# Project UV
## from camera

v1.30

**Thank you for buying this product**.

This documentation cover the usage of the Project UV from camera Unity Component. If, despite this documentation, you encounter issues. Don't hesitate to contact me.

## Summary

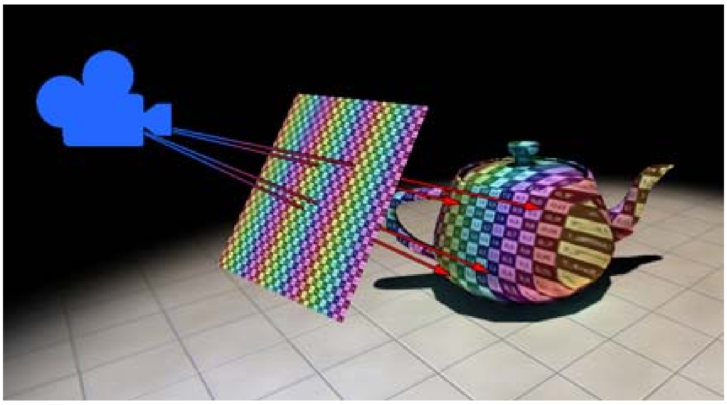
Legal stuff

I do not assume the consequences this component could have on you project nor the use you could do with it.

# Project what ?

## What are UVs ? (for those who don't know)

UVs are coordinates assigned to each vertex of a mesh that determine the position of a bitmap mapped on it.



**The purpose of this Unity package is to project UV from a camera (screen) to the mesh of your choice. This is useful to project a texture on a mesh dynamicaly (or not) during gameplay**. It's a sort of dynamic camera mapping.

## Features

- Project by one button the UVs from camera to the object(s)
- Realtime or manual projection
- Planar or spherical projection
- UV tiling
- Display watermark to preview projection over the object
- Keep the watermark aspect ratio or fill the entire screen
- Set up watermark opacity
- Project UV on single object or on sub-objects.
- Restore original UVs with a button or by deleting component.
- Instantiate sub-objects meshes.
- Optimize CPU usage (minimum updates mode, nth frame projection)
- Layer filtering
- Project UV1 or UV2 channel
- Oculus VR compatible

Unity 4 & 5 compatible

# Quick start

To quickly use the component :

1/ Select the object you want to project its UVs from camera.

2/ Assign the Project UV component (located in menu ***Component/Project UV/Project UV from Camera***).

3/ Click on the « **Project UVs** » button to project the UVs from the camera once, or check the checkbox « **Realtime projection** » to project UVs each frame.

4/ Enjoy.

# Planar and Spherical mapping

Project UV from Camera can project with two different methods : Planar & Spherical.

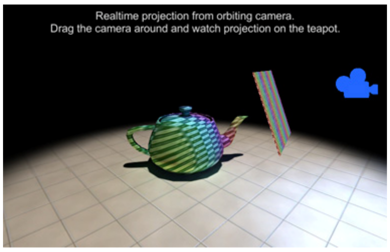
## Planar mapping

With this type of projection, the uvs are projected as flat from the camera screen space right in front of it. This kind of projection is fairly simple and do not involve very cpu intensive computations. You can use it in realtime quite easily. This depends on the amount of vertex you want to project on.

### Spherical mapping

Spherical mapping extends the projection beyond the camera screen space. It projects UVs all around the camera on a virtual sphere. This kind of projection is usefull to re-create a whole environment around the player for example. This projection is useable in realtime but it has a drawback : the realtime projection only move vertex uv coordinates. Since it's a sphere projection, the seams might not be accurates. So the mesh structure must be modified. This operation is very cpu intensive because it needs to parse every vertex of every triangle that compose the meshes. You can project precisly (not in realtime) in this way, by using the « precise spherical projection » checkbox or with the API.

# Demos scenes



Realtime projection from orbiting camera.
Drag the camera around and watch projection on the teapot.

