I've noticed with the large number of new people coming into vrchat there have also been people wanting to create their own worlds. But are having performance problems. This document is to help do some of the basics of world optimization.

Static objects
Baked lighting
Occlusion culling
Lod
Instancing/Prefab
Texture size/Polycount
Physics
Shaders
Vrchat

Static Objects
source: https://docs.unity3d.com/Manual/StaticObjects.html

one of the basics is to make any objects in your world that don't move to a static object.
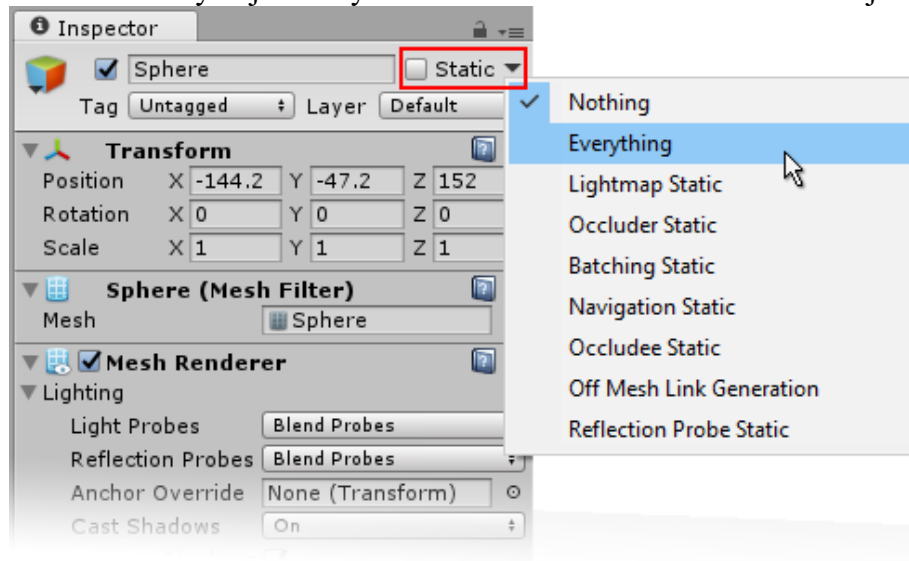
If this is not checked the object is classified as dynamic making the game recalculate it every frame.

