

# Billboard Baker

Pro



nexus  soft™

# Table Of Contents

1. Introduction.....	3
2. Installation.....	3
3. Quick Start.....	4
4. Billboard Component.....	15
5. Billboard Baker Pro Editor Window.....	17
7. Addons.....	17
7.1. Atlasing Addon.....	17
7.2. Dynamic Lighting Addon.....	17
8. Current Features.....	19
9. Feedback.....	20

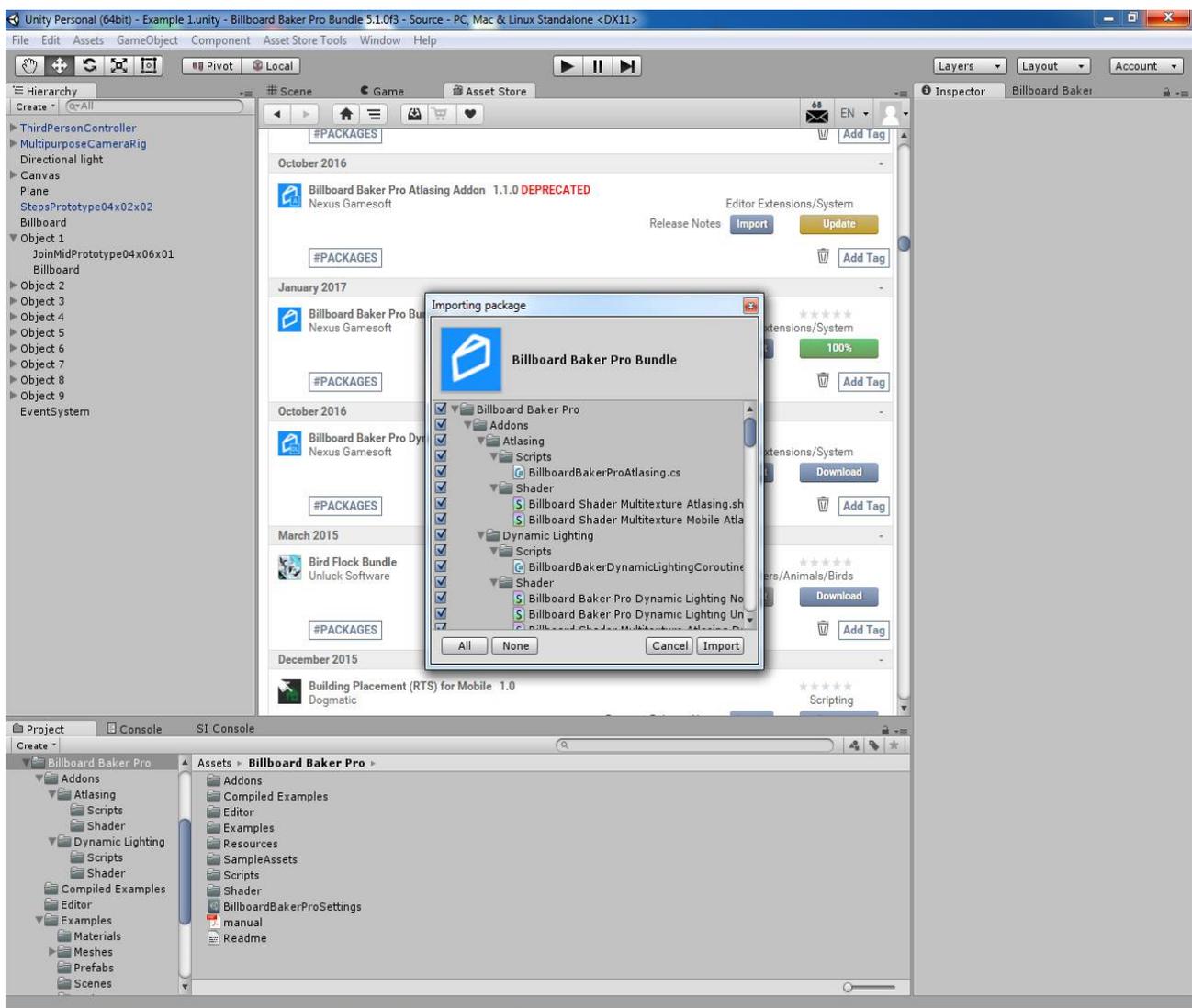
# 1. Introduction

Thank you for purchasing Billboard Baker Pro Bundle and thus supporting the project.

Billboards are a very common technique to retain objects visual appearance in the distance by reducing the mesh topology dramatically. Whereas the first level-of-detail stages usually render the original mesh with reduced faces and vertices each stage, billboards are usually quad meshes textured with a snapshot from the target 3d object. The purpose of Billboard Baker Pro is to achieve a high distance view running with high performance and keeping the objects original look.

# 2. Installation

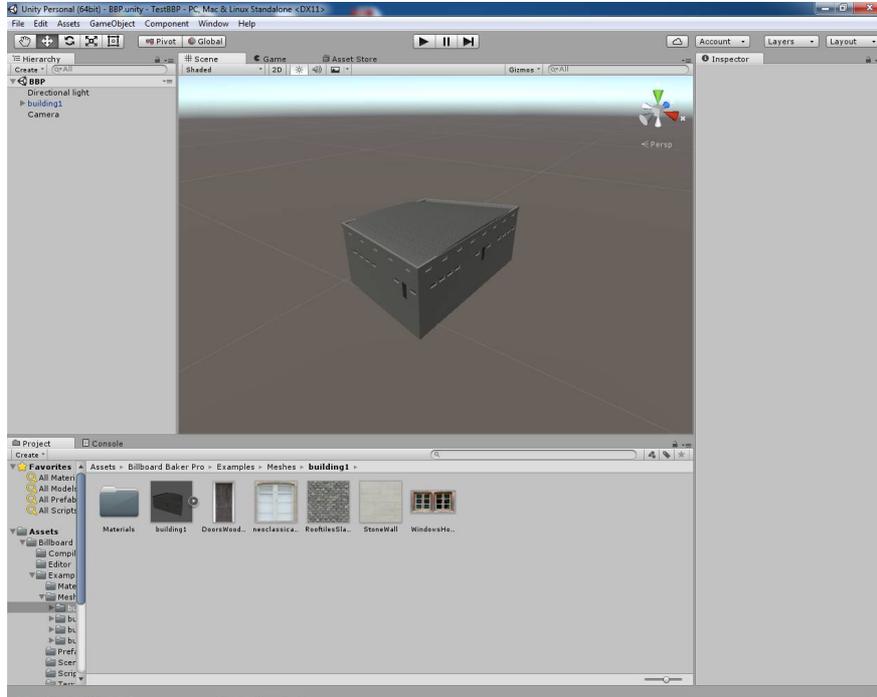
An important goal of this package is to keep usability as simple as possible. To install Billboard Baker Pro Bundle, just import the core package Billboard Baker Pro Bundle into your project. Package will be imported into the folder Billboard Baker Pro by default.



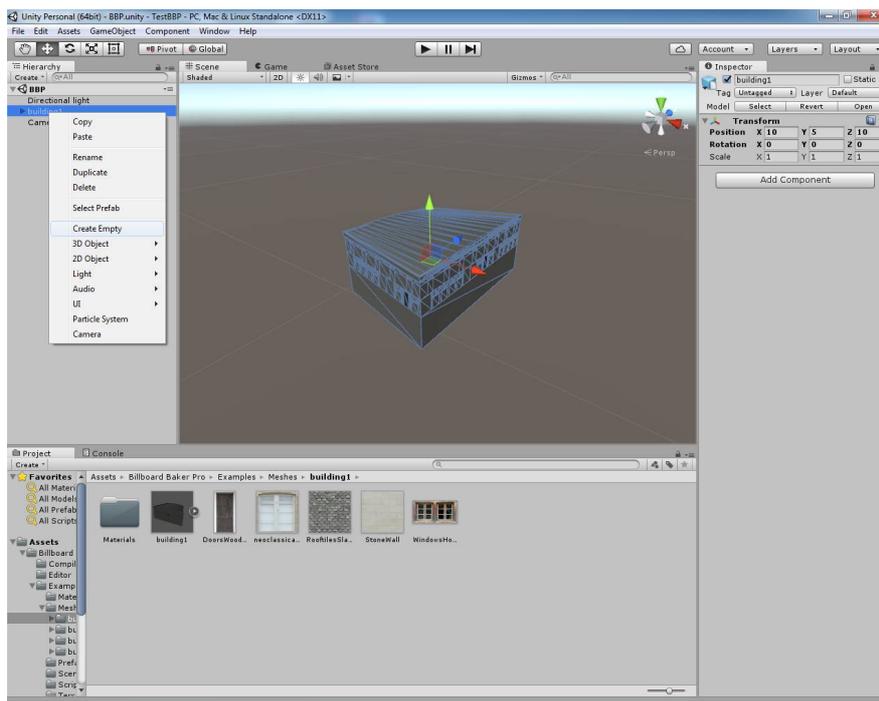
To learn how to use the package, please read the following chapter Quick Start.

