# COMP2007 - Game Development

## Scene Optimisation & Debugging

When our scenes have lots of 3D meshes, performance can be a problem on lower powered machines.
Large worlds to explore also need management of the scene assets so we only do processing work we need to.
The **Occlusion Culling** system allows us to selectively render only meshes in front of the camera and within set boundaries.
**Occlusion culling** allows unity to stop performing operations it does not need to for rendering.

Objects far away in our scene can also be optimized.
**LOD** or **Level Of Detail** for meshes will swap a mesh at runtime for a lower polygon mesh over certain distances away from the camera.
We can define several versions of a complex mesh with gradually less detail for viewing nearby and far away distances.

GameObjects with meshes can be optimized for performance using the **static** tickbox.
**Static** meshes cannot be moved at runtime, but provide the best performance.
The static setting on a GameObject can optimize its mesh in several systems including lighting, rendering, navigation and batching.

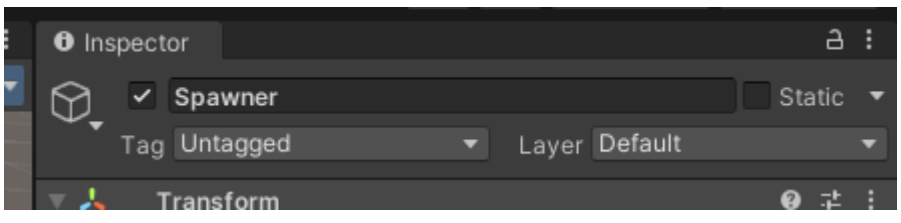
**Overlays** are part of the **Scene view top menu**, they allow us to check our meshes in our scenes for density, UV usage, alpha transparency usage and much more.

The **Game view**'s **Stats** dropdown can give us a brief overview of our games performance while it is running. This includes FPS (Frames Per Second), how many meshes we are rendering, how much memory our graphics are using and more.

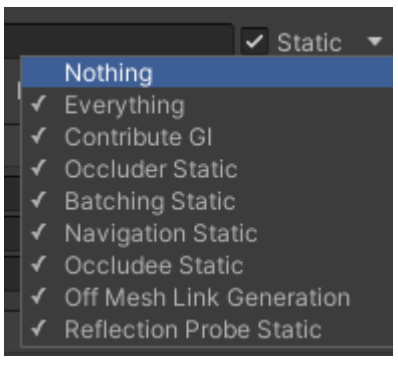
# Occlusion Culling

Occlusion Culling works with Static Meshes and Terrain.
To use Occlusion Culling, look at your scene and determine which GameObjects you wish to become static.

For all the selected GameObjects, tick the **Static** box in the inspector




The dropdown arrow besides the Static setting shows us the many systems we can affect with the setting.
We can set these individually if need be.
The setting **Occludee static** and **Occluder Static** are used with the Occlusion system.

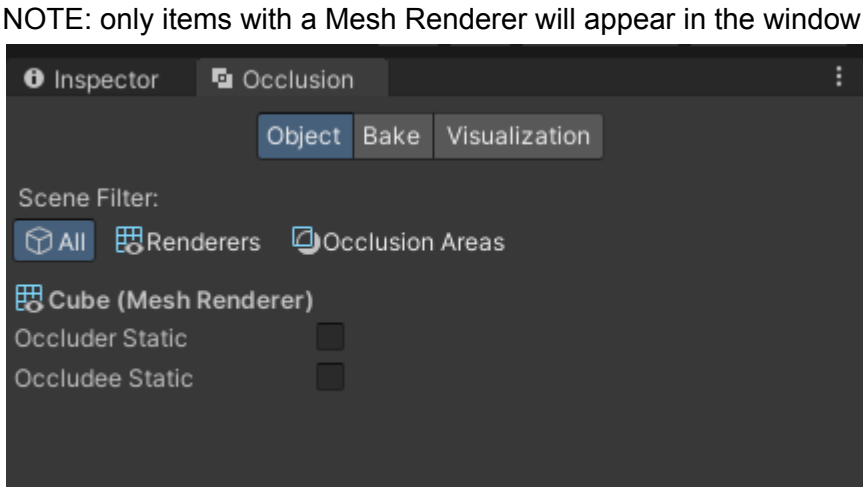



[A guide to Static GameObjects]

An **Occludee** can be occluded by other GameObjects, this is used for smaller meshes.
An **Occluder** will occluse other GameObjects, this is used for larger meshes.
You can tick both by default and not worry for most circumstances!

