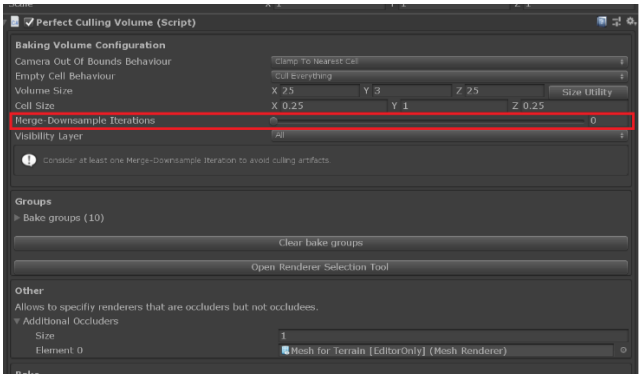
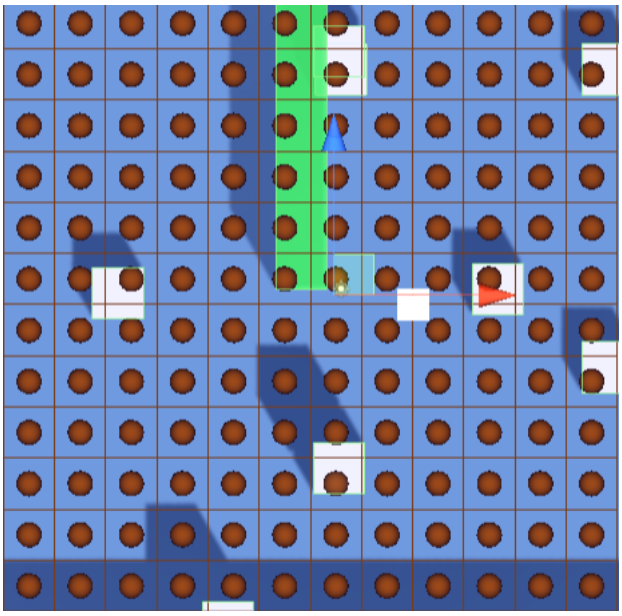


Merge-Downsample

This part of the documentation just further explains what this mysterious feature is really about, why it is great and you should use it but also why it can also cause issues in some situations.



Merge-Downsample Iteration slider on Perfect Culling Volume

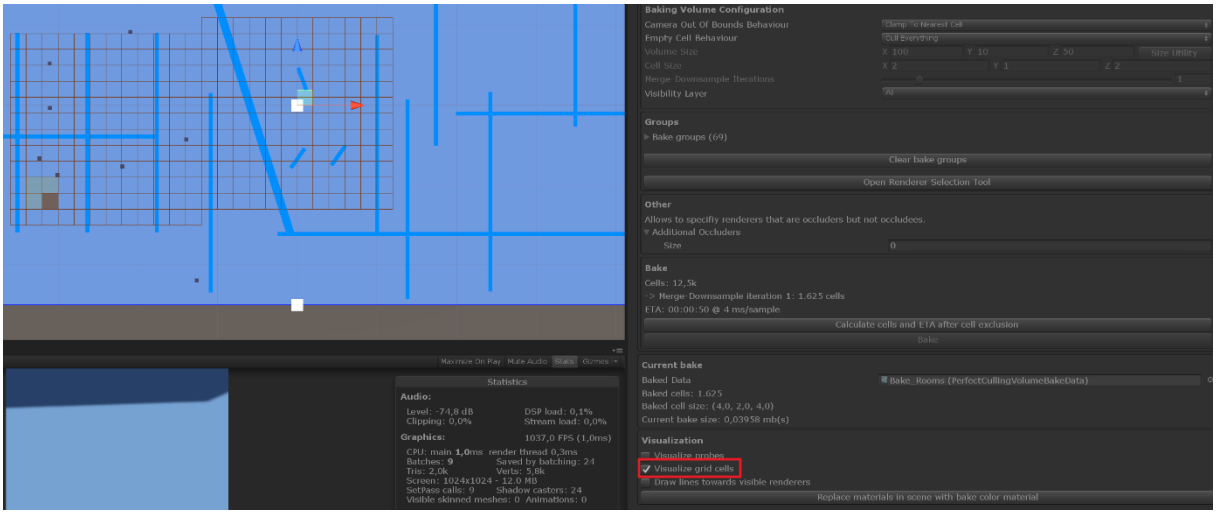


Sampling pattern

Perfect Culling samples and stores visibility in a grid pattern. The visibility at the sample point will be 100% accurate but the further you move away from this point the less reliable is this information. You would notice this as you move the camera around because assets pop in and out.

Because we know that the visibility information is correct at each individual point, we can improve the information for in-between locations by simply combining nearby points.

