

# ABI: Automatic 3D Billboard Imposters

---

## About

### What is a Billboard?

A billboard is simply a 2D image that replaces a 3D model, and always faces the camera. While very efficient, this suffers from rotation artifacts.

### What is an Imposter?

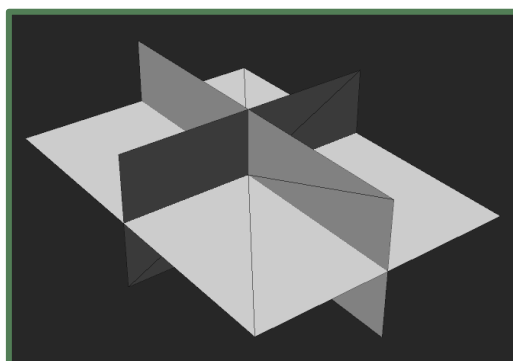
An imposter is a billboard that uses a tile set instead of a single 2D image. At runtime, the appropriate image is selected according to camera the viewing angle. Traditional imposters use a large amount of texture memory to cover all viewing angles at sufficient tile resolution, which makes them impractical for scenes with lots of unique objects.



<http://philliphamlyn.wordpress.com/2013/04/03/billboards-imposters-and-meshes/>

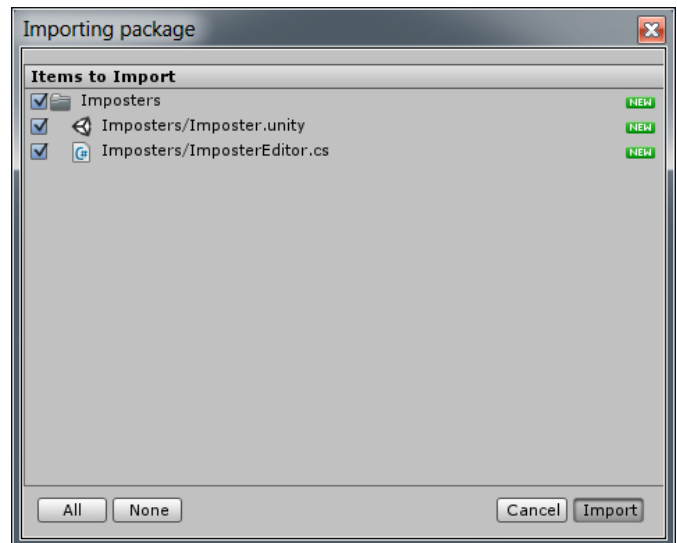
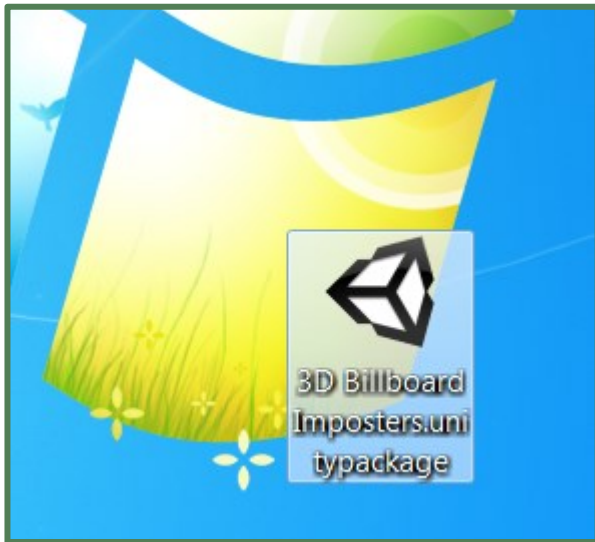
### What is a 3D Billboard Imposter?