### 3. Quick Start

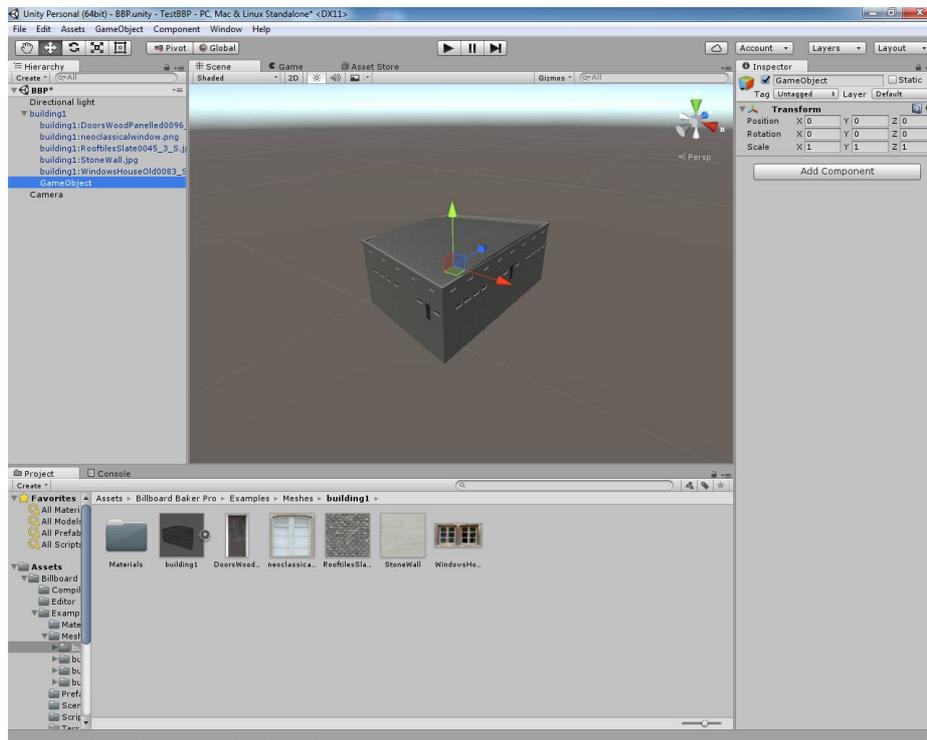
In this Quick Start I will show you how to create a level-of-detail setup. Take any object you want to create a billboard from. In this case we take a house model.



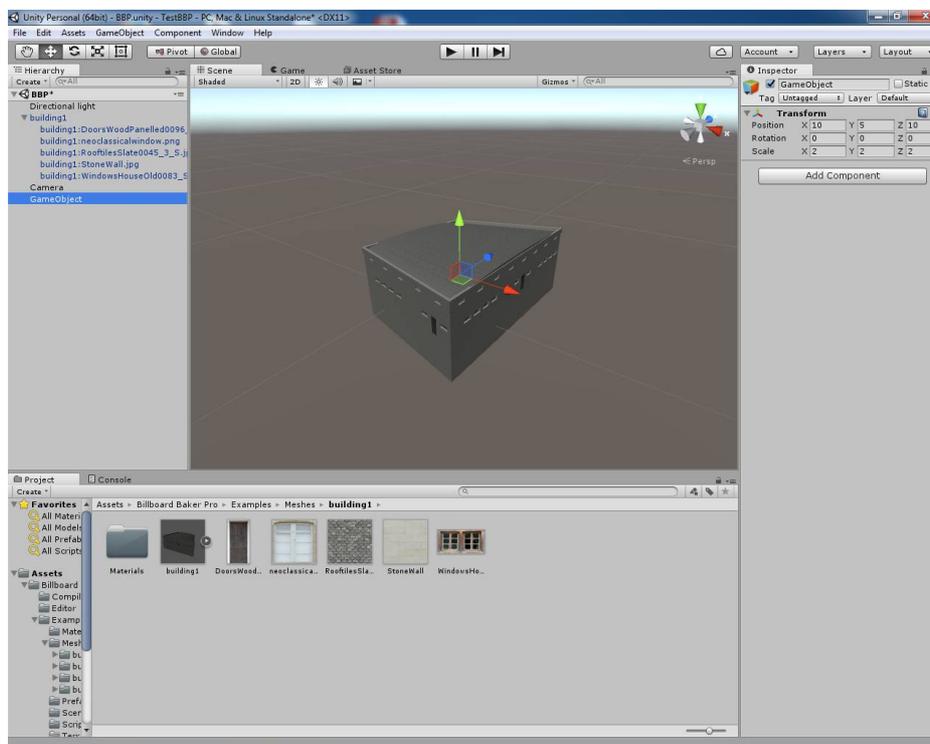
Right-click on your GameObject in the hirachy and press create empty.



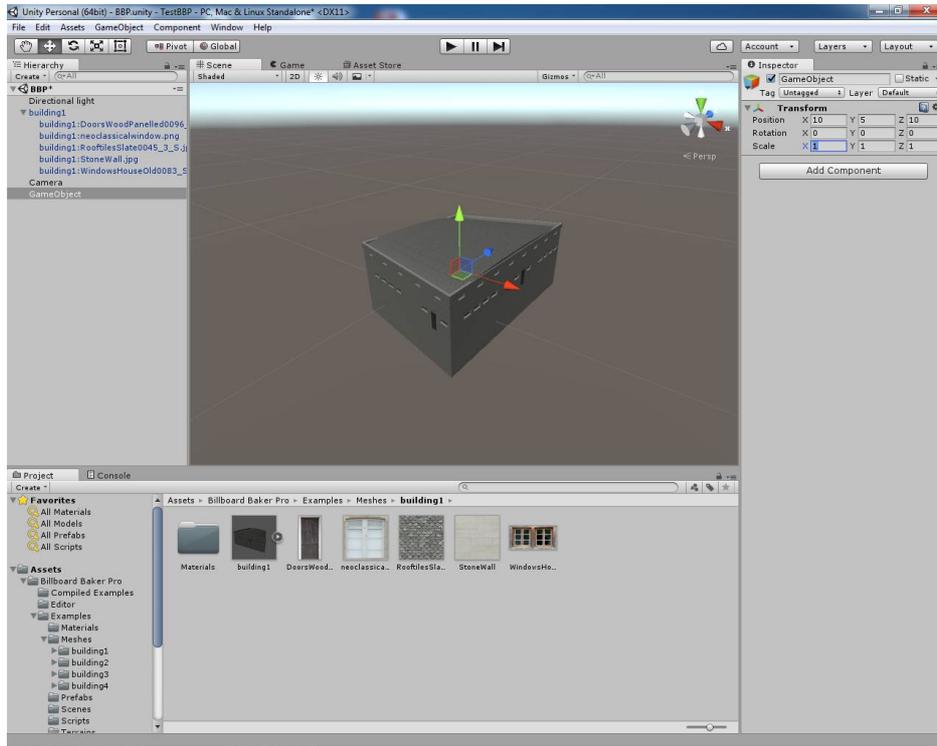
This creates an empty GameObject at root position as child of the house GameObject.



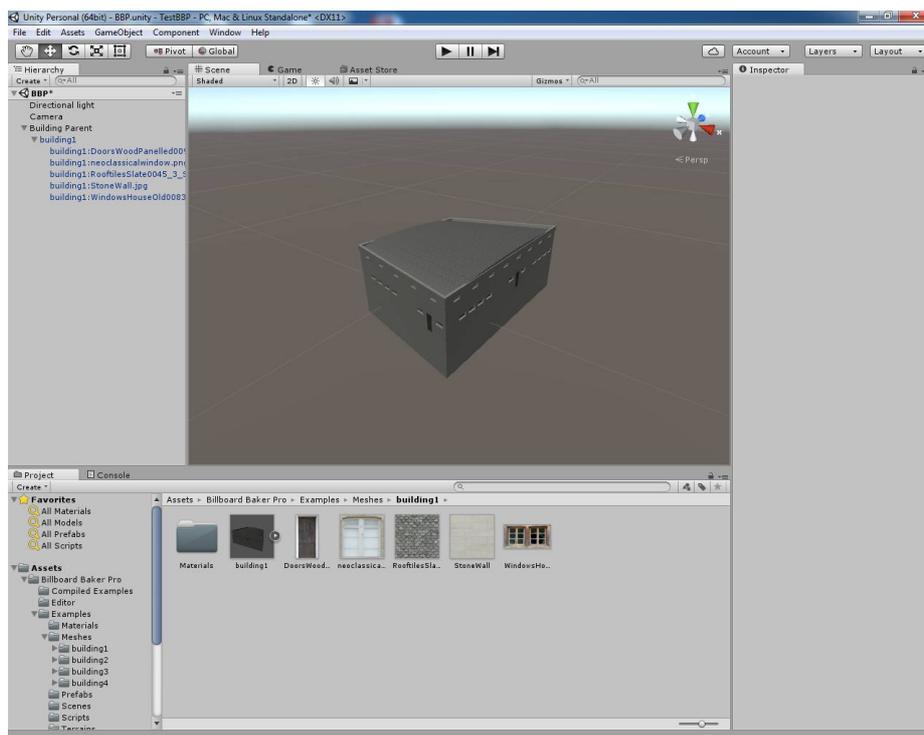
Now we drag the created empty GameObject to the root hirachy.