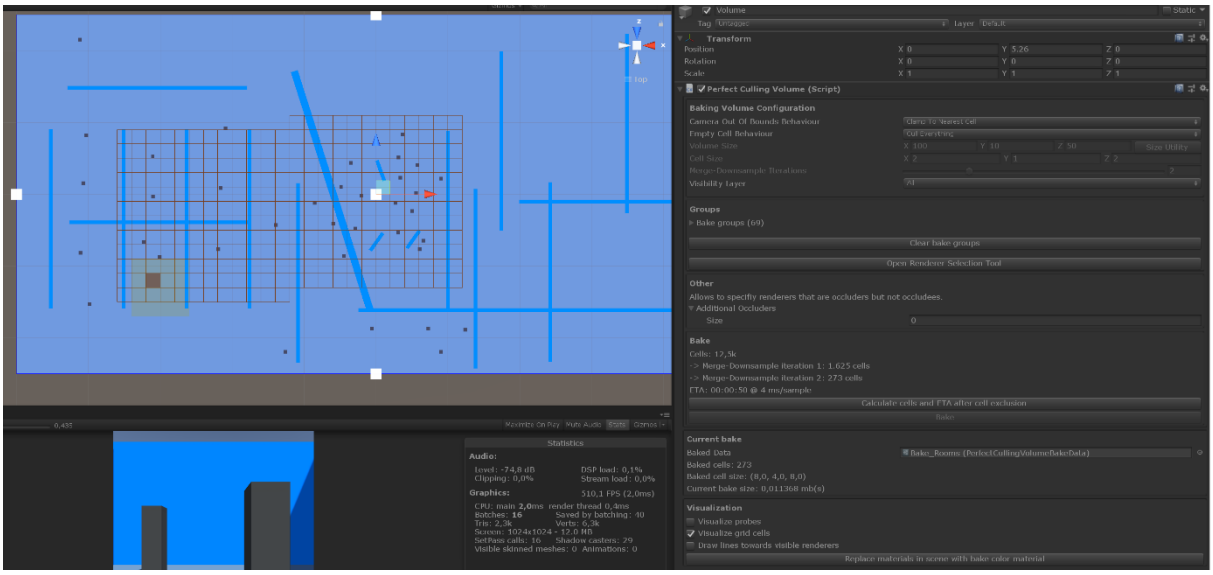
That is pretty much what merge-downsample does. It takes cells and combines them into a larger cell. Each iteration thus doubles the cell size and because there is less cells to store it also reduces the bake size. Pretty much a win-win!



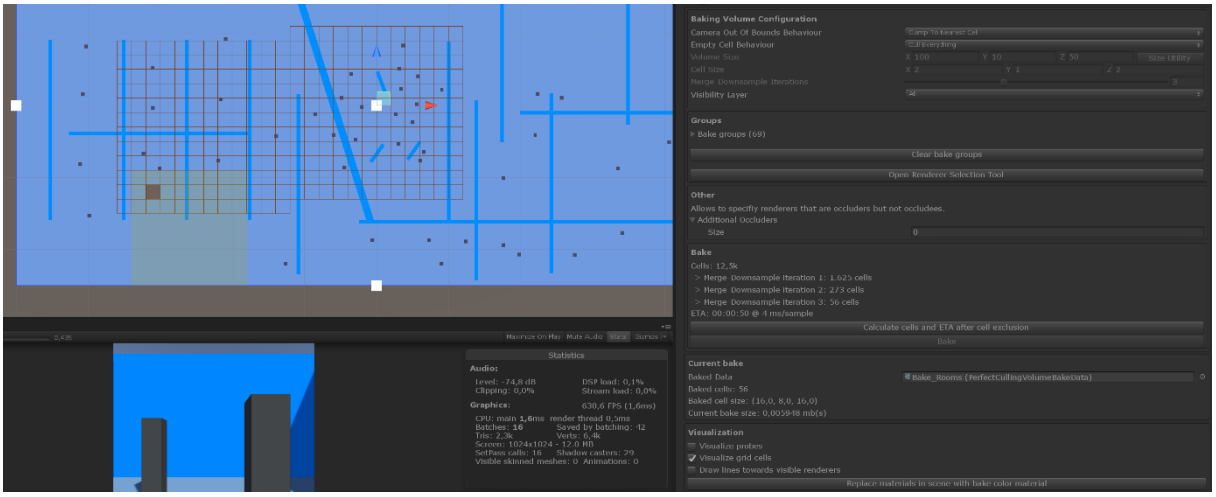
One iteration of Merge-Downsample Iterations

To visualize this enter Play Mode and select your Perfect Culling Volume and enable “Visualize grid cells”. What you see in the screenshot is the active cell for the camera (red cell) and the cell that was created by combining neighbor cells (yellow cell). All yellow cells contain the same information.

You can also see that the combined/merged cells grow larger and larger in every merge-downsample iteration. That is because the previous combine result is used for the next iteration.



Two iterations



Three iterations

Because the merged cells are composed out of many smaller cells the culling is becoming more and more conservative and reduces culling effectiveness. As a matter of fact, you could even end up merge-downsampling so often that only a single cell is left. That is usually not what you want (unless you have a mostly immobile camera; here that might be handy).

You can also see how the size of the bake was reduced in every iteration.

<u>When to use merge-downsample:</u>

- To remove culling artifacts such as renderers popping in and out
- To reduce bake size to save memory

<u>When not to use merge-downsample:</u>

- Still got plenty of memory available
- Culling is already too forgiving
- No culling artifacts are visible

The Perfect Culling Camera also features an option called “Include Neighbor Cells” that performs a similar step at run-time and allows you to test how merge-downsample would change your culling results.

However, this comes at a performance cost and ideally you only use it for testing. The merge-downsample bake step does not cause any performance overhead at all and should always be the preferred option.