A 3D billboard imposter bakes just three views onto three intersecting planes, requiring a much smaller amount of texture memory (left), while still giving excellent results when viewed from a distance. It can be seen from any viewing angle (right) without rotation artifacts. The planes fit the geometry and can be adjusted according to the model requirements:

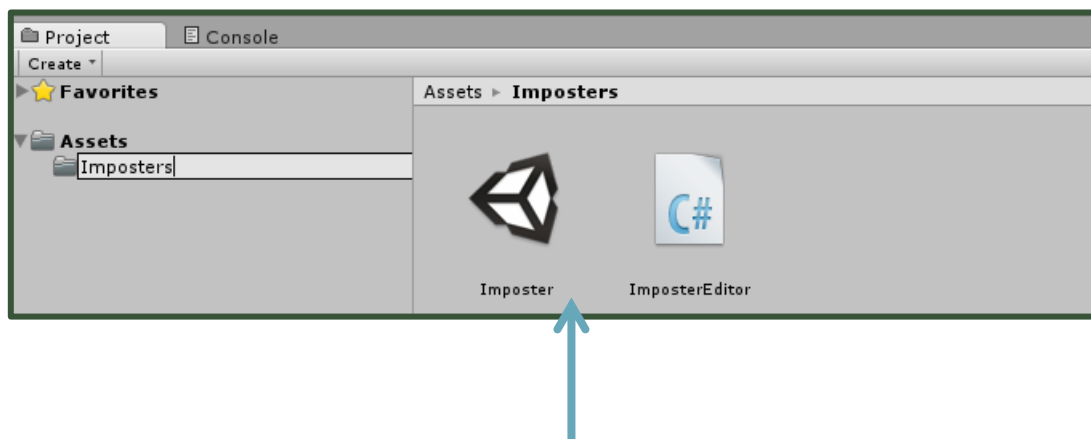


## Installation

Firstly, import the package into your project.

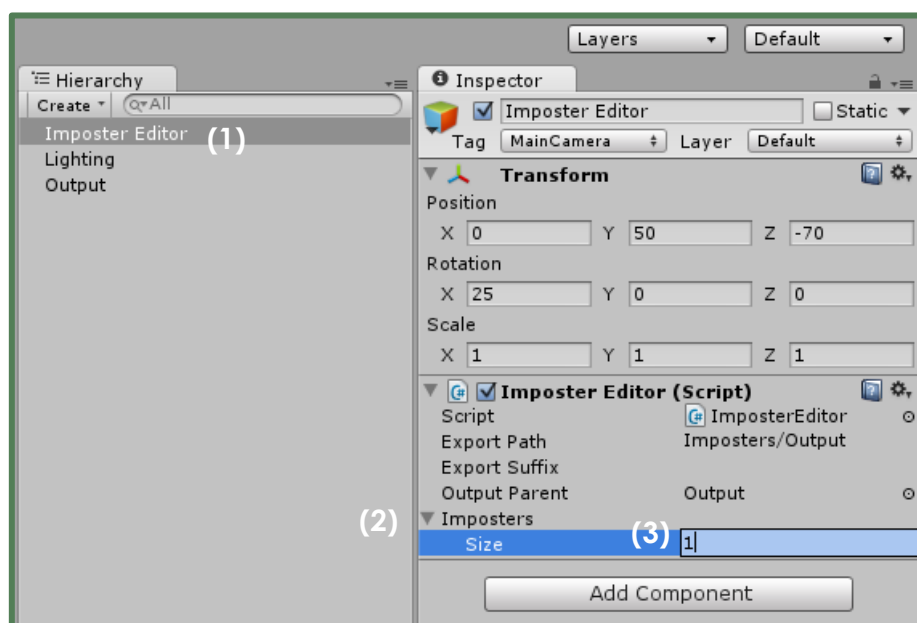


You may now drag the package to a preferred location, or select it and press "F2" to rename it to whatever you like (always move/rename inside Unity3D otherwise you may break dependencies).