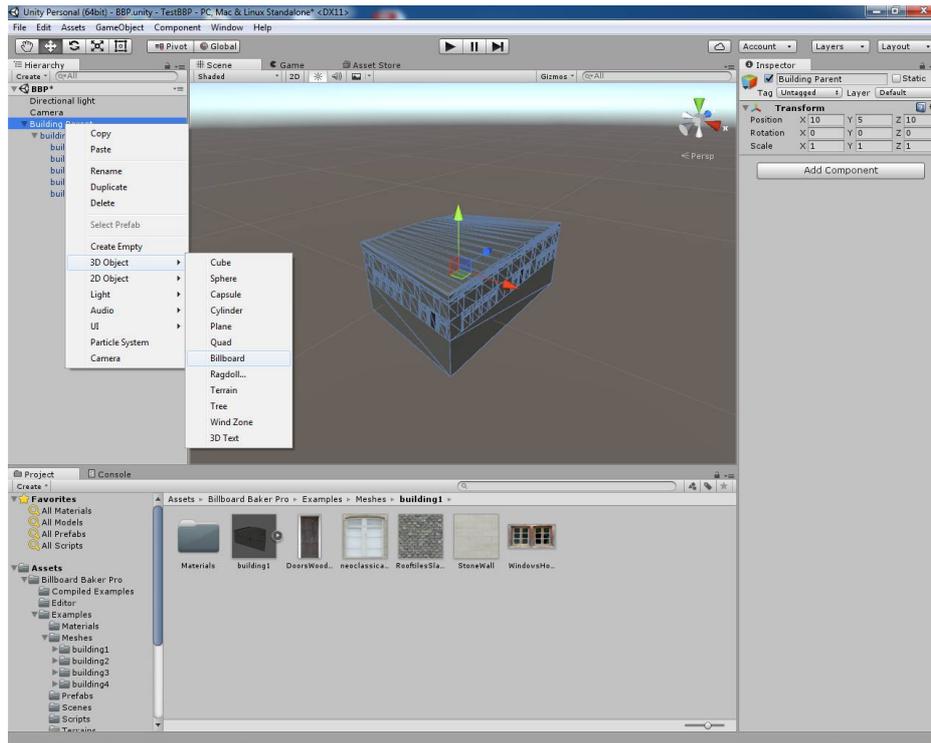
Our empty GameObject has now the same position as our building. It is important, that both share the same position, as billboard won't otherwise bring best results. We also want our new empty GameObject's scaling to be 1 in all directions. This is very important, as our billboard in the end would bring wrong result.



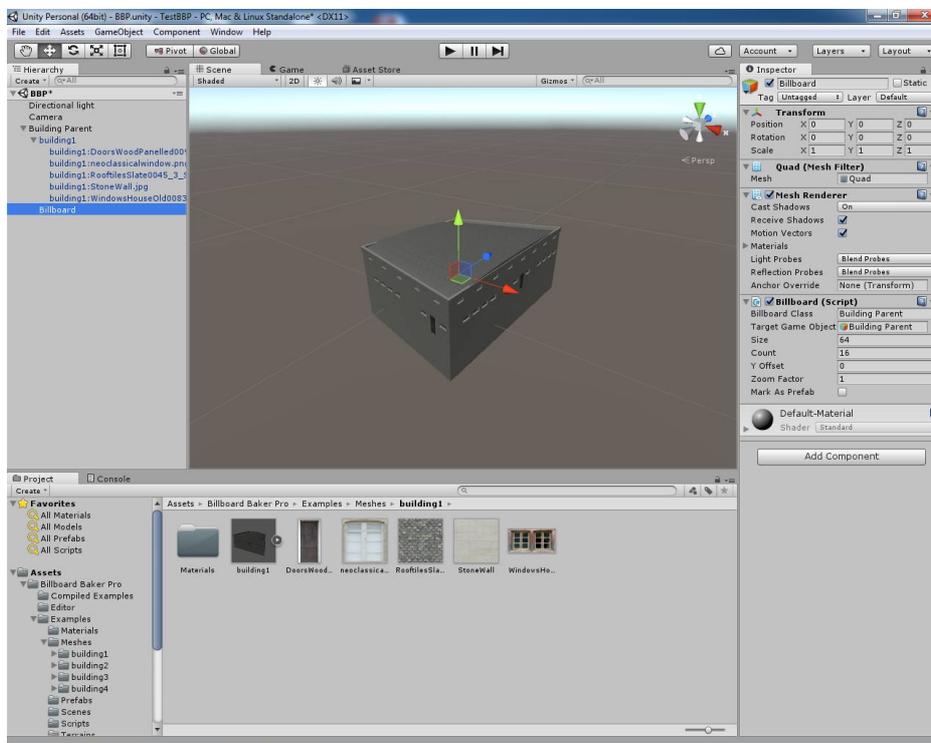
Lets now rename our empty GameObject to Building Parent and make the GameObject building1 child of Building Parent. The scale of Building Parent game object can now be set different to 1, but still needs to be equal in all directions.



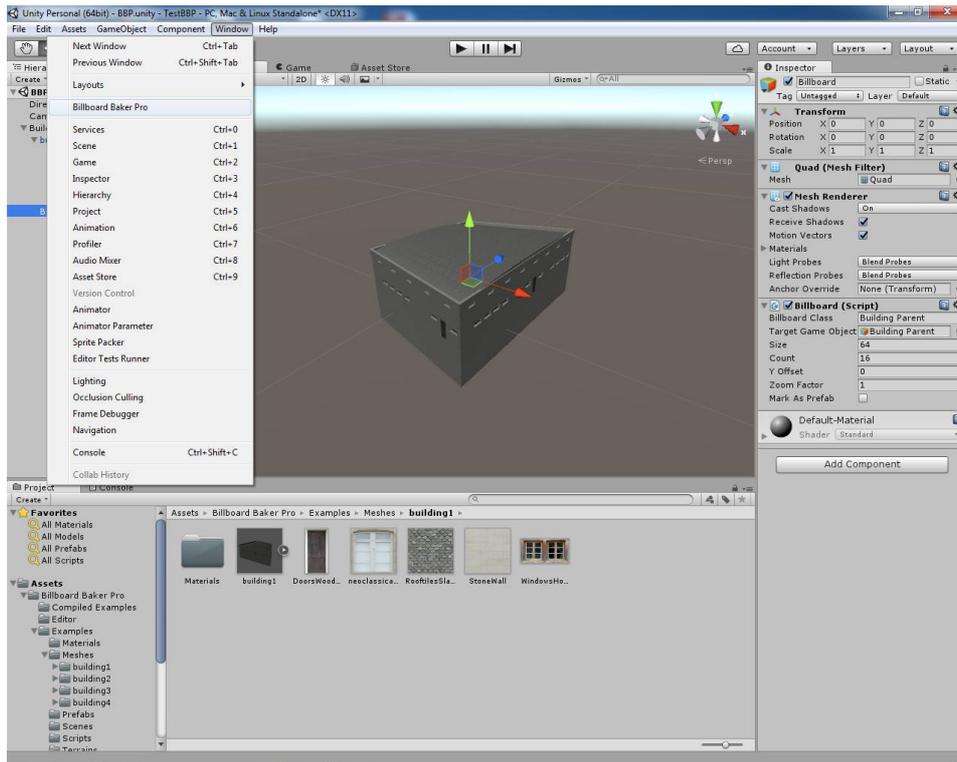
Now let's create a Billboard GameObject. Right click on Building Parent and press 3D Object → Billboard.



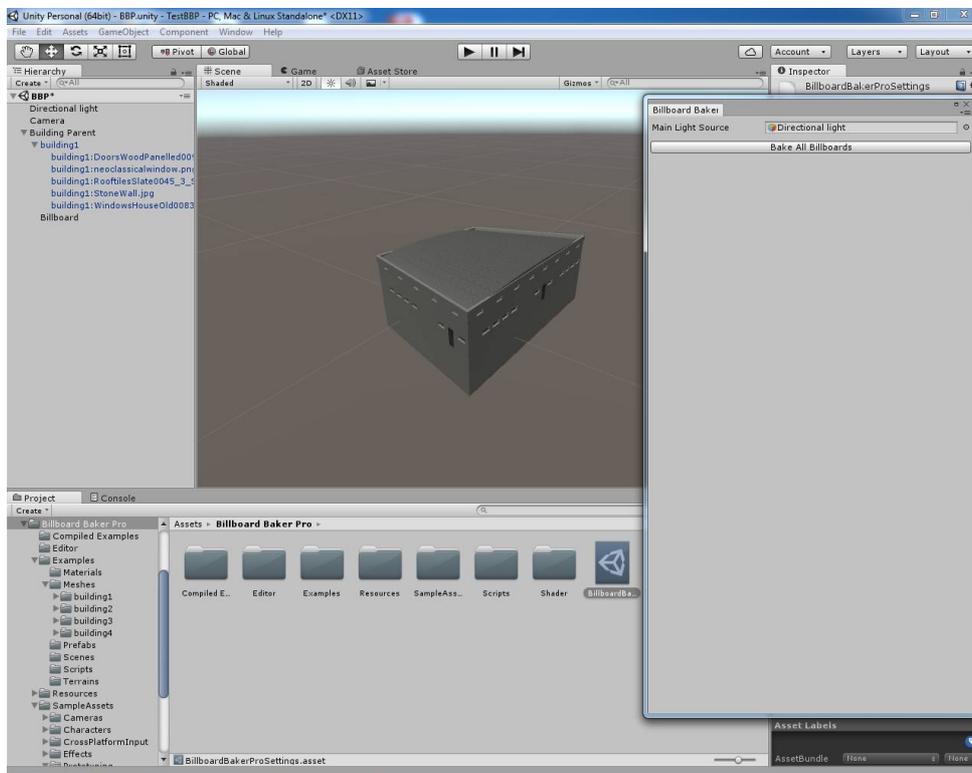
A new Billboard GameObject will be created as child of Building Parent. If you want to know more in depth about the Billboard Component, please move to chapter Billboard Component.