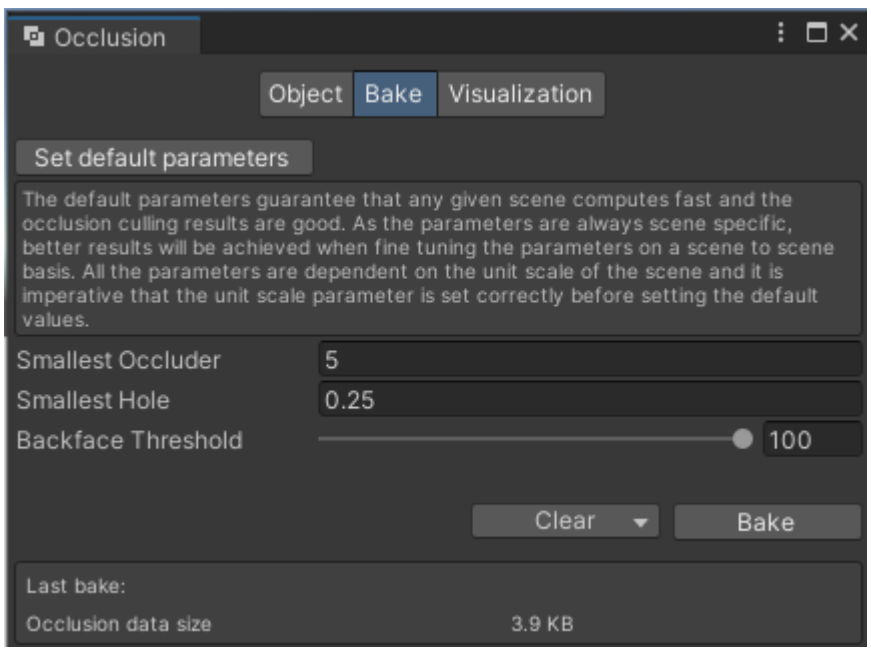
The **Occlusion Window** allows us to **add objects** to the Occlusion list, **visualize** the current Occlusion in the scene and **bake** our current settings.
Open the Occlusion window from the top menu: Window -> Rendering -> Occlusion Culling.
[Occlusion Culling]

We can add a new object and set its Occlusion settings in the **Object tab**
NOTE: only items with a Mesh Renderer will appear in the window when selected

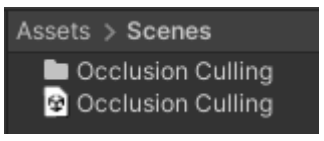



[Occlusion Culling settings window]

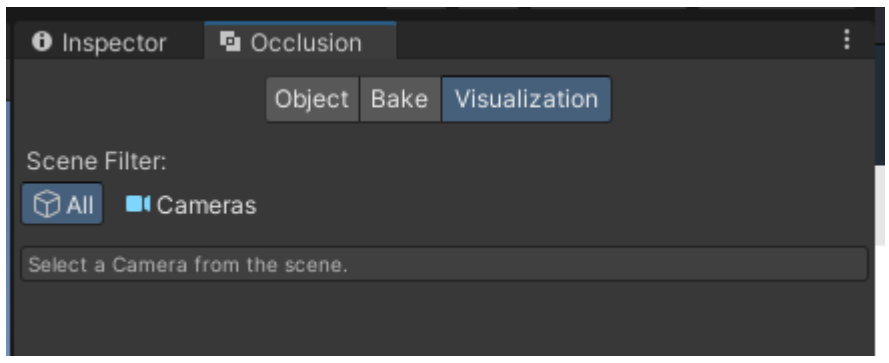
The **Bake tab** allows us to bake **Occlusion data** into our scene.
We can set the size of items we wish to occlude and any holes or cover back faces that may be present in our geometry.
Click the **Bake button** to bake the occlusion data.
Click the **Clear button** if you wish to delete the Occlusion data.