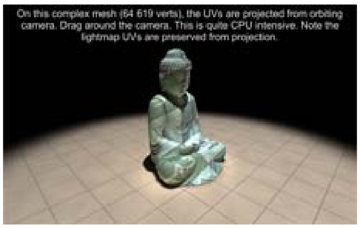
Demos scenes are located in the ***ProjectUV/demos*** folder. Those demos underline some features of Project UV.

### Realtime projection

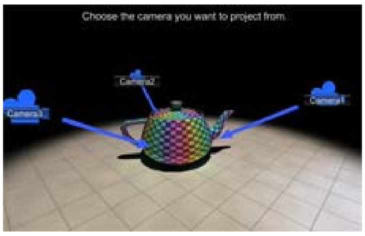
This simple scene show the realtime projection feature. A simple teapot (FBX object) has its UVs projected from a moving camera. A flat cube has been placed in front of the orbiting camera to simulate the projection. Drag the camera around to modify the projection.

### Complex gameObject

Is a scene that project UV over a quite heavy object (64 619 vertices) is used to project its UVs in realtime. The same orbiting camera system allow you to move the projection around the mesh. The performance depends on your configuration and resolution of game window. More complex meshes could be handled of course. It's up to you to test it.



On this complex mesh (64 619 verts), the UVs are projected from orbiting camera. Drag around the camera. This is quite CPU intensive. Note the lightmap UV's are preserved from projection.



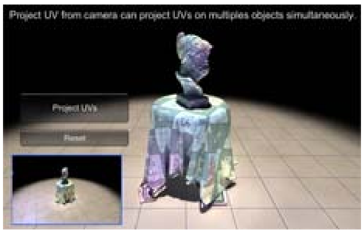
Choose the camera you want to project from.

### Simple API example

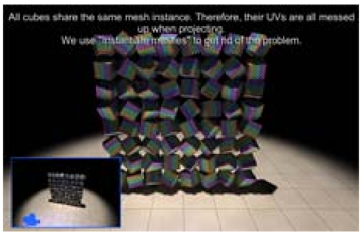
This scene shows up the very basic usage of the API.
Watch the script « CameraUI ». Three cameras are orbiting around the teapot. Drag them around and click one of them to project.

### Multiple gameObjects

This scene demonstrate how to attach the script to multiple gameObjects easily. The API is used to project manually (and reset) the UV set. Click on the buttons while in game to project manually. The meshes are about 46 000 verts).



Project UV from camera can project UVs on multiples objects simultaneously.



All cubes share the same mesh instance. Therefore, their UVs are all messed up when projecting.
We use "InstantiateMeshes" to get rid of the problem.

### Instantiate meshes

When using multiple gameObject that share the same mesh instance, modifying its UV end up to a mess.This scene use the same standard cube mesh instance. Instantiate mesh is used to obtain a proper projection all over.

### Spherical projection

This scene makes you evaluate the realtime spherical projection mapping. Note the ugly seams produced. This is due to the box mesh structure which is not made for spherical mapping. To fix this seams, try using precise spherical projection option, then click « Project UVs ».
Note that the floating sphere is here to proove the uv projection is not faked with some sort of GUI artifice.

# Properties

ProjectUV contains some properties to control its features :