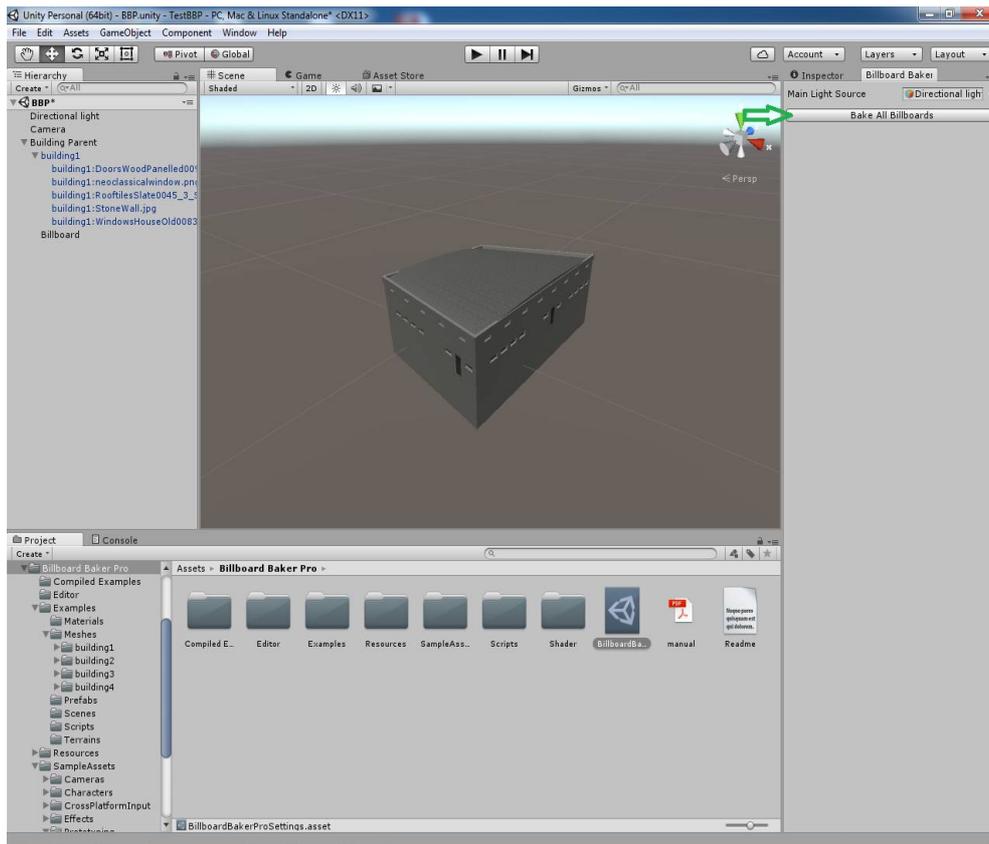
Let's now open the Billboard Baker Pro Editor Window by clicking Window → Billboard Baker Pro.



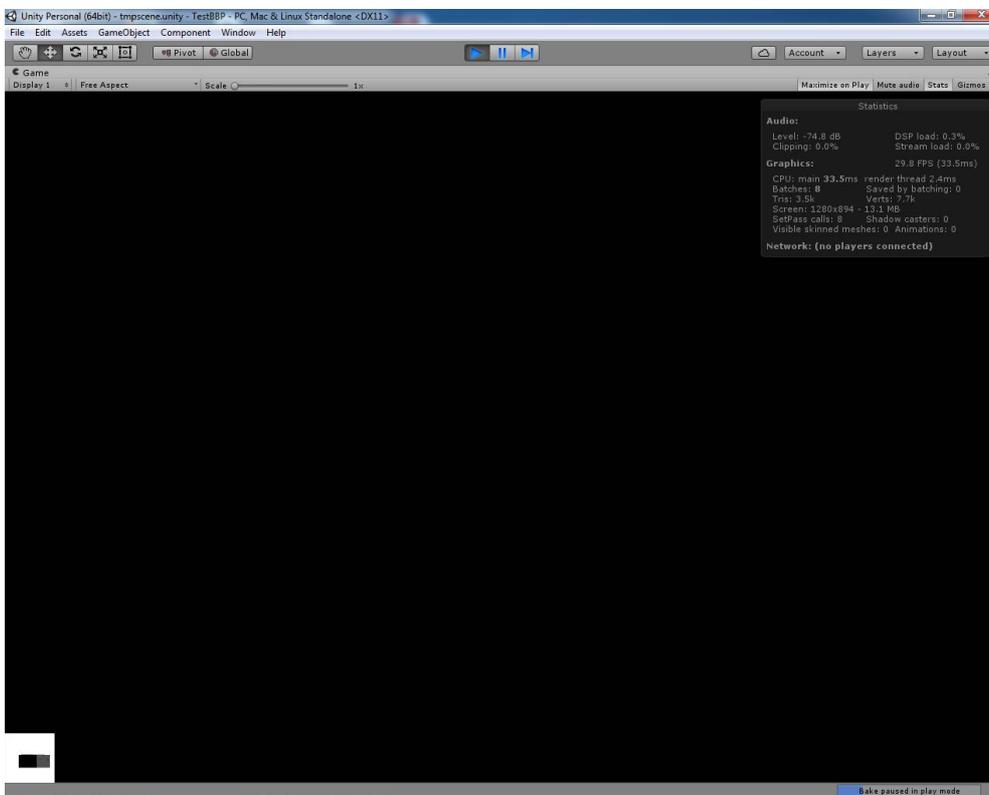
A new window will open on the right side. To get more indepth information about it, please go to chapter Billboard Baker Pro Editor Window.



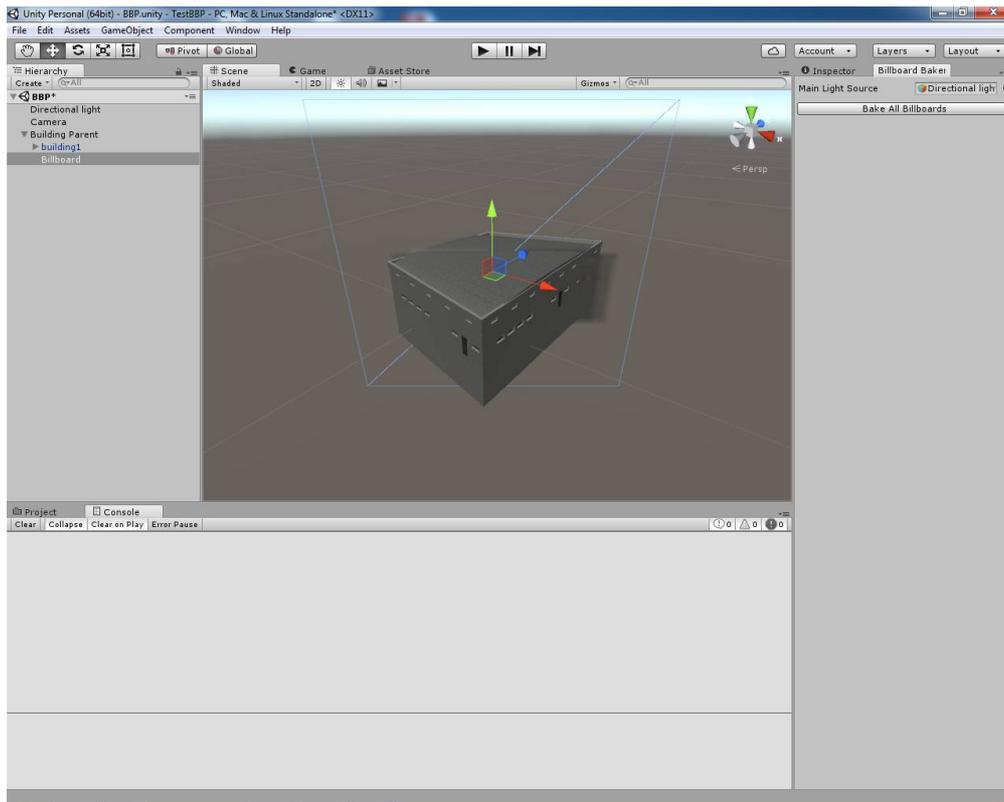
Click Bake All Billboards.