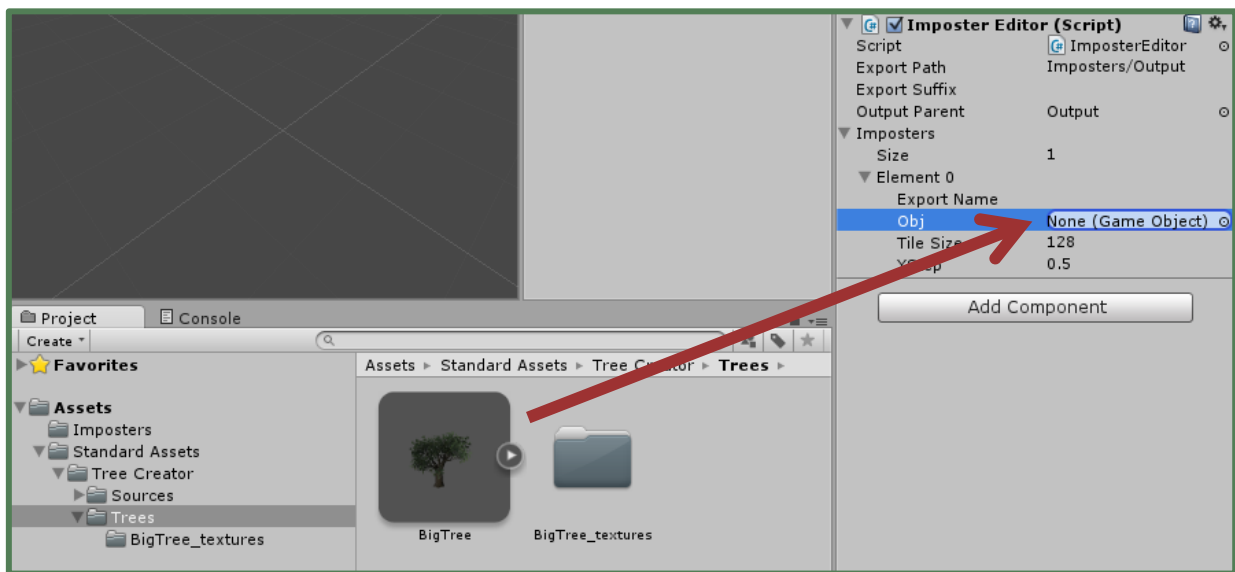
## Setup

To create imposters for your content, load the *Imposter* scene. Now **(1)** click the "Imposter Editor" in the project hierarchy, **(2)** expand the "Imposters" property, and **(3)** type in the number of models that you wish to generate imposters for:



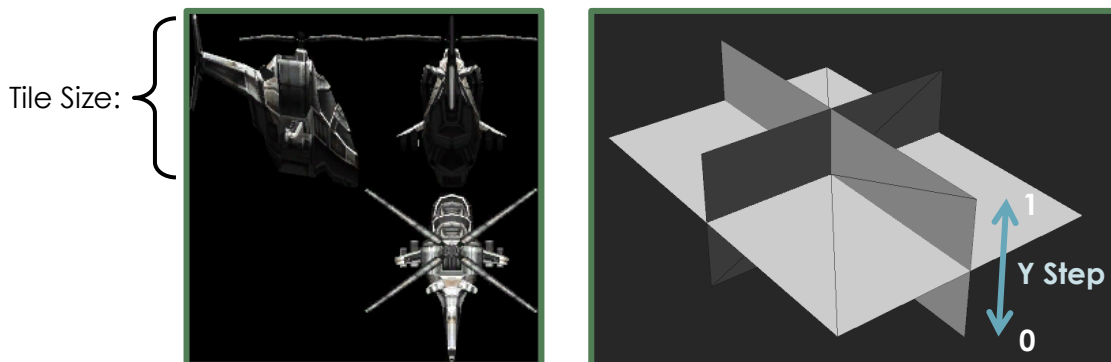
## Imposter Settings

You may now drag a model object from the project explorer into the “Obj” slot, and change the “Export Name” to whatever you like.