## Mapping type

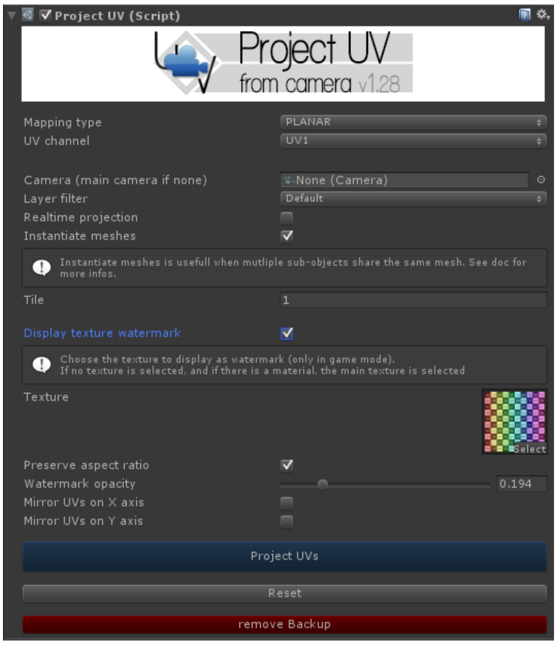
Choose the type of mapping you need.

## UV Channel

Choose the UV channel to project (by default UV1). Usually UV2 is used by Unity for lightmapping.

## Precise spherical projection (experimental)

This option allow you to correct mesh structure to create appropriate seams. This is used, in conjonction with the threshold value, to create a better quality spherical projection. With this option activated, realtime projection is NOT possible.
Backup need to be active for this option to work.
Considering the computing time and the fact that it is not usable in realtime, this option ought to be used in editor only for mapping setup.



☑ Project UV (Script)

Project UV from camera v1.28

Mapping type: PLANAR
UV channel: UV1

Camera (main camera if none): None (Camera)
Layer filter: Default
Realtime projection: □
Instantiate meshes: ☑

Instantiate meshes is usefull when mutliple sub-objects share the same mesh. See doc for more infos.

Tile: 1

Display texture watermark: ☑

Choose the texture to display as watermark (only in game mode).
If no texture is selected, and if there is a material, the main texture is selected

Texture

Preserve aspect ratio: ☑
Watermark opacity: 0.194
Mirror UVs on X axis: □
Mirror UVs on Y axis: □

## Precision threshold

This value set the precision limit for the precise spherical projection. A value of 0.25 is often a good choice. The more the value approach 0.0, the more vertex uvs are corrected. Use it carefully, a bad value might produce some nasty distortions.

## Oculus compatibility

Check this to activate the Oculus compatibility mode. This mode will replace the camera selection field by an Oculus prefab root controller. This mode also change the watermark display method to be visible for Oculus stereo view.

## Oculus prefab

If Oculus compatibility mode is checked, this field allows you to select an Oculus prefab root controller ("*OVRPlayerController*" or "*OVRCameraRig*"). The UV projection is done from right eye.

## Camera

Select the camera you want to project UVs from. If no camera is indicated, the main camera will be chosen. Of course, at least a camera need to be present for the component to operate.

## Layer filter

This feature allow you to filter the gameObject affected by the component. This has no effect if the gameObject doesn't have any sub-objects. This feature is usefull in case you have a complex sub-objects parenting system that you wish to filter the UV projection. Select the layer you wish to project UV. Some sub-objects must be part of this layer for their UV to be projected. Other sub-objects won't be affected.

## Realtime projection

When checked, UVs are projected each frame.
Note : When precise spherical projection is checked, no realtime projection is available. (see « Spherical and Planar mapping »)

### Minimum updates mode

This mode optimize cpu usage in realtime projection. Use it wisely (see Performance/Minimum updates mode)

### Nth frame projection

You can tune up the realtime projection frequency. For example, a value of 3 project UVs every 3 frames. This (virtually) increase by three the performances and let you use more cpu resources for others tasks. The default value is 1 (project on every frames).

### Instantiate meshes

When the component is attached to a gameObject or multiple sub-objects that share the same mesh instance, the projection of their UVs leads to a mess since the projection is done on a unique mesh. This property instantiates each mesh so the projection works correctly. Since this option increases slightly the memory footprint, don’t use this option if gameObjects don’t share any mesh instance (It would be useless).

### Tile

This controls the UV tiling. The value can be negative or positive. If you want to fill the entire screen with the texture, set the value to 1.0 (by default).

