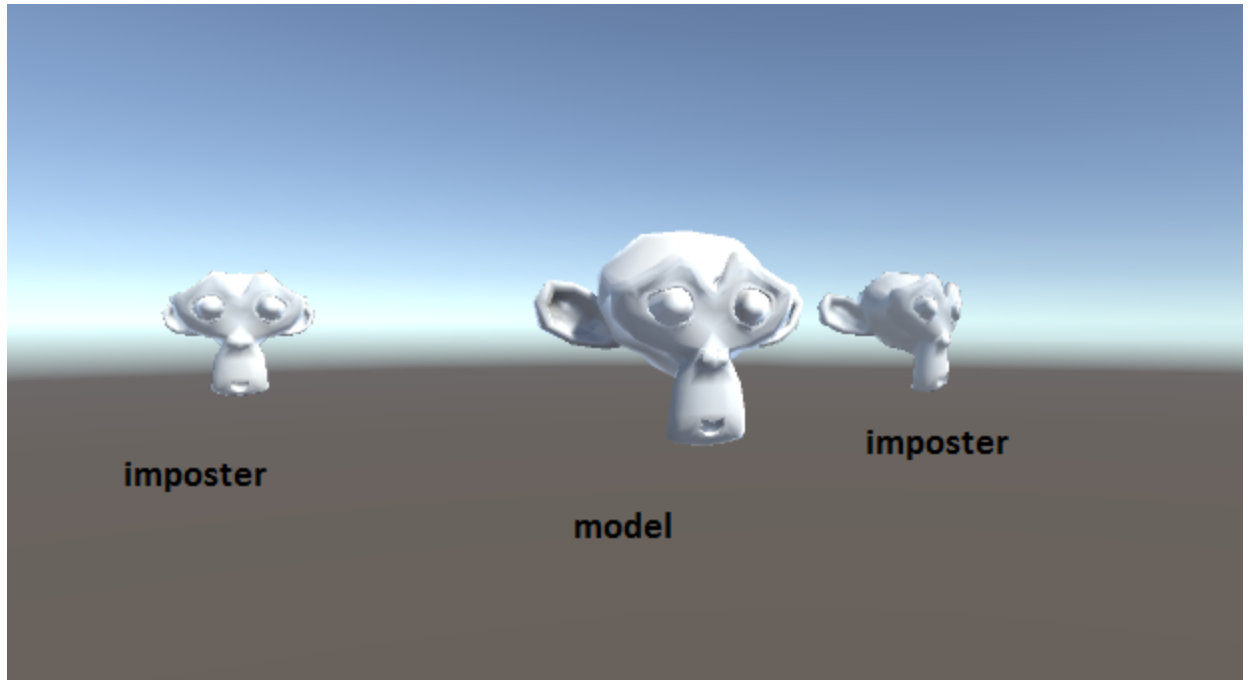


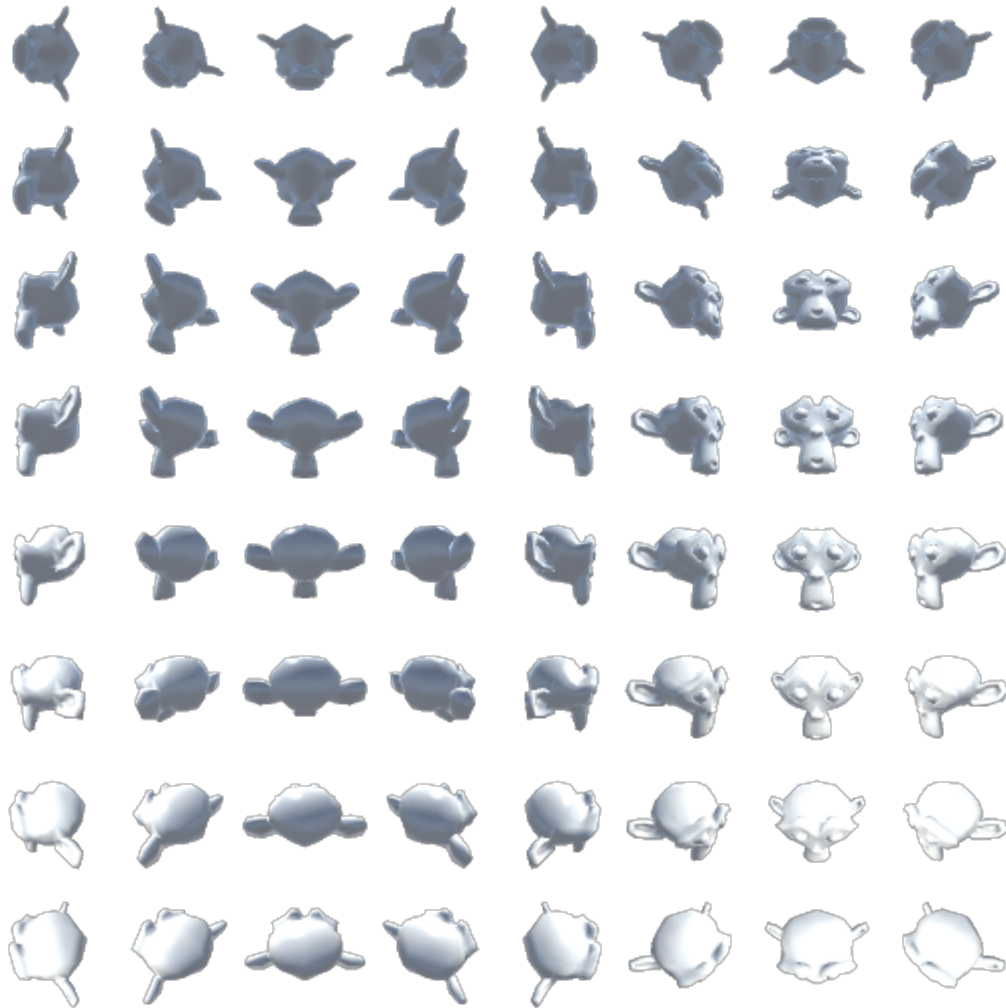
Simple Imposters for Unity3D

1. Overview

Imposters are advanced billboards, that are not only quads that always face the camera but also depending on viewing angle display proper texture. On the image below you can see how they work together with LOD system in engine.



When model is close enough to camera its being displayed normally, however when it gets far away it's being swapped by LOD system for an imposter billboard. According to the viewing angle proper part of tiled texture will be displayed that will mimic the look of an complete model.



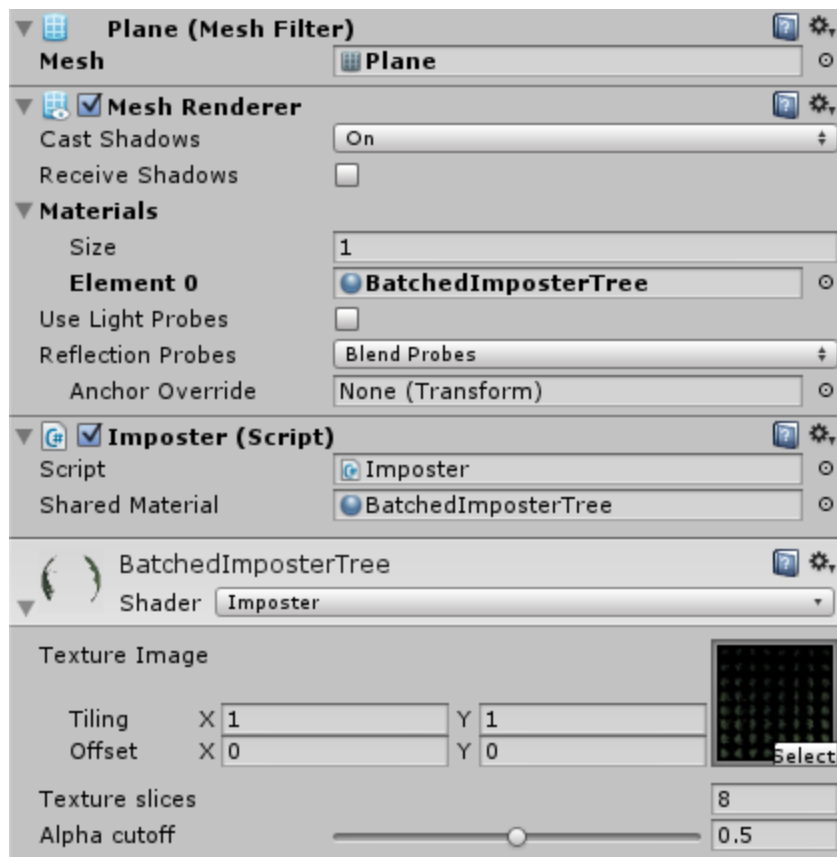
Exemplary imposters texture

2. How to

The only thing you have to do to setup an imposter is first to generate texture for it. To do that open **GeneratorSetup** scene, and place object you want to use at world center position. Then Under the **Generator** object adjust camera distance (Z position) to capture the entire object. Next on Generator object, there's a script called **ImposterTextureGenerator**. Just leave the properties as is and hit Generate button. After generation if finished Unity will show you the texture generated in Assets folder.

In **ImpostersGenerator/Models** ther's a **Plane** model prepared to work for this system. Just simply set it for **MeshFilter** on your object. Create Material with generated texture and set its shader as **Imposter**. Next set material in MeshRenderer and add **Imposter** script to that object as well as set **Shared Material** property to that

material. (you can always check in **ImpostersGenerator/Prefabs/TreeTest.prefab** for proper setup or see image below).



In the **ImpostersManager** script there's a property called **maxObjectsToUpdatePerFrame**. It determines how many Imposters will be updated during one Unity Update method call. The less the better for performance, but for many objects or games where camera moves fast Imposters could not update often enough.

Best way is to open Demo scene and check how it works and how it is setup.

[Demo contains only imposters, however they could be setup as part of LOD group that is only visible far enough]

3. Under the hood

When running the game Imposters will be added to **ImpostersManager** that will perform update calls on chunks of them instead of calling **Update** on each object individually (optimization).

Each Imposter during update will rotate to face camera and then calculate (according to viewing angle) the proper Material from look up materials table.

By default system works with 8x8 view images capturing objects from bottom to top, 360 degrees to make it more universal not only for trees but all other objects. During start $8 \times 8 = 64$ materials are created for imposters to work with batching. That and calculating everything in script rather than shader was the only way I could find it to work with Unity batching system. For performance reasons you can tweak shader and scripts to work with less viewing angles which in turn will produce less materials. You could also move script calculations to vertex shader if you e.g. don't want to use batching, updating scripts the way I did it, but would like to use imposters for some high quality models using only shader code *[because of batching billboard will not work, passing some values from script like position of object will also break batching]*.