Autosetup

Describes how the automatic setup function from QuickLod works.

More information's:

Website:	http://tirelessart.ch/en/quickcode/quicklod/
Forum:	http://forum.unity3d.com/threads/237196-QuickLod
Demo:	http://tirelessart.ch/en/quickcode/quicklod/demo/

Introduction

There are currently two LodObject components. Both components have a different setup:

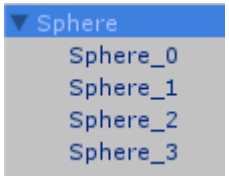
- LodObjectReplacement
- LodObjectMesh

LodObjectReplacement

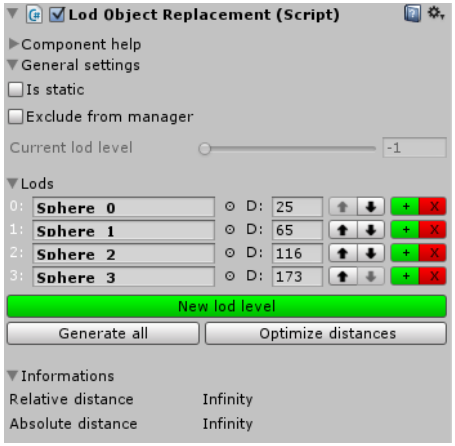
The LodObjectReplacement offers you great flexibility but uses more resources than the LodObjectMesh.

It works with child objects, which are activated/deactivated depending on the distance to the nearest camera.

LodObjectReplacement - Setup



Create a hierarchy where all lod level are children from the same parent.
All lod level should have a name with the lod level indexer (number) at the end.



Add the LodObjectReplacement component to the parent (Here: “Sphere”)
The component will automatically take all children with a number at the end and reference them in the lod section (ordered by the number)

Also for each level, a distance will be generated.
This generated distance is based on:

- The size of the object
- The resulting screen space of the object

You can always generate the levels again, or just optimize the distances.
When you did everything correct, you don’t need to do anything inside the component.

LodObjectMesh

The LodObjectMesh has a more complex setup automation. For you it’s just more intuitive, as it offers you multiple way’s to store your data.

It works with mesh replacement in the MeshFilter and thus needs no additional child objects.

LodObjectMesh - Setup

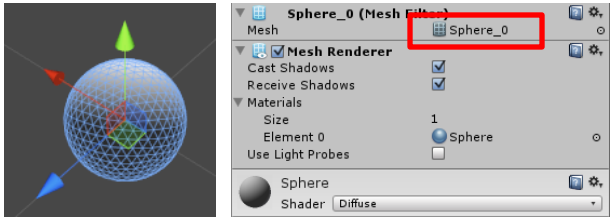
There are two possibilities how you can provide the needed meshes:

- All meshes in one file
- All meshes in multiple files

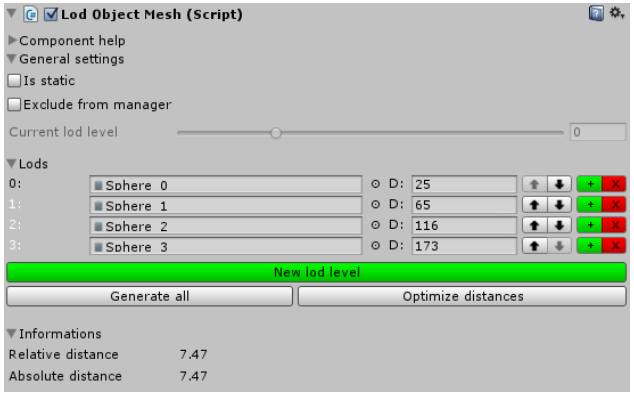
When you have multiple meshes in the same file, QuickLod assumes, that you have ALL meshes in this file!

All meshes in one file

This is the case when you have something like this, all meshes in the same file



Create a single game object, with a MeshFilter and a MeshRenderer. It should be look like this.
Ensure, that one of the meshes is referenced in the MeshFilter. (Here: “Sphere_0”)



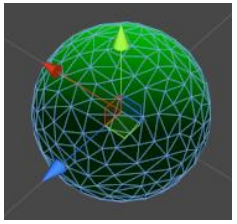
QuickLod will now take all meshes from the group, which have the same base name (name without number) and have a number at the end. With these meshes, it will setup your component.

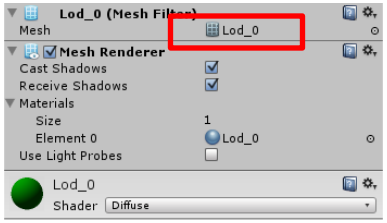
The distances are generated by the same way as in the LodObjectReplacement component.

You can always generate the levels again, or just optimize the distances.
When you did everything correct, you don’t need to do anything inside the component.

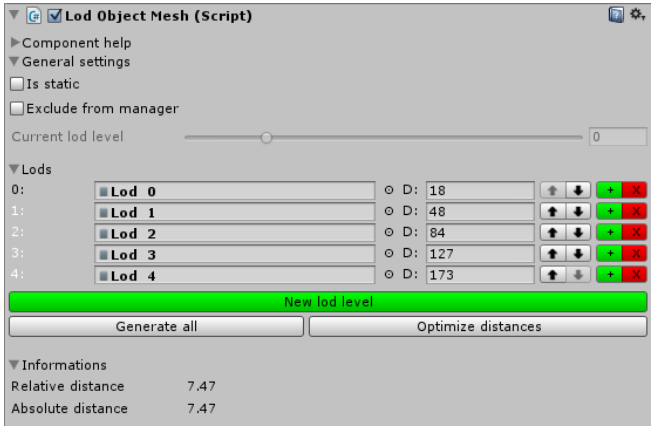
All meshes in different files

This is the case when you have something like this, all meshes are in different files.





Just drag and drop one of the files into the scene. The resulting game object is already usable and look like this.
Make sure that the mesh is referenced in the MeshFilter.



QuickLod will now take all files, which have the same base name (name without number) and have a number at the end. From this files, it will take the mesh and setup the component.

The distances are generated by the same way as in the LodObjectReplacement component.

You can always generate the levels again, or just optimize the distances.
When you did everything correct, you don't need to do anything inside the component.