### Display texture watermark

In order to preview the UV projection accurately, this displays GUI texture over the entire screen corresponding to the projection.
When selected, you can choose the texture watermark to display. When the component is attached to a single gameObject with a texture, the mainTexture is picked up by default.
The GUI texture fill the entire screen disregard of its aspect ratio by default.
The Watermark is disabled when spherical projection mapping is selected. When Oculus compatibility mode is activated on Unity Free, the watermark display is disabled in Oculus view (since it need Unity pro feature to be displayed)

### Mirror on X axis & Mirror on Y axis

Let you mirror on both axis the UV layout.

### Preserve aspect ratio

By checking this option, you can keep the texture aspect ratio to the projection. This option is only available when a texture is selected (Difficult to keep a texture aspect ratio without a texture…).

### Watermark opacity

You can setup the watermark opacity.(0 = transparent, 1=opaque). The default value is 0.5.

### ProjectUVs (button)

When clicked, this project manualy the UVs over the object(s). This has no effect if realtime projection is on.

### Reset (button)

When clicked, this restore original UVs (even if Unity is closed meantime). This button will disable realtime projection.
*Note* : When the component is deleted the orignals UVs are restored so as the meshes instances.
*Note bis* : When the backup has been disabled (via the « Remove backup » button), this button disapear.

### remove backup (button)

When the component is added to a gameObject, datas are stored concerning all gameObjects involved (UVs and instances). This ensure to be able to get back to original state in case of unfortunate modification. Clicking on this button delete all backup datas.
Some developers like to keep a clean project and the amount of extra backup data may be useless from their point of view. Carefully use this functionnality, because after disabeling backup, all modifications applied to gameObjects will be undoable (at least not automatically in most case).

## Errors

In some case, the component display Errors messages. Fix the issue that prevents Project UV to operate (see FAQ), and click the « Recheck » button to re-activate the component.

# Oculus Rift 2

This compatibility mode is quite experimental since the Unity plugin for the oculus regularly evolves. By the way, the goal for this mode is to ease the use of Project UV with Oculus Rift. I've only tested on Oculus Rift DK2 0.4.4 (Firmware 2.12). I do not assume this works flawlessly on any other version.

## Watermark

For testing purpose mainly, the watermark on Oculus compatibility mode is "displayable" in game. In editor, since the watermark display uses Pro features, it is disabled on Unity free version.
For this moment, and for performance reasons, the projection is done from right eye, so you may encounter parallax issue if you try to project from quite a near distance.

# API

The component can be controled via script.
See the API doc (in **ProjectUV/Documentation/API.htm**) for more details.

# Performances

UV projection is only supported by CPU.
So, the performance rate depends directly on some points :

- **Your mesh and the platform you will run you project on.**
  If your mesh has a lot of vertices, this may be heavy to treat. Particulary on mobile platforms.
- **Realtime projection**
  Activate realtime projection is often not necessary when one projection is needed. The preview via the watermark is far less cpu expensive. Avoid realtime projection as much as possible if you are facing performances issues, especially on mobile platform.
- **Preserve aspect ratio**
  Preserving aspect ratio of watermark texture to the projected UVs is a bit more CPU intensive. Somes ratios are computed (watermark texture, screen size) and the texture is scaled and offseted in the proper way. This explain the cpu ressources needed.
- **The rest of your scene**
  If your scene include many meshes, lights, shaders, graphical effects, components that use CPU cycles etc..., you may ended with a lymphatic ProjectUV component… Manage wisely your resources.

**Special note for spherical projection :**
While using precise spherical projection, the projection might be computing for some seconds and hangs the editor. Don't panic, just be patient, projection will be done soon. Avoid at all cost using this option on heavy meshes.

## Minimum updates mode

This mode is only available when realtime projection is on. When on, the component tracks position and rotation of both camera and gameObject. If none of them move from one frame to another, the projection is delayed. This ensure to avoid dramatical useless CPU peaks.
Carefull : for performance reasons, sub-objects are not tracked. If sub-objects transforms are changing, the projection will not occur in minimum updates mode.