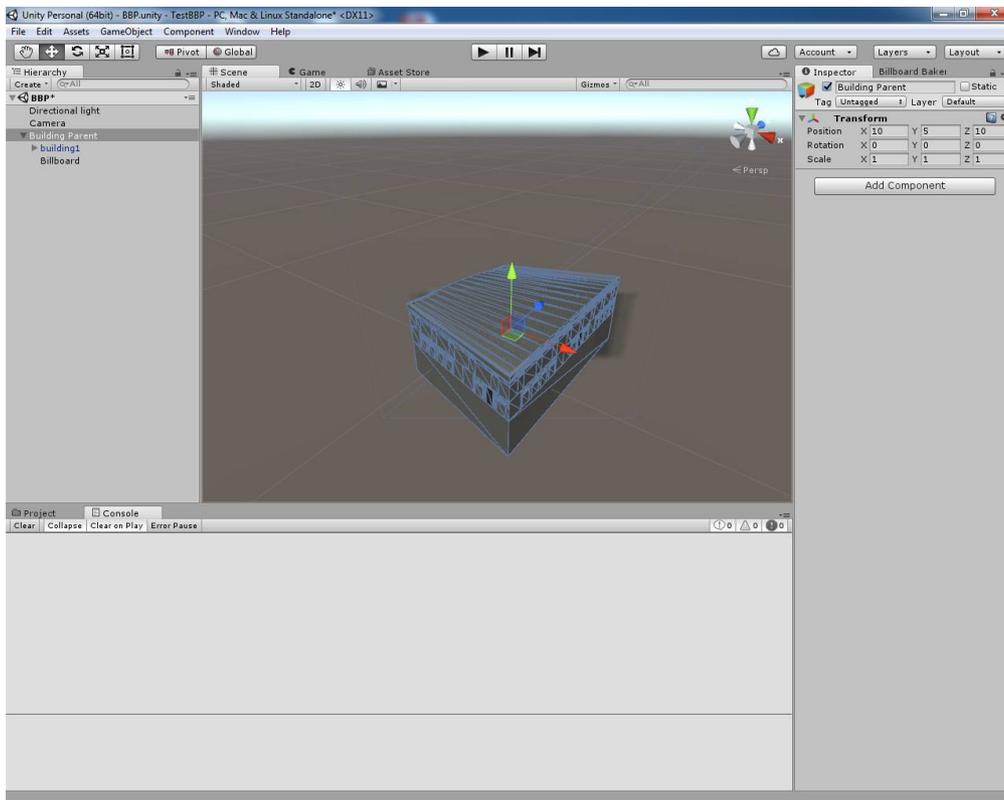
The baking process starts now.



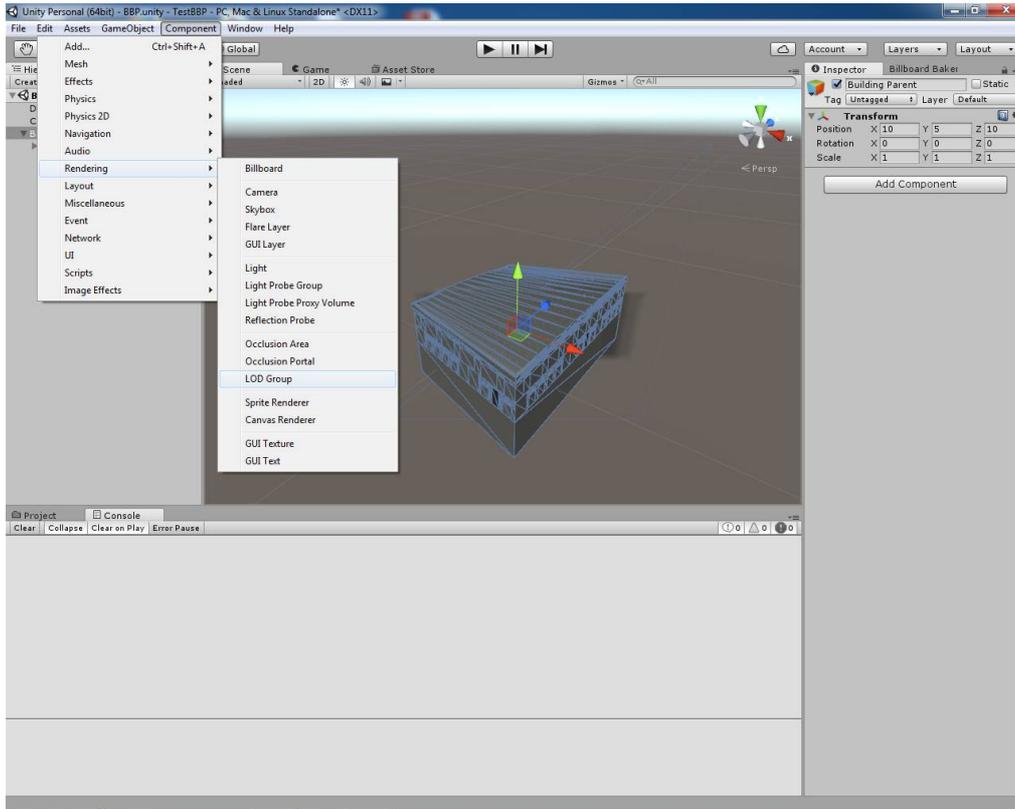
After baking, you will see the building and the billboard at the same time.



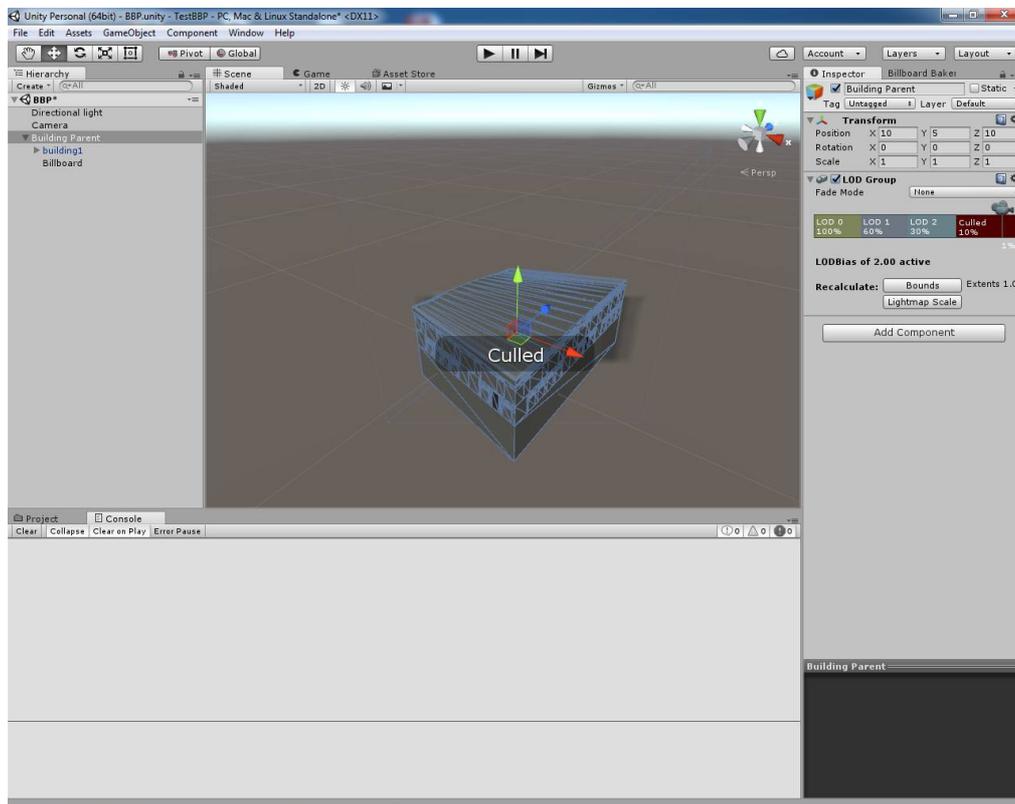
We now create a level-of-detail component, by selecting Building Parent Game Object first:



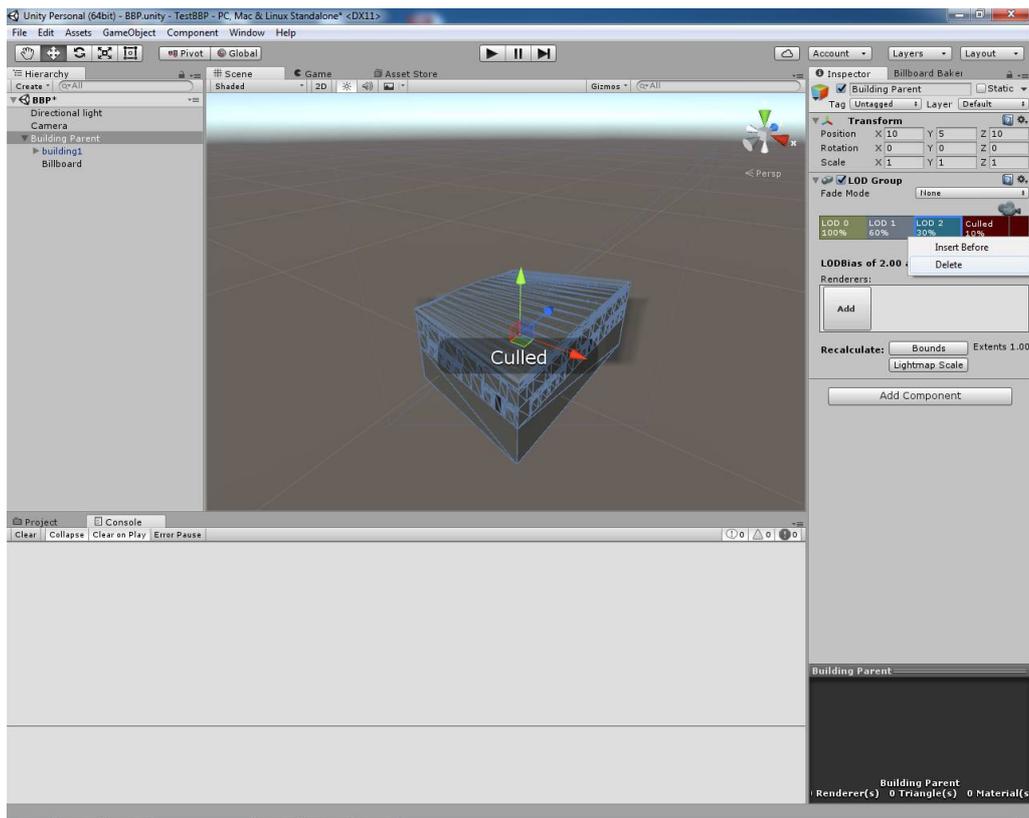
And then clicking on Component → Rendering → LOD Group.