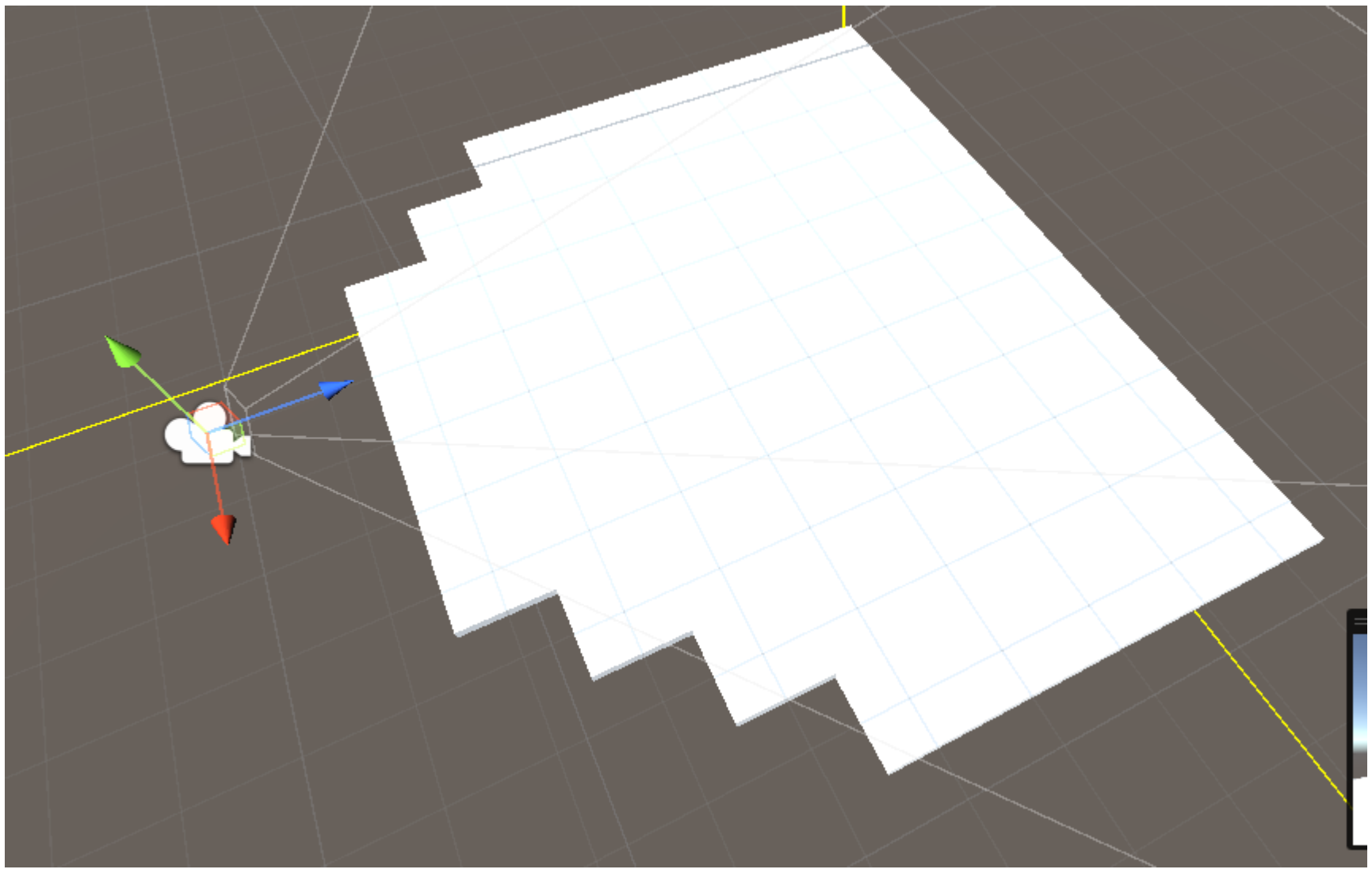

**Occlusion data** is stored in folder next to our Scene file in the project view



The visualization tab allows us to see the current Occlusion in our scene view




Objects not appearing in the camera frustum (view angle) will be invisible in the scene view