## Nth frame projection

This option is only available when realtime projection is on. It allows to tune up projection frequency. Increasing the value decrease the frequency of UV projection. With this, you can then priorize cpu resources for other tasks than UV projection if needed. This is usefull if you don't need to spend very much cpu resources on UV projection but need to project it regularly.

# FAQ / Troubleshooting

1- **I have attached the component to my gameObject but nothing happens.**
The component, by default, doesn't project UV at start. You need to activate either realtime projection, either click on the « Project UVs » button to proceed.

2- **The projection is all messed up. What's wrong ?**
Maybe you have objects that share the same mesh instance. Since there is indeed one only mesh, when modifying its UVs, all its instances get the same UV set. To get ride of this problem, just check « Instantiate meshes », this would fix the issue.

3- **I have an error wich states : *« The component needs at least a camera to operate. ».***
No camera is on the scene. Just add one and click « Recheck ». This will fix the problem.

4- **I have an error wich states : « Project UV cannot be assigned to Camera ».**
Quite explicit : You cannot attach the component to a Camera, just gameObjects. This component get meshes from MeshFilter of gameObject or sub-gameObjects. Camera doesn't have any.

5- **I have an error wich states : « Project UV cannot be assigned to Light».**
The component can only be assigned to gameObject.

6- **I have an error wich states : « Static object's UV cannot be edited. Please uncheck "Static" property of all objects and sub-objects to be able to use this component. ».**
Since UV are uneditable on static objects. You need to uncheck the Static property of all gameObject and sub-objects the component is attached to.

7- **« Instantiate meshes » doesn't appear anymore. Where is it ?**
Instantiate meshes option appear only when sub-objects are attached to the gameObject Project UV is on.

8- **Reset button has disapeared ?**
You have probably disabled backup previously. This delete all original UVs datas or instances. By the way, restoring to original state become impossible. This explain why the Reset button is no more visible. The only way to get it back is to remove the component and add it again.

9- **I need to control the component by script. How can I do ?**
To control the component by script, you need to read the API doc. This documenting the properties and methods to access the component.

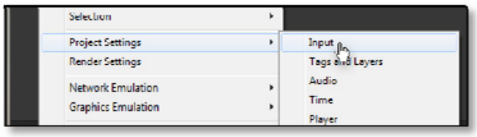
10- **The performances are poor. This component is a piece of #?!/@.**
The goal of this component is to project UVs from camera. Dispite the code optimisation, realtime projection is a feature reserved to « relatively » low vertices meshes. Use it very carefully. The UV projection is not supported by GPU, so the more your mesh is heavy, the less your CPU is powerful, the more ProjectUV will be jerky. Use as possible watermark to allow the gamer to preview the projection. Don't forget to use API methods to project it manually. This will be more efficient and will less sollicitate CPU ressources. See Performances topic.

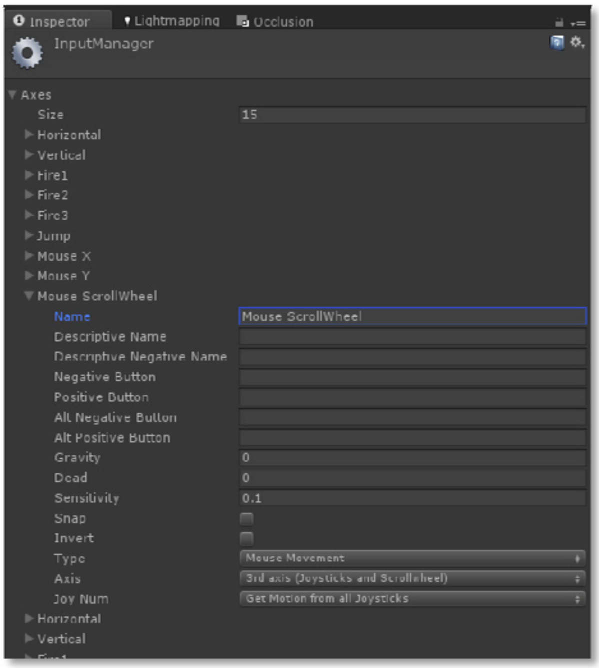
11- **What about my lightmap ? Will it be affected ?**
Absolutly not (if you choose to project UV1 and not UV2). The lightmap channel is not affected by projection. You can safety use it with your lightmap. All demo scenes use baked lightmap wich is not affected. To bake lightmap Unity have to use static gameObjects. But ProjectUV need to have dynamic gameObjects. Just bake your lightmaps with static property on, then, when it's done, change back and uncheck « Static » checkbox to be able to use it with ProjectUV.

