



# PIDI

## GAME DEVELOPMENT FRAMEWORK™

BY IRREVERENT SOFTWARE™



**PIDI**  
**GAME DEVELOPMENT FRAMEWORK** <sup>TM</sup>  
BY IRREVERENT SOFTWARE <sup>TM</sup>



XFur Studio

**USER MANUAL**

# Index

## Introduction

- 1. XFur Studio. Introduction to Fur Shaders.....1
- 2. XFur Studio. New LOD System.....4
- 3. XFur Painter. Basic Guide.....5
- 4. Final notes and Performance Tips.....10

## **PIDI – XFur Studio Introduction**

PIDI- XFur Studio is a complete system of tools and shaders designed to add fur easily to any model in Unity. The shaders included with our package are packed with many features not seen in most other fur solutions, such as full PBR integration, full lights and shadows support or full deferred and post process effects support.

This package also includes XFur Painter, a tool that allows you to add fur and precisely control all its settings by painting directly over any 3D model inside the Unity Editor to then export the resulting fur maps in a format compatible with the shaders. No need for any external tool nor any complex edition process.

This shaders collection has been developed to provide the highest quality possible with a good performance. The high fidelity of the rendering techniques used as well as their complexity make it a AAA-ready solution for high end games.

This shader has not been designed nor is intended to be used with low end PCs nor mobile devices. While many optimizations have been made and a lot of work has been done to reduce its memory footprint and draw call requirements it is still best suited for newer and dedicated graphics cards. Using a dedicated graphics card you can render dozens of animated models using this shader without a significant performance loss.

This manual will explain the basics of the shaders and how to set it up to be compatible with either third party models (such as those bought in the Asset Store) or with your custom-made 3D assets.

It will also teach you the best practices on lighting and rendering of fur-enabled meshes to get the maximum performance from every scene, as well as how to use the XFur Painter tool with ease to quickly get fur maps ready to use.

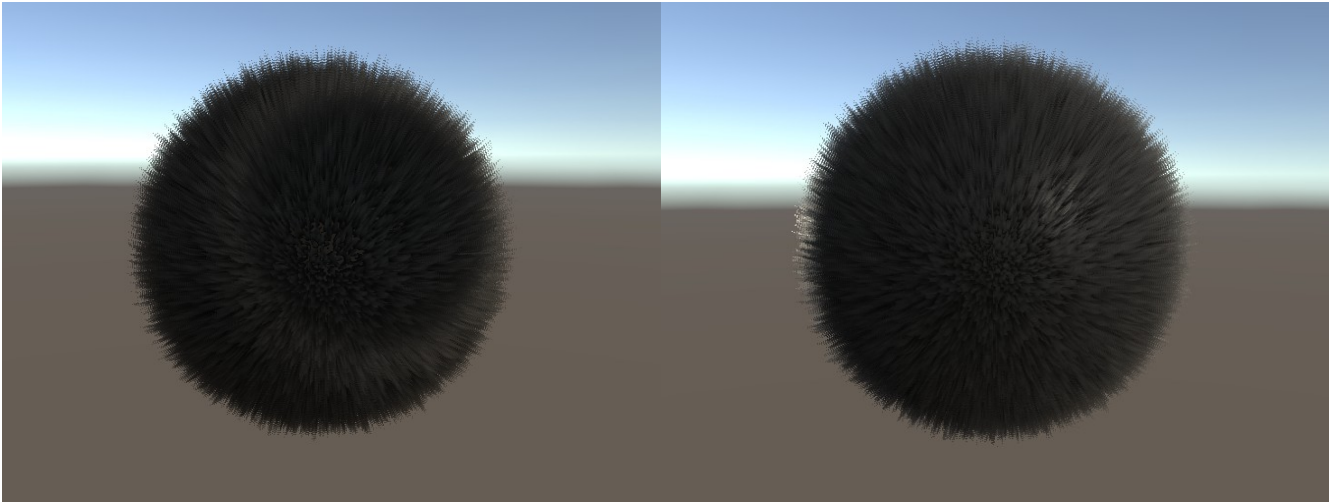
Thank you for purchasing our tool, we really hope that it will help you to keep making amazing games!

Irreverent Software

## 1. XFur Studio. Introduction to Fur Shaders.

Starting with XFur Studio version 1.1 there are 2 main shader branches split into 4 variants at different amounts of samples for a total of 68 shaders.

The main two branches are **Forward Anisotropic** and **Standard Specular**. These branches' main difference is the lighting model used on each of them. The first one uses a custom lighting model with realistic anisotropic highlights based on the Kajiya-Kay model designed to work in Forward mode only. The other one is a more compatible version based on standard specular highlights that can be rendered in both forward and deferred setups.