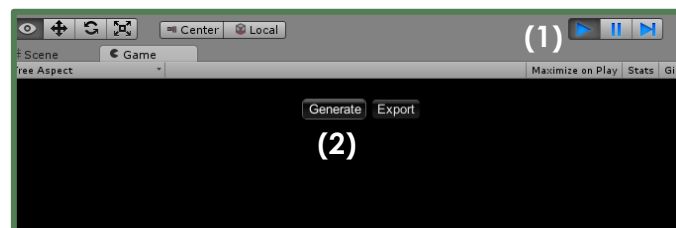
Also, you can set the following parameters for each imposter object:

1. Specify a tile size (pixels) *(32-256 are good default values)*
2. Specify the step of the y-plane *(0..1) (0.5 is a good default value)*



## Generation

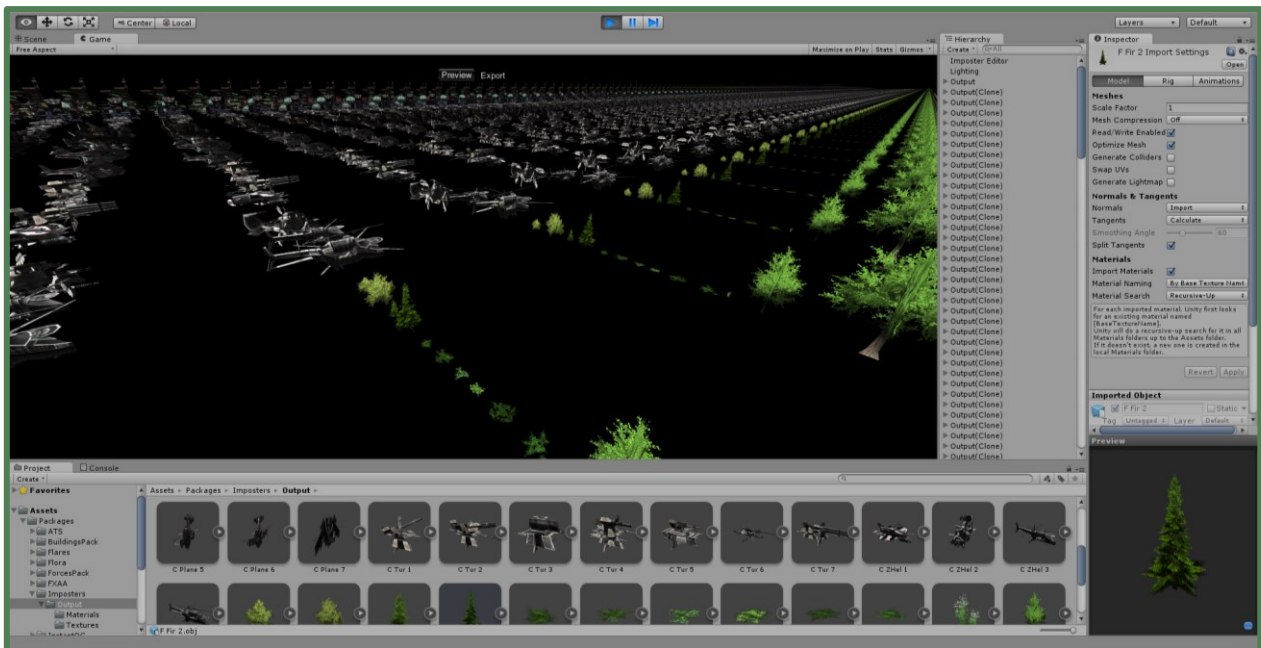
To generate imposters, **(1)** click “Play” then **(2)** click “Generate”:



This will generate imposter geometry which matches the proportions of the input models, and it will map textures for each axis accordingly.

## Preview

After clicking Generate, the “Preview” button will become available. Pressing this multiple times will automatically arrange clones of the generated imposters for viewing and performance evaluation.



## Export

When satisfied with the results, click the “Export” button. This will create the output folder, materials, textures, and link OBJ files for your new imposters. *Do not panic* if the OBJ icons appear to have no textures on them in the project explorer; Unity3D can take a while to build the icon cache. Please be patient with exporting large numbers of objects, as it is doing a lot of internal work!