12- **I use the RestoreUVs() method API but nothing is restored. Why ?**
You may have disabeled Backup by clicking on the « remove Backup » red button. Thus, the original UV datas/instances are lost since you have applied new UV projections directly to the meshes instances involved.

**13- When I start the scene in game mode, I get the following error : « UnityException : Input Axis Mouse ScrollWheel is not setup. To change the input settings use : Edit -> Project Settings -> Input. »**

This error occurs when an axis is not setup correctly. So Unity doesn't recognize mouse scroll wheel. To fix this issue, go to *Edit > Project Settings > Input* and add a new input element called « *Mouse ScrollWheel* » with the 3rd axis setup has the screenshot below.





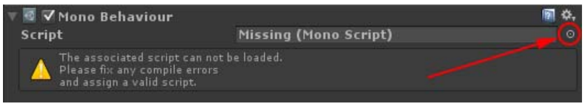
Axes
Size 15
Horizontal
Vertical
Fire1
Fire2
Fire3
Jump
Mouse X
Mouse Y
Mouse ScrollWheel
Name Mouse ScrollWheel
Descriptive Name
Descriptive Negative Name
Negative Button
Positive Button
Alt Negative Button
Alt Positive Button
Gravity 0
Dead 0
Sensitivity 0.1
Snap □
Invert □
Type Mouse Movement
Axis 3rd axis (Joysticks and Scrollwheel)
Joy Num Get Motion from all Joysticks
Horizontal
Vertical
Fire1

**14- I get a Missing (Mono Script) in place of my Project UV component. Heeeeeelp !**

This is due to dll assembly since the component's meta guid in Unity is not the same from the one you have inserted before. This is a Unity issue.

To fix this, three different solutions :

- Either delete the component (by clicking the gear icon on the right) and add it again.
- Either Change it manually by clicking on the little circle-dot on the right side of the « Missing (Mono Script) » input and select « Project UV ».



Script Missing (Mono Script)

The associated script can not be loaded.
Please fix any compile errors
and assign a valid script.

- Either Use the excellent script of daikonforge.com @ http://www.daikonforge.com/dfgui/missing-script/
  This script scan for missing scripts (not only Project UV) and offer to fix the missing ones.

**15- Why dll ? Why don't you release source code ?**

This will be done in a future release for sure. Dll assembly is an elegant way of making asset from my point of view. When this asset will be fully debugged and featured, I will deliver the source code. If you have a question, or a request to integrate some new features. Do not hesitate to contact me.

**16- The projection is not accurate at all… maybe even false ! This component is a crap !**

Not exactly. If your mesh doesn't have enough vertices in the field of view. Its projected UVs might not be accurate. Try to either distance the camera, either increase the number of vertices. Without enough vertices to define the UVs, the mapping would appear to be very distorted. The more your have vertices, the more precise, the mapping will be. Remember that you need enough vertices in the field of view for the component to be able to project it accurately. For big objects, or if you approach very near, the number of vertices might become too weak to maintain a good projection.