Baked Lighting
source: https://docs.unity3d.com/Manual/LightMode-Baked.html
This can only be taken advantage of if the object is static. If the object is dynamic the object lighting will not be baked. The lights in unity usually have 3 modes real-time, mixed, and baked. The real-time lighting should only be used if the light source itself is moving around. Other than that it should be mixed or baked. Mixed lighting is what most people will use it bakes static objects but also lights dynamic objects. Where baked lights will only light static objects not lighting dynamic objects.
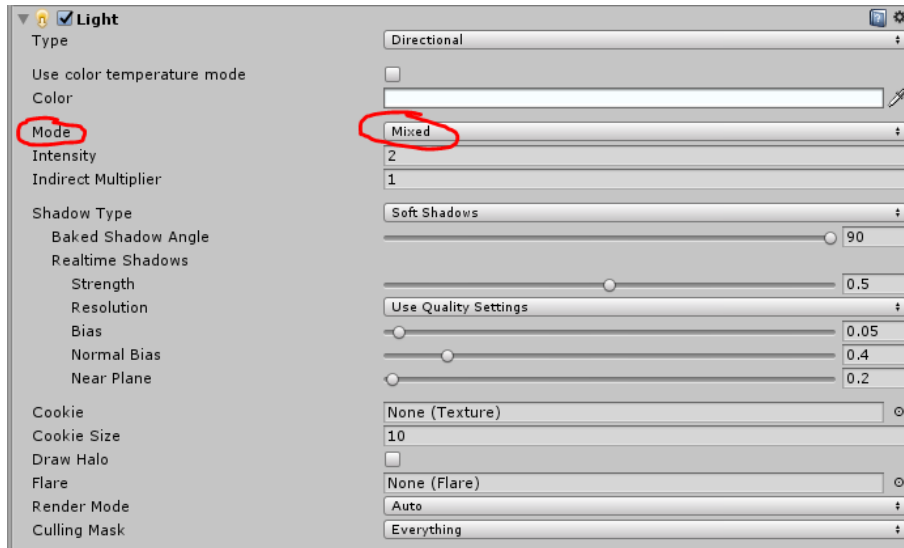
to have more control over the baked lighting there is a lighting window which you can open by going to Windows>lighting>settings. In this new windows, you can change the resolution of your baked light mapping in the lightmapping settings area. The default is 40 texels but 20 or 10 is ok. Also making the lightmap size to 2048 will help with draw calls.
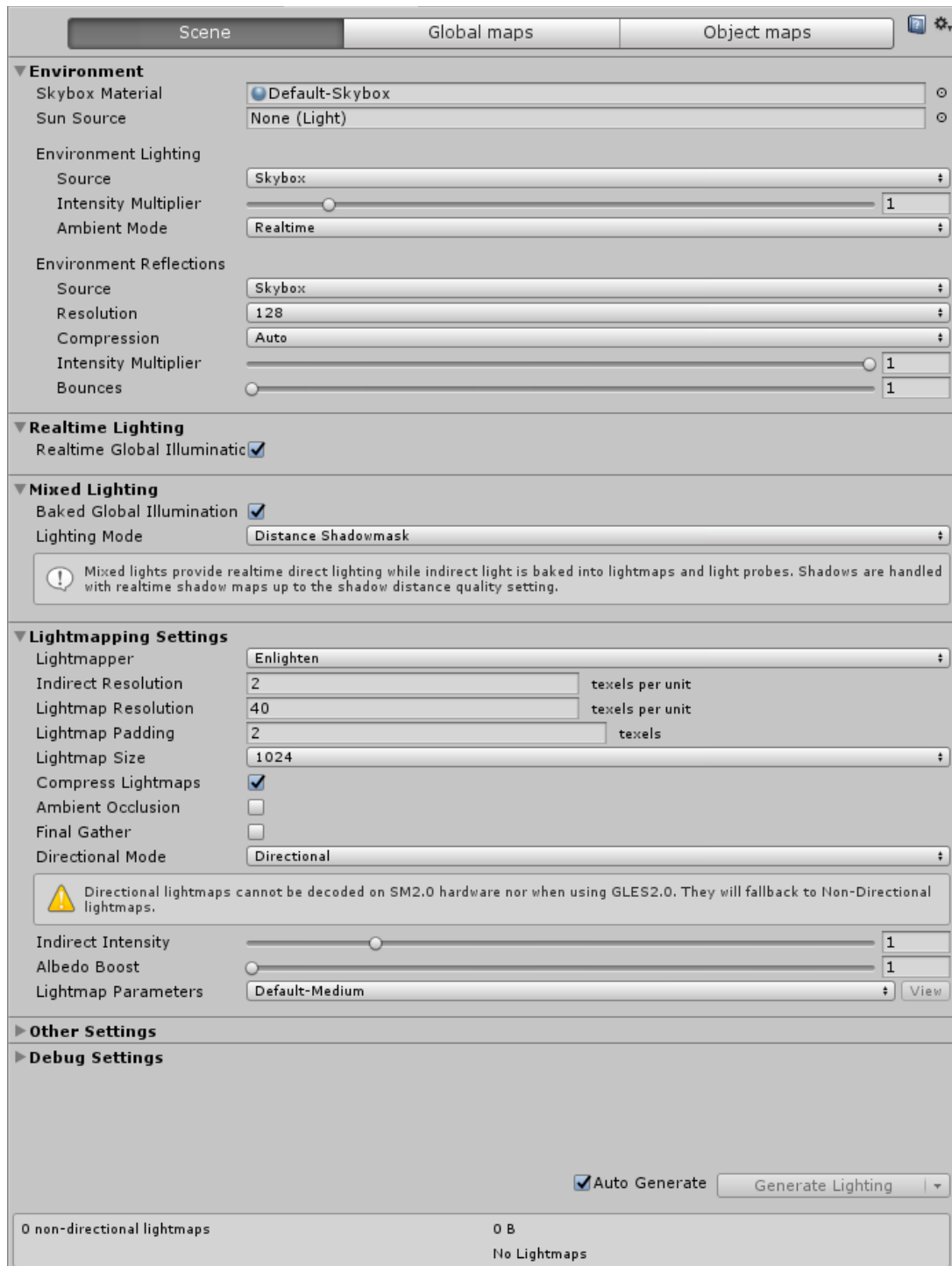
Occlusion culling
Source:
this can help in a big map or a map with a lot of objects. Occlusion culling makes it so objects not seen by the camera will not be rendered. So all your GPU power is being used on what actually being seen. This can be accessed by going to Windows>Occlusion culling. Usually, the default settings will be fine so just click bake. But it is suggested to look up more information about this since it is a powerful tool used properly.
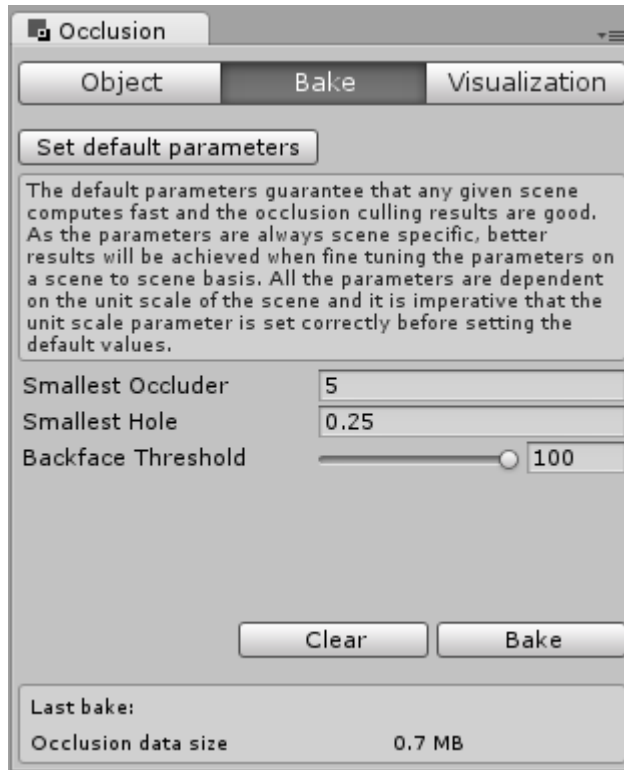
Lod (level of detail)