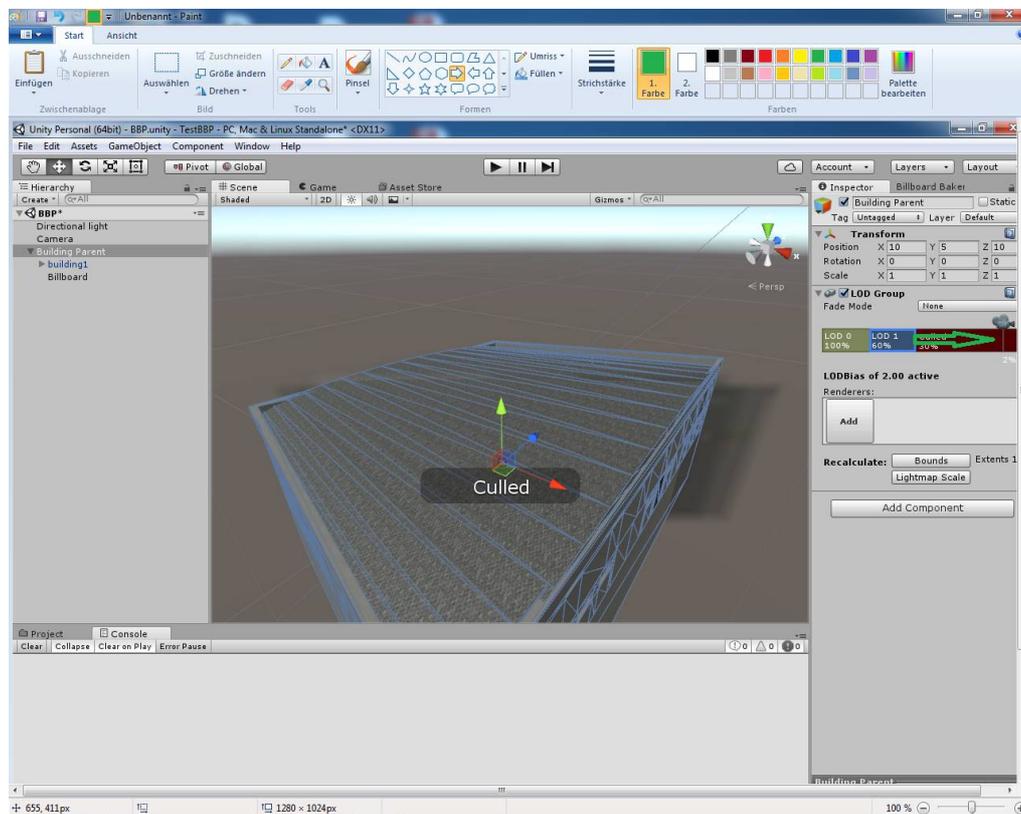
A new LOD Group component will be added.

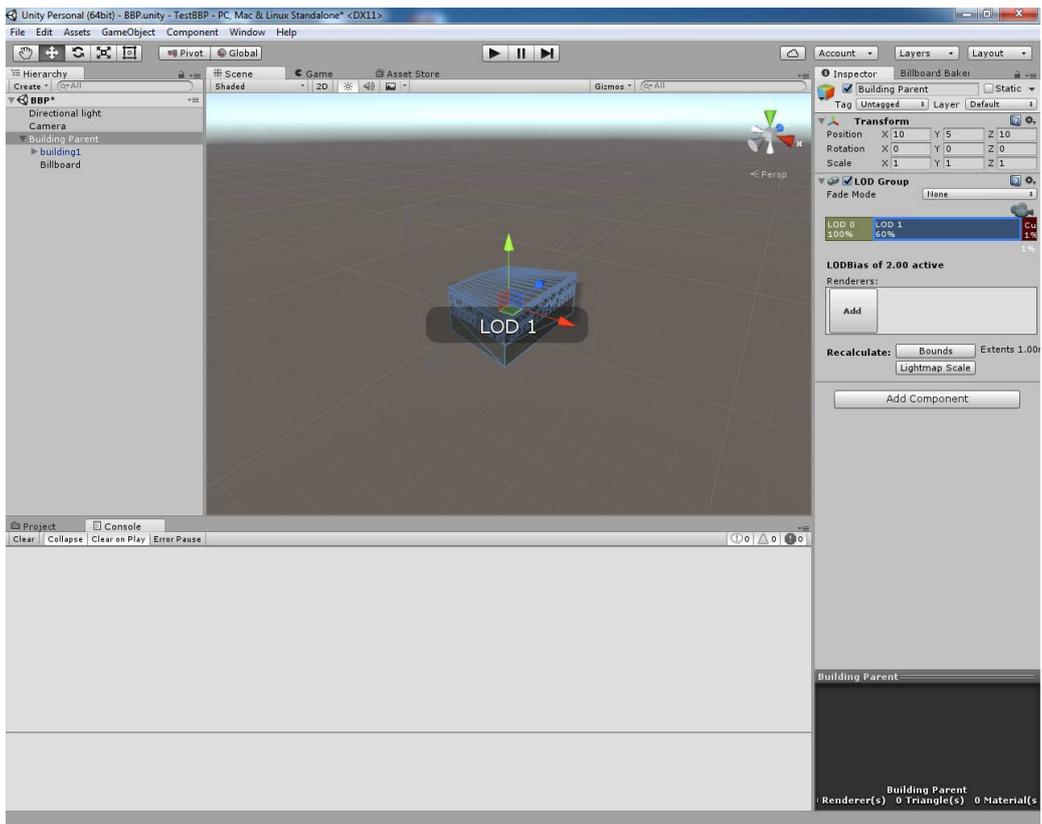


Right-click on LOD 2 stage and press delete, as we don't need it here.

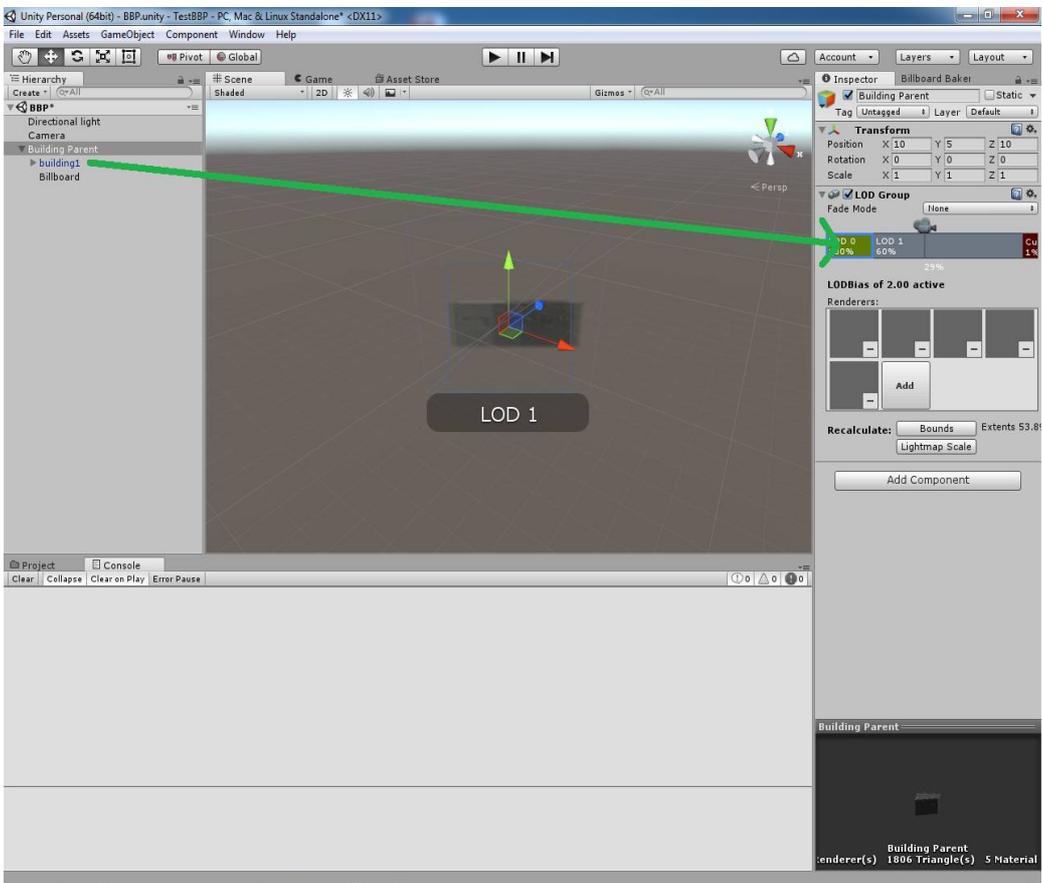


Move culled border (line between LOD 1 and Culled) to the right, till culled region is around 1%.