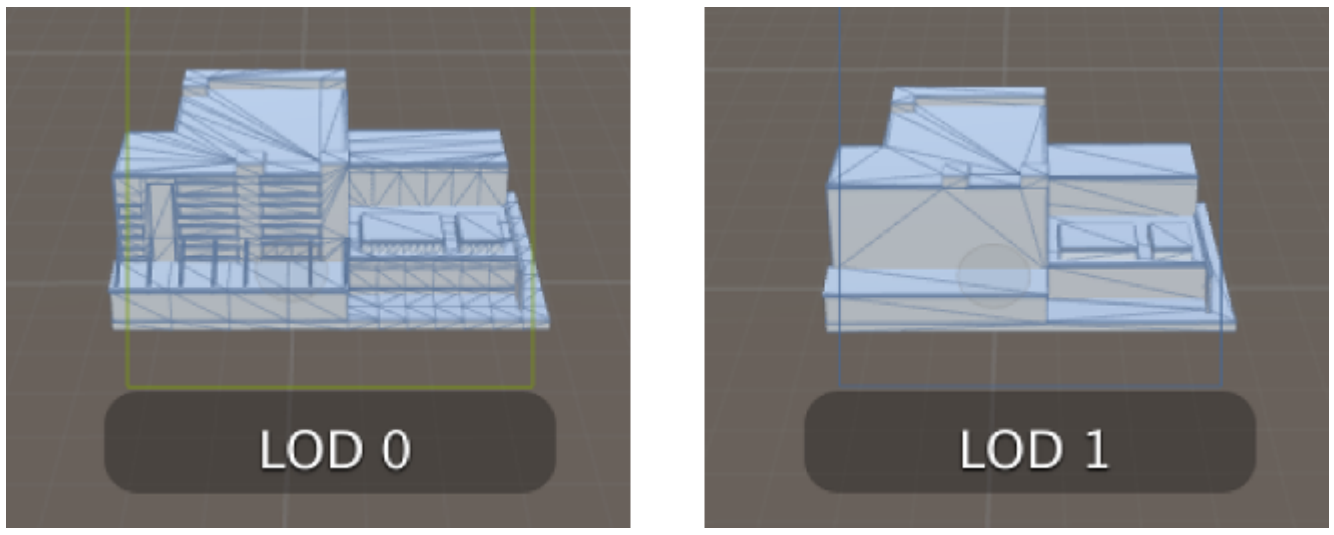


# Level Of Detail (LOD)

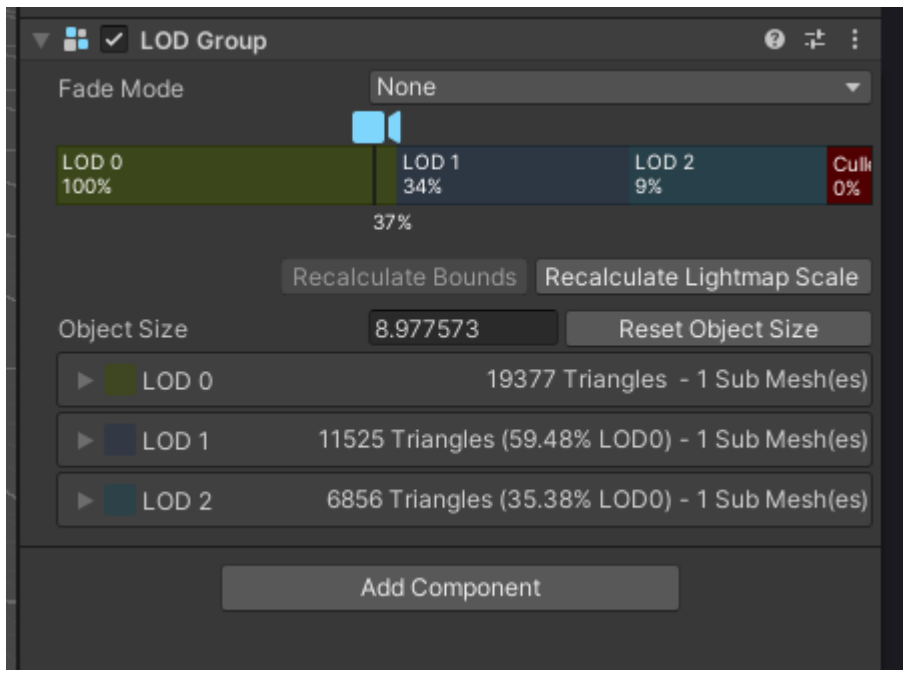
We can change the mesh a user sees in the **camera** depending on the **distance to the mesh**.
Meshes with less detail and smaller textures take up less processing; we can optimize our game's performance by swapping out meshes at certain distances from the camera.

The LOD system will automatically swap our meshes at specified distances.
The system assigns a number to each mesh, zero being the most detailed and further numbers have less detail.
The lowest detail possible is a Billboard asset, which is a flat plane with an image of the mesh as a material.
[LOD (Level Of Detail)]

The image shows an LOD 0 (high delta) and LOD 1 (less detail)




The LOD component handles the swapping of the mesh and its swap distance from the camera
[LOD Group component]



## Scene view Draw modes

We can use the draw modes drop down in the scene view to check how our assets are performing.

A Guide to Scene view Draw modes
Here are some of the more useful options:

- **Shaded** is the default mode
- **Wireframe** mode shows all the meshes as wireframes
- **Shaded wireframe** is a mixture of the previous two options
- **Render paths** shows the type of shading or lighting on an object
- **Overdraw** shows the overlap of meshes in the camera view
- **Mipmaps** show the ideal texture size of an object
    - Red is too large, blue is too small

## Links

Occlusion Culling
Occlusion Culling settings window

Static GameObjects

LOD (Level Of Detail)
LOD Group component

Scene view Draw modes