source:https://docs.unity3d.com/Manual/LevelOfDetail.html

this is used in almost every modern game. This is a component in unity letting different objects to be loaded in depending on your distance to the object. Cause if you're viewing a high-quality model from far away you can barely see the details but the computer still has to render all the details. So by replacing that high-quality model with a low quality one far away, you wouldn't be able to tell the difference.

This can also be used with small objects in your world so you don't render them until you get close to them.

This talks about lod a bit more:http://www.indiegamedeveloper.tv/about-level-of-detail-lod-in-game
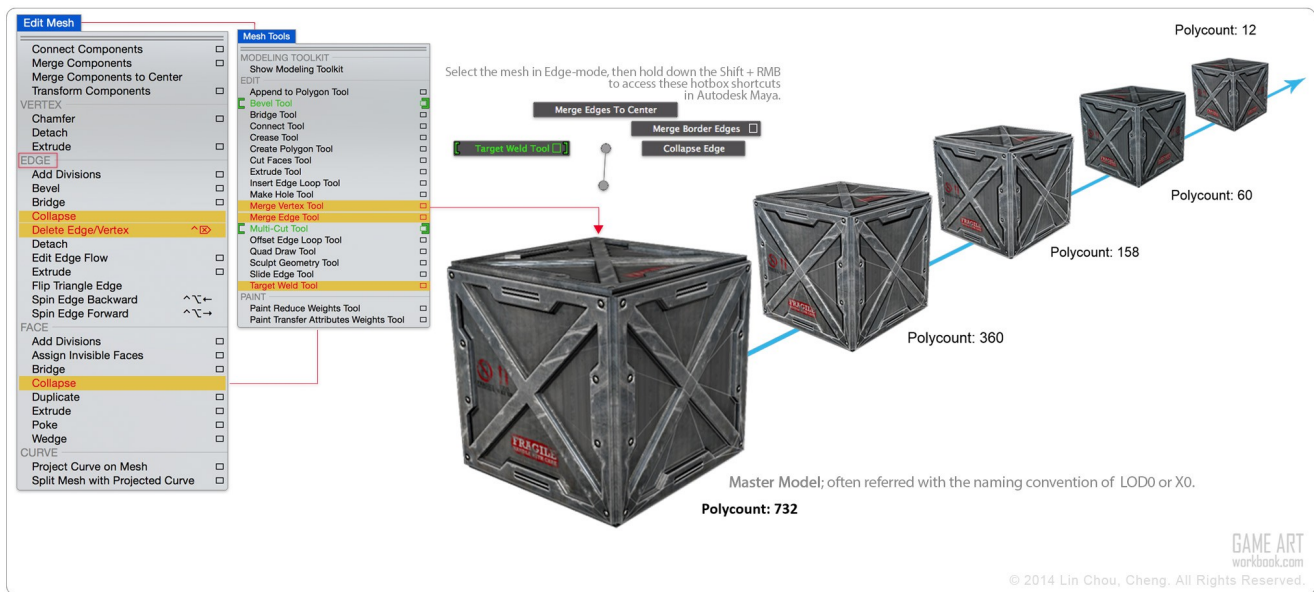
Instancing/Prefabs