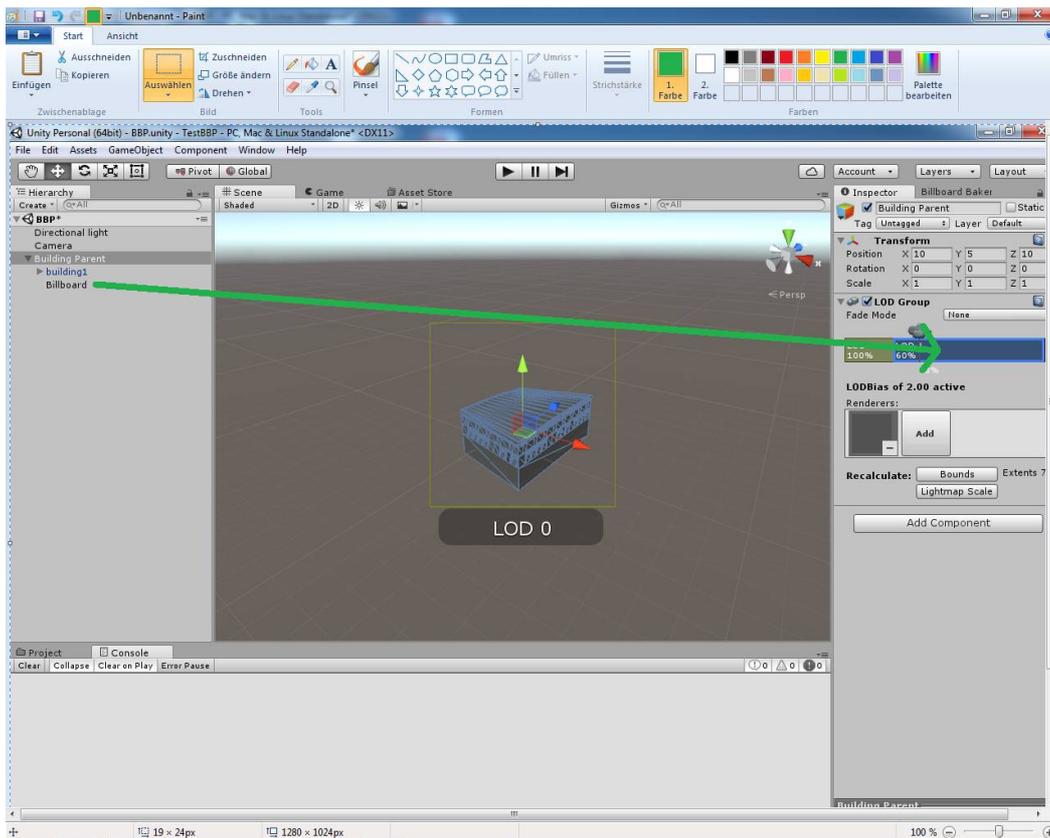




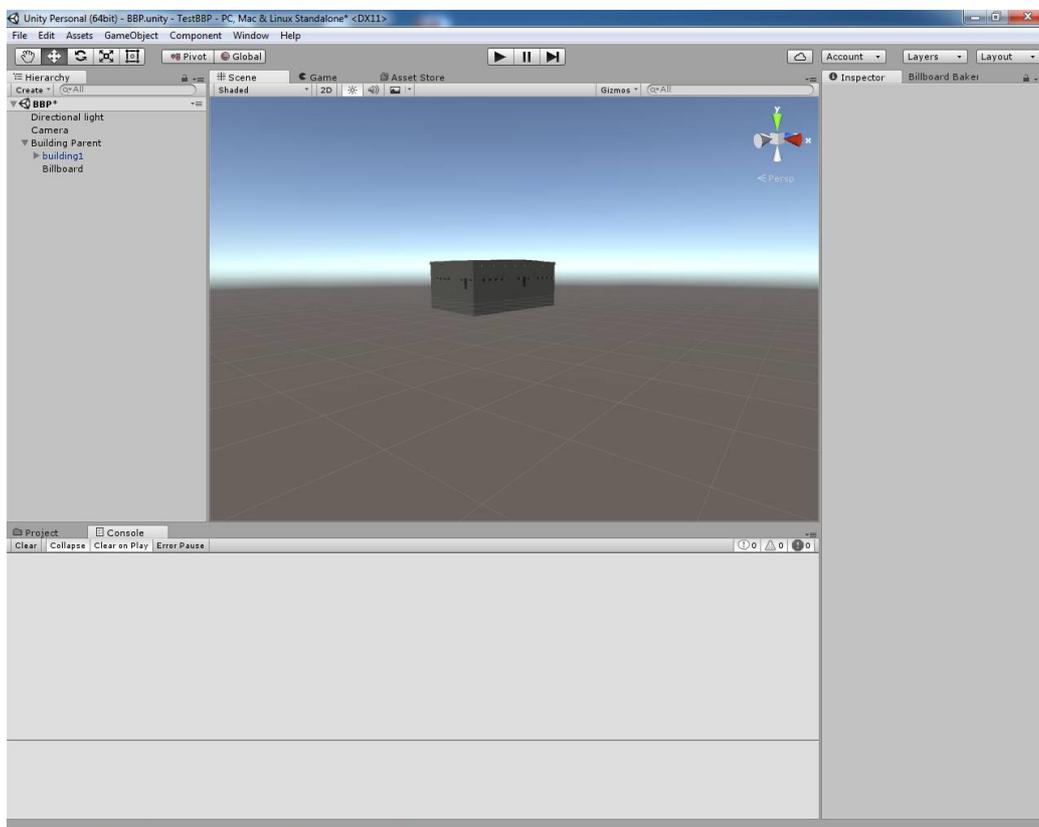
Now Drag&Drop building1 GameObject to LOD0 rectangle. All renderers of building1 will now be visible in this stage.



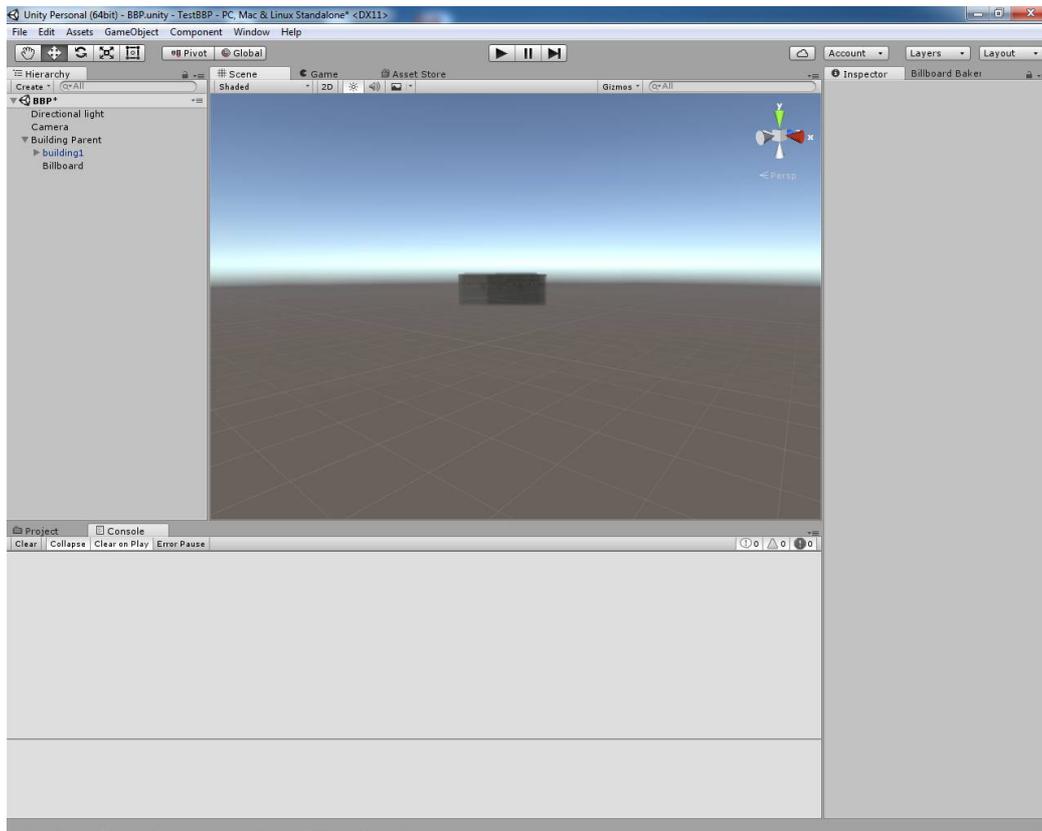
Do the same with Billboard GameObject and LOD1 rectangle. Drag&Drop Billboard GameObject to LOD1 rectangle. All renderers of Billboard will now be visible in this stage.



Deselect any GameObject. Your LOD Group is now setup correct. If you want to change billboard settings, please go to chapter Billboard Component. After applying changes, you need to Bake the billboards again. Here you see LOD 0 stage.