*Fig. 6 – Forward Anisotropic (Left) vs Standard Specular (right) Lighting models*

Each of this branches is split into variants that support different features targeted for different purposes and with different performance considerations. The variants are :

- **Shadows** : Heavy and complex shaders with full support for shadows, depth based effects, post process effects, occlusion, probes, GI and more. Designed for high-end devices and better suited for a deferred based workflow with temporal anti-aliasing enabled for higher quality results.
- **No Shadows** : Forward oriented version of the shaders above. When used in forward rendering these shaders will ignore shadows and most depth based effects.
- **Simple** : These shaders come with and without basic shadows support but are limited to partial light support (just 1 per-pixel directional light, everything else is calculated per-vertex) and forcefully render in forward mode. They do not support depth based effects.
- **Painter** : Shaders designed to work with XFur painter, with some features exclusive to this software.

Furthermore, all variants come with versions that do not require any especial UV mapping for the fur to work, marked with (Triplanar) on their name. These shaders use a special UV-free projection that displays fur in a uniform way over the surface of the mesh.

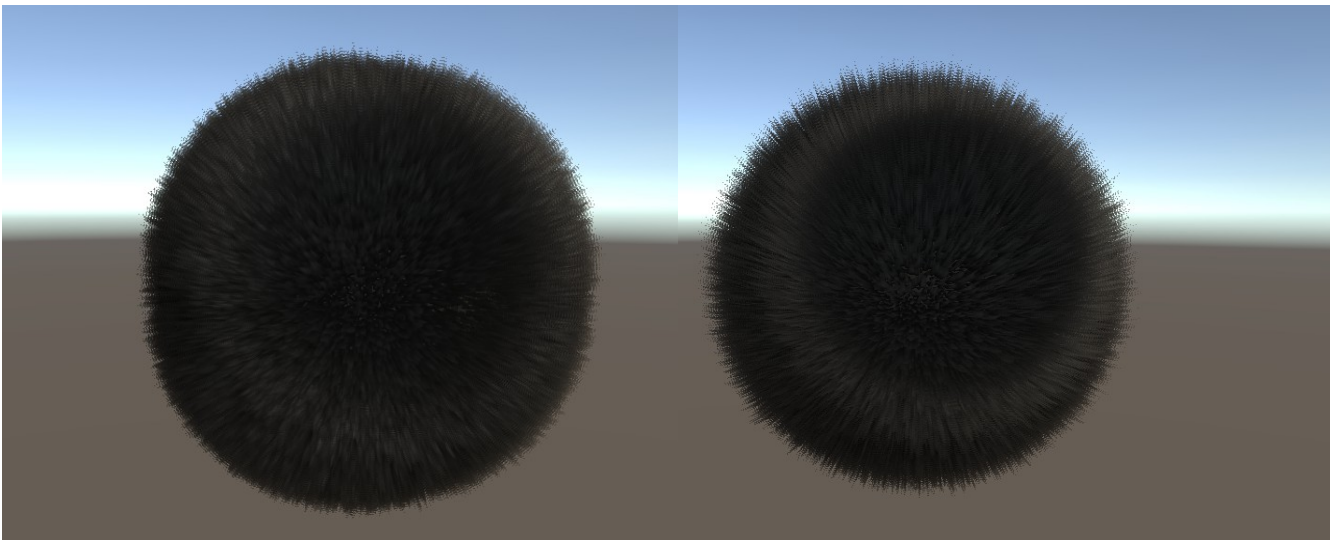
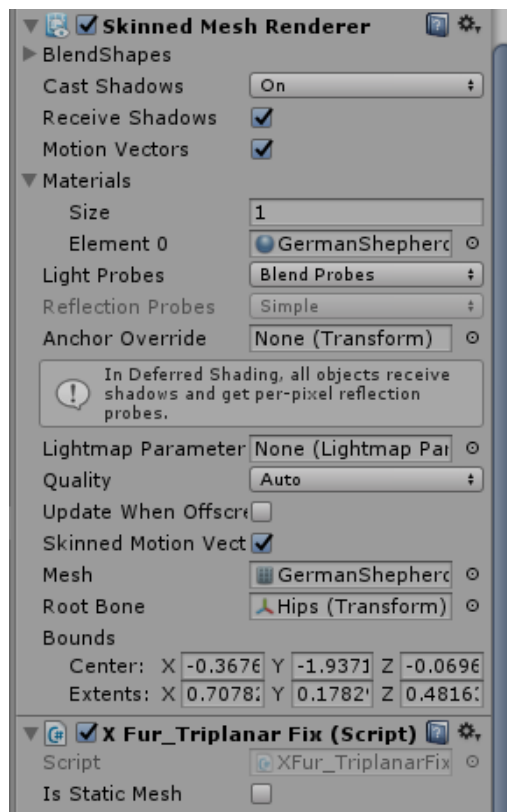


Fig 7. UV-Free Triplanar shader (Left) vs default shader with artist-made UV Map (Right)

For most third party models where there is no secondary UV map designed for Fur or where the mapping does not suit the shader as expected, the new UV-Free triplanar model is heavily recommended.

XFur Triplanar shaders are not standard triplanar shaders. They require some information to be stored in the vertex colors of your mesh so they can work with Skinned meshes without graphical glitches. To use a XFur Triplanar Shader, first drag the Xfur\_TriplanarFix into the object that has the Skinned Mesh Renderer. This will patch your mesh and add the required color information automatically. **Be warned that this will overwrite the original mesh and any color information it contained.**



There is another consideration to make when working with XFur shaders and that is the sample count and the rendering setup being used. XFur shaders look different when switching from deferred (coarser) to forward (softer) and from gamma to linear color modes. If your game needs to switch between these modes at runtime it is recommended to have different materials configured by hand to get the best look on each mode.

Shaders come in 4, 8, 16 and 24 samples versions. However, lower sample versions produce different results due to them being lower quality. It is important to consider this when working with the dynamic LOD systems included in this tool.

## 2. XFur Studio. New LOD System

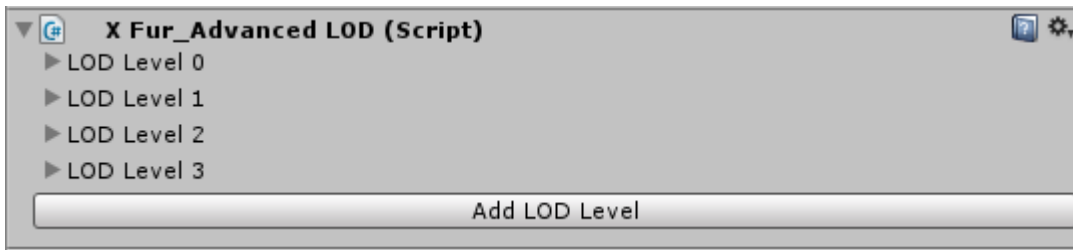


Fig. 8 – The new XFur Advanced LOD component

Starting with version 1.1, XFur Studio comes with a highly improved LOD manager. While it requires more setup than the previous version it also allows far greater customization, a bigger performance boost, better control over the results while also working in the scene view, helping you to edit more easily fur-heavy scenes.

Using it is quite simple. When you add the Xfur\_AdvancedLOD script to an object it will automatically create 4 LOD levels with configured for different distances. Each LOD level has a list with all the materials in the object's renderer. By switching the different materials on each LOD level you can decide which material is used at which level, switching between higher sample counts and full shadows support on close ups to lower quality versions or no shadows support when far away.

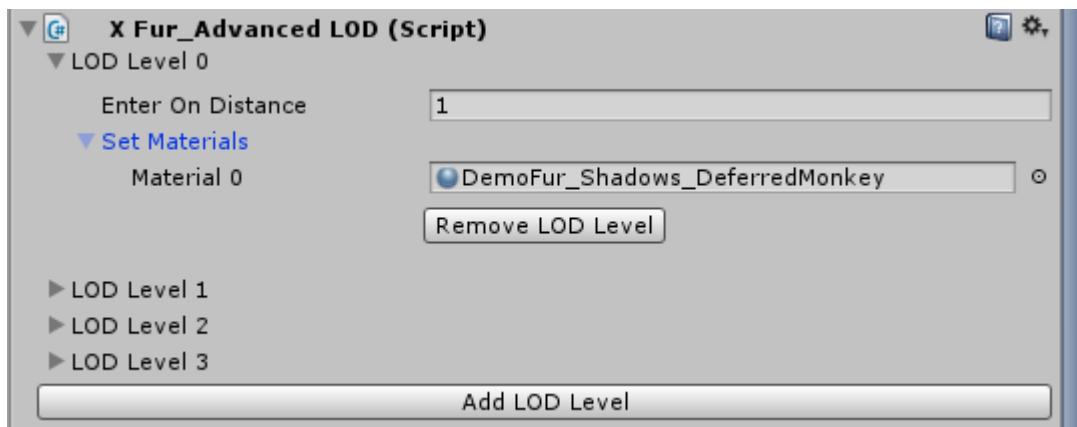


Fig. 9 – How to set up the LOD Levels in the new XFur Advanced LOD component