## Advanced Usage

### Save

Saving the Imposter.unity scene will save the Imposter Settings, allowing you to continue making imposters at any time. Clicking “Export” will overwrite any existing imposters with the updates.

### One imposter for groups of objects

You can easily create imposters for groups of objects: drag the parent GameObject of the group into an imposter slot, as you would any other model, and the script will figure out the rest.

### Shaders and Lighting

It is recommended to use Stippling on both the imposter and the input model. StipplingAlphaTest, StipplingDiffuse, and StipplingVegetation shaders have been provided; simply apply these to your model materials before generation. If you do not wish to use stippling or the provided shaders, you may assign a different shader – such as ["Mobile/Particles/VertexLit Blended"](#) to the imposter in the “void Start()” function of the ImposterEditor.cs script.

## Cleanup

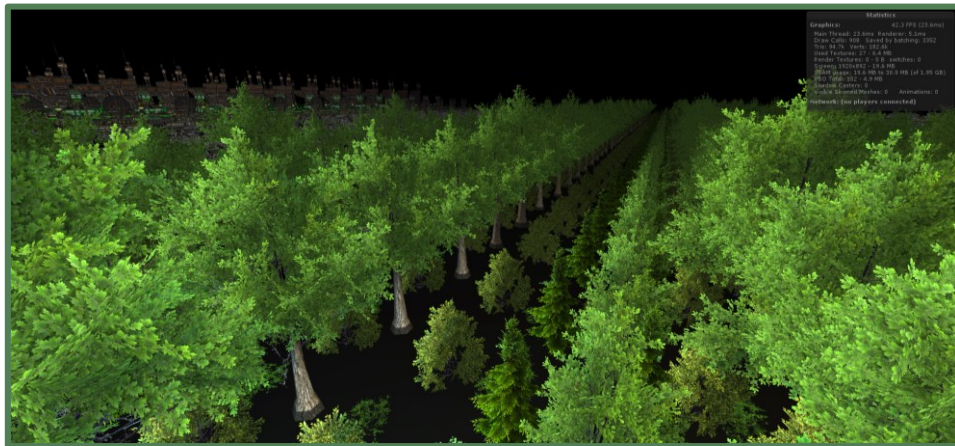
At any time you can delete the output folder, it will be recreated when you next generate the imposters.

## Batching

The imposter pipeline supports efficient batching; even with per-model stippling effects! If you do not see batching working correctly, check that you don't have any long range point lights (dynamic batching requires uniform lighting influencing your objects).

## Automatic LODs

With “Export Prefabs” selected (enabled by default), ABI will automatically generate fully linked GameObjects with the ABI imposter and LOD script automatically prepared for you. It will derive meaningful LOD distances for each object, depending on its renderer bounds, to minimize screenspace error. If you would like to adjust the LOD distances, lower the **lod scaler** parameter (for mobile) or increase it (high-end desktop) depending on your needs.



## Common Mistakes

1. Make sure that the `imposter.unity scene` is always empty or that your objects are disabled; placing active models inside the scene will interfere with the screenshot generation process.
2. If it looks like the materials are broken or not working, check that you do not have duplicate export names (each export name needs to be unique), or that you have entered the correct number of imposters in the size parameter
3. Finally, if you only see an outline of your model, you may need to adjust the alpha cutoff value the generated material files for the "Imposter Clip" shader.

## Careful with Updates!

If you have lots of game objects, it is important to make a copy and rename the imposter.unity scene. This is because an update import will automatically replace the imposter.unity scene file with the newer version therefore destroying all your hard work.

## Bugs / Support

If you have any problems with this project, please email [chris.willcocks <at> gmail.com](mailto:chris.willcocks@gmail.com) and I will try my best to get back to you within one working day. If you experience any unusual behavior, please send a test model. Depending on the sales, more imposter types and texture packing optimizations may be added in the future. Please rate 5\* to support such further development.