**17- I can't see my texture on a big part of my object. Why ?**

This is probably due to the TextureWrapMode used. Use Repeat instead of Clamp.

**18- I'm using spherical projection and on some edges, the projection is all distorted. What happens ?**

The realtime projection for spherical mapping is not perfect. It needs to alter mesh structure which is not possible in realtime. Try using precise spherical projection option to project a little more accurately. If you already use it, try changing the threshold value. This might solve the problem.

19- **When I click « Project UV » on the editor, Unity hangs and nothing happens.**
You probably use the precise spherical projection. This option alter the mesh structure (uv, vertices, normals, triangles), and this operation is very cpu intensive if your mesh is very heavy. This computation might last for seconds. Be patient and try avoiding using very heavy meshes.

20- **I've discarded the backup but I want to retrieve the original mesh datas. Help !**
When backup datas have been deleted, the only way to restore your mesh at its original import state, is to reimport it. Right click on the mesh prefab in your assets, and select « Reimport ».

21- **What the point of nth frame projection ?! If I increase this value, the projection seem more and more laggy !**
Yes, if you increase the value, the projection is less frequent (a value of 2 project UVs every 2 frames instead of every frames). This is indeed more laggy, but this frees more cpu resources for other tasks. This option is usefull if the projection do not need to be too frequent or too precise to optimize cpu usage (when the camera or the gameObject do not move too much for example).

22- **Why is the watermark isn't available ? I don't see the checkbox to display it !**
Well, you might be in Oculus mode and running Unity free. Since the watermark display process isn't the same for Oculus display mode at all, it needs Unity pro feature… That's why (when Oculus compatibility mode is on), the texture watermark display is only available for Unity pro users.

23- **Is it possible to project simultaneously both UV1 and UV2 channel of a same gameObject ?**
Of course it is. You just need to create two separate Project UV instances on the gameObject. One will be for UV1, the other for UV2. You can then, for example, bind two different cameras on the projections. They are both independent from each other.

24- **The demo scenes doesn't have any lightmaps ?** You probably use Unity 5. Since Beast isn"t supported anymore, the lightmap have to be recomputed with the new enlighten engine. In order to ensure backward compatibility with Unity 4.5+, lightmap cannot be computed with enlighten yet.

25- **Why is the UV projection isn't working on SpriteRender ?** Unfortunately, Sprite UVs are read only. Thus, they cannot get re-projected or customized. I suggest to use Quad as sprite instead to benefit Project UV projection capabilities.

# Contact

In order to continuously improve this product, feel free to contact me if you experience any issue, or want to suggest any improvements, addons, or new packages.

**info@graphiste3d.net**

# Release notes

## V1.30

- Performances improvements
- Nth frame projection optimization method
- Fixed some issues
- Unity 5 compatibility check
- Oculus compatible
- Documentation update

## V1.28

- Added the possibility to project UV2
- Documentation update

## V1.27

- Fixed texture wrapMode issue
- Documentation update

## V1.26

- Fixed Windows Asset Store export issue
- Documentation update

## V1.25

- Downgraded the package to Unity 4.3 compatibility
- Added Mirror on X & Y axis feature.
- Documentation update

## v1.24

- Optimizing profiler drawcalls during gameplay
- Documentation update

## v1.23

- Added Layer filter feature
- Added Minimum updates mode
- Documentation update

## V1.22

- Fixed initialisation issue after instantiation.

## V1.2

- Documentation update
- Added spherical projection
- Added precise spherical projection option
- Fixed some issues.

## V1.01

- Documentation update
- Preventing some Input errors in MouseOrbitAdvanced script
- Fixing some dll compiling issues.

## V1.0

- Manual and realtime projection
- Backup datas
- UV projection tiling
- Watermark preview
- Watermark opacity
- Respect watermark aspect ratio or not
- Restore datas
- API support

- Instantiate meshes
- Demo meshes credit : http://archive3d.net/