most people do this without realizing it. Instancing lets the computer know it just has to copy and past a calculation. In the photo, you see a lot of cubes if instancing didn't exist the computer would have to calculate every object as different pieces losing performance. But with instancing it knows that each cube is the same so it just has to calculate the one cube then past it multiple times. You can create your own instances with prefabs so if you have an object or objects that are repeated save it as a prefab.
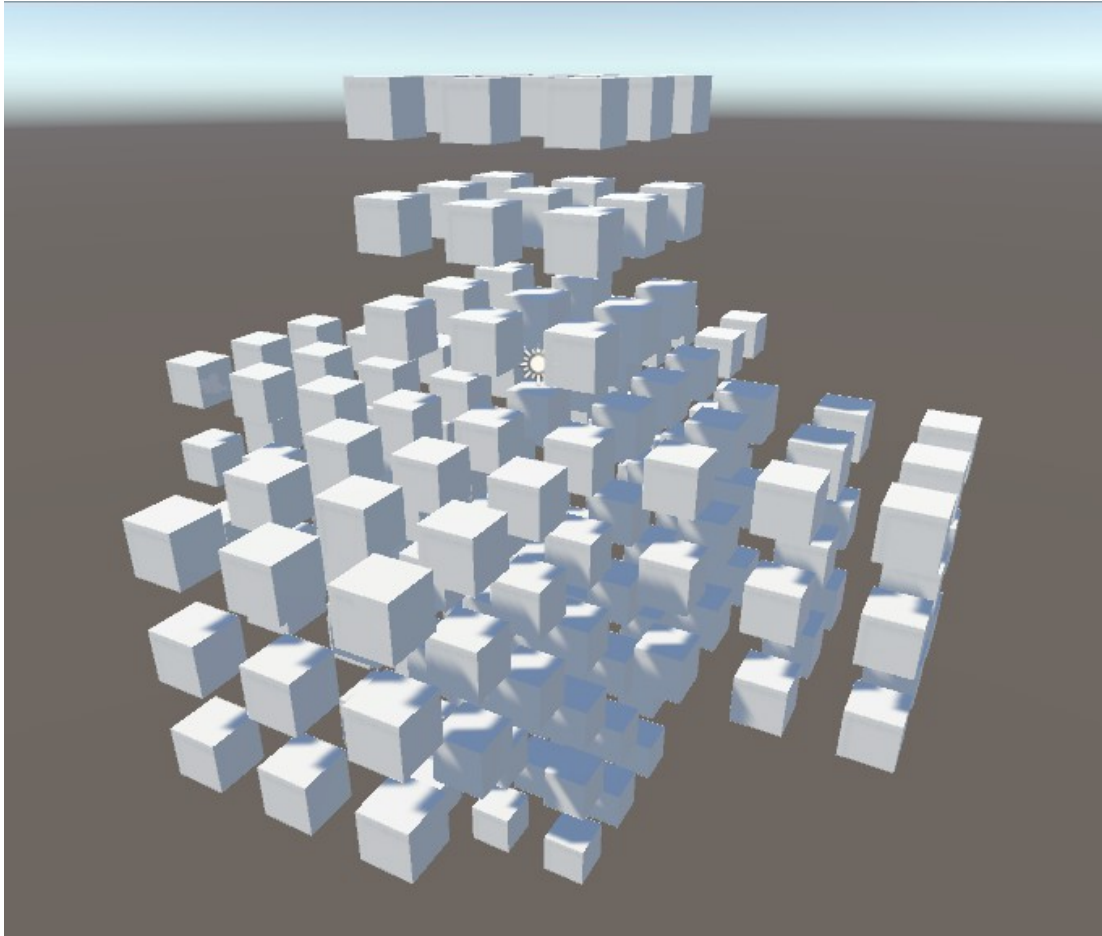
<div align="center">Texture size/polycount</div>

this is a basic one but the main point doesn't have any unnecessary detail. If the detail can be made on a texture don't use the extra polygons for that detail. If you only see that texture from far away or not up close don't use a 4k texture. Keep polycount low especially if the object doesn't need to be very detailed.

<div align="center">Physics</div>

if graphics is not a problem physics could be one. Make sure the number of objects with physics are limited.

<div align="center">Shaders</div>

this is one I have a lack of knowledge about but I do know this can be a source of problems. Were some shaders can be simple and don't take up processing while others can.

<div align="center">Vrchat</div>

there are a few tools in the vrchat that can cause bad performance the big one are mirrors. Make a button that enables and disables the mirror locally so unless someone wants to look at it it's disabled.
While not as bad video panels have also been known to decrease performance. Also having a lot of avatar pedestals can make your world load in slow.

Helpful links:
https://wiki.unity3d.com/index.php?title=General_Performance_Tips

https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html