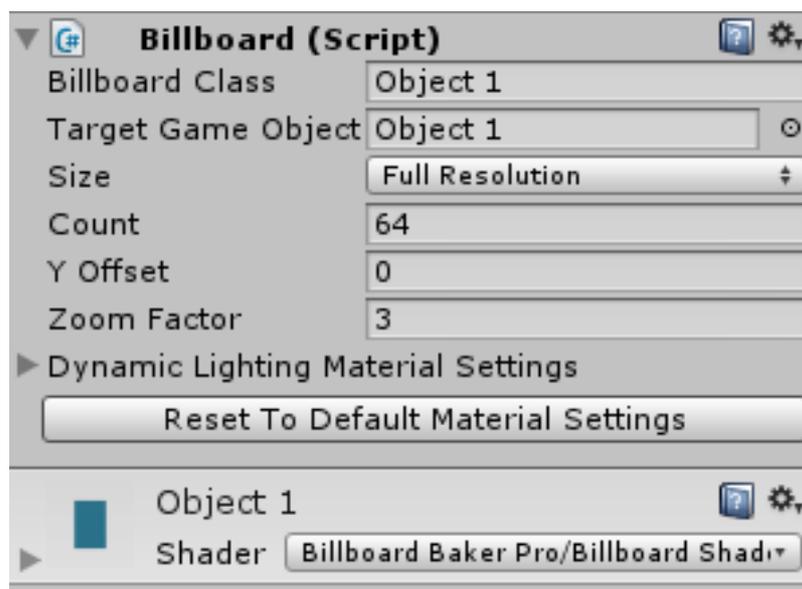


Here you see billboard in LOD 1 stage.



If you have any questions, please contact us at [mail@nexusgamesoft.com](mailto:mail@nexusgamesoft.com).

## 4. Billboard Component



<b>Variable</b>	<b>Description</b>
Billboard Class	A string defining a class of billboard. Each class is processed during bake. You can have many billboards of same class in the scene. They all need to be exact duplicates. Target Objects must be duplicates and may not be different, especially in scaling, rotation and content. Else you will not get proper billboards (wrong scaling, rotation and content not matching).
Target Game Object	The GameObject you want to make a billboard from. Target GameObject is processed during bake.
Size	Size of all frames, relative to Max Texture Size setting in Billboard Baker Pro Editor Window.
Count	Count of frames the billboard has. Each frame captures the Target Game Object from a different angle.
Y Offset	Y offset of Target Game Object during bake.
Zoom Factor	A zoom factor of 1 always ensures the Target Game Object to be within the captured frame. You can increase the zoom to better fit the Target Game Object within the captured frame.

## 5. Billboard Baker Pro Editor Window



Variable	Description
Dynamic Lighting	Enable dynamic lighting feature. You need this if you have moving lights or want to rotate the object.
Main Light Source	The main light source that is considered for baking.
Use Atlasing	Use Atlasing Feature combines multiple billboard materials into one single material with a single texture called atlas texture. This increases performance by reducing drawcalls.
Max Texture Size	Reference size for billboard texture and billboard atlas
Bake All Billboards	Executes baking of billboards. Target objects are captured from different angles.

## 7. Features

Billboard Baker Pro Bundle now contains all addons to ease the development process.

### 7.1. Atlasing Feature

When objects get rendered, every material raises an additional drawcall. Drawcalls are commands from the cpu to the gpu and are quite slow. To enhance performance it is best to keep material count as low as possible. Atlasing puts several materials into one by copying each texture together into a so called atlas texture.

To use this technique activate "Use Atlasing" within the Billboard Baker Pro editor window.

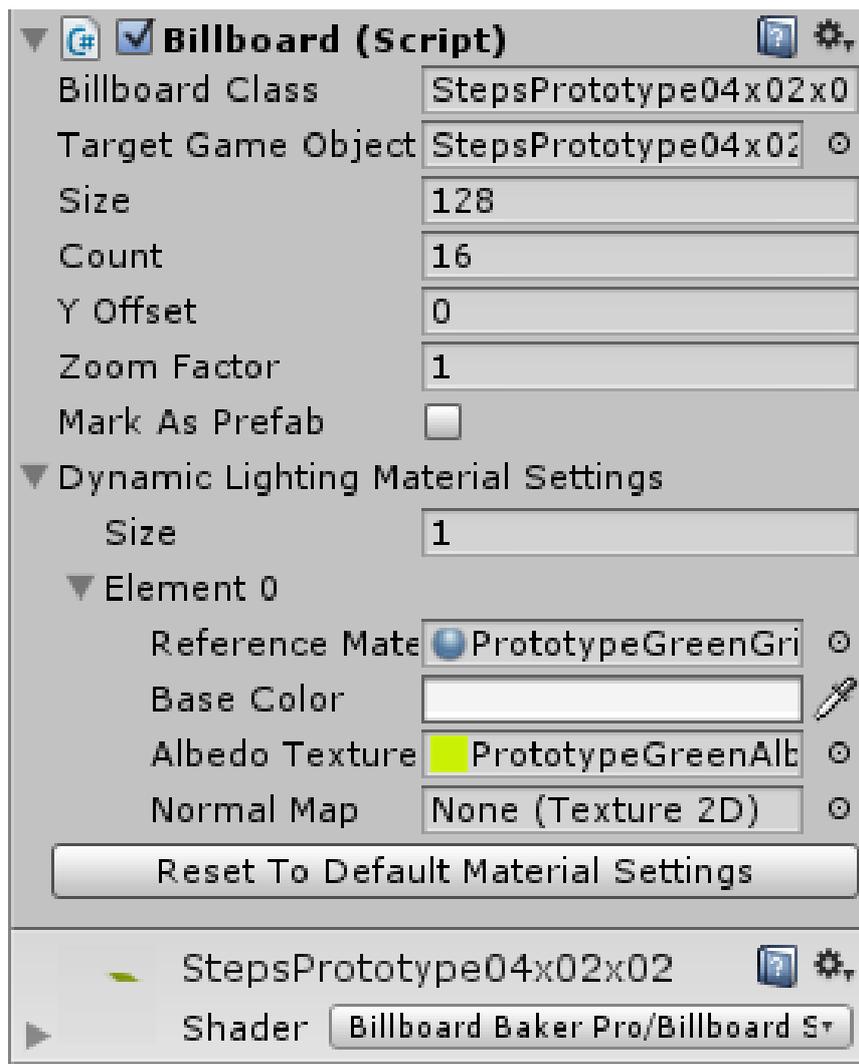


## 7.2. Dynamic Lighting Feature

As Billboard Baker Pro catches the render output of a scene, it was impossible to use dynamic changing lights. This addon takes another approach to create light responding billboards. Instead of capturing a rendered and lightened scene, baking with the dynamic lighting addon renders the albedo texture first and then renders the objects normals into a normal map. Lighting of the billboards is then calculated at runtime in the scene. It is possible to toggle this option on by activating “Use Dynamic Lighting”.



The relevant Billboard component settings for dynamic lighting can be found under “Dynamic Lighting Material Settings”.



<b>Variable</b>	<b>Description</b>
Size	Number of Dynamic Lighting Material Settings. Usually same count as materials in Target Game Object. This settings are used to create the billboard dynamic lighting matrial.
Reference Material	The reference material we create the settings for. By default settings are taken from this material.
Base Color	Base color of material.
Albedo Texture	Texture to use as albedo texture.
Normal Map	Normal Map of Object.
Reset To Default Material Settings	Creates Settings from reference material.

## 8. Current Features

<b>Feature</b>	<b>Description</b>	<b>Available</b>
Horizontal 360° View	Captured and displayed from all sides in horizontal layer.	Yes
Full 360° View		No
Dynamic Batching	Instead of drawing each mesh, meshes are batched to a single mesh within runtime. Increases performance by reducing batches.	Yes
Atlasing	Packing materials into less atlas materials. Reducing Drawcalls.	Yes
Dynamic Lighting	Enables changing light conditions and rotation of objects.	Yes
Dynamic Lighting Shader	What shaders are used to create dynamic lighting billboards. In future shaders for each unity shader may be possible leading to less difference between 3d object and billboard.	Blinn-Phong at the moment.
Mobile Ready	Shaders for mobile exist and will be handoptimized in future, as we now switched from Shaderforge to handwritten shaders.	No

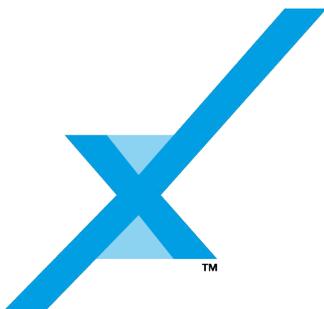
Feature	Description	Available
Instantiating Prefabs In Runtime	Now possible. Also duplicating billboards works now out of the box.	Yes
Crossfade Support	Integrated in billboard shaders to fade between mesh and billboard.	Yes
Rotating	Rotation around y-axis now possible in dynamic lighting mode. Makes no sense in static lighting mode, as lighting would need to be rebaked and you 'd need to create a seperate billboard class for each rotation you use.	Yes (Dynamic Lighting)
Scaling	Needs to be equal for all axes.	Yes (Uniform)
Moving in runtime	Although dynamic batching removes information about billboards object position, we no longer need to store this information within the uv's of the billboard mesh. It is now calculated from the normal direction and size.	Yes

## 9. Feedback

Please don't forget to give feedback and consider, that the project is still in continuous development.

You can write a mail to [mail@nexusgamesoft.com](mailto:mail@nexusgamesoft.com) or post your review in the Unity Asset Store.

Thank you in Advance!



Nexus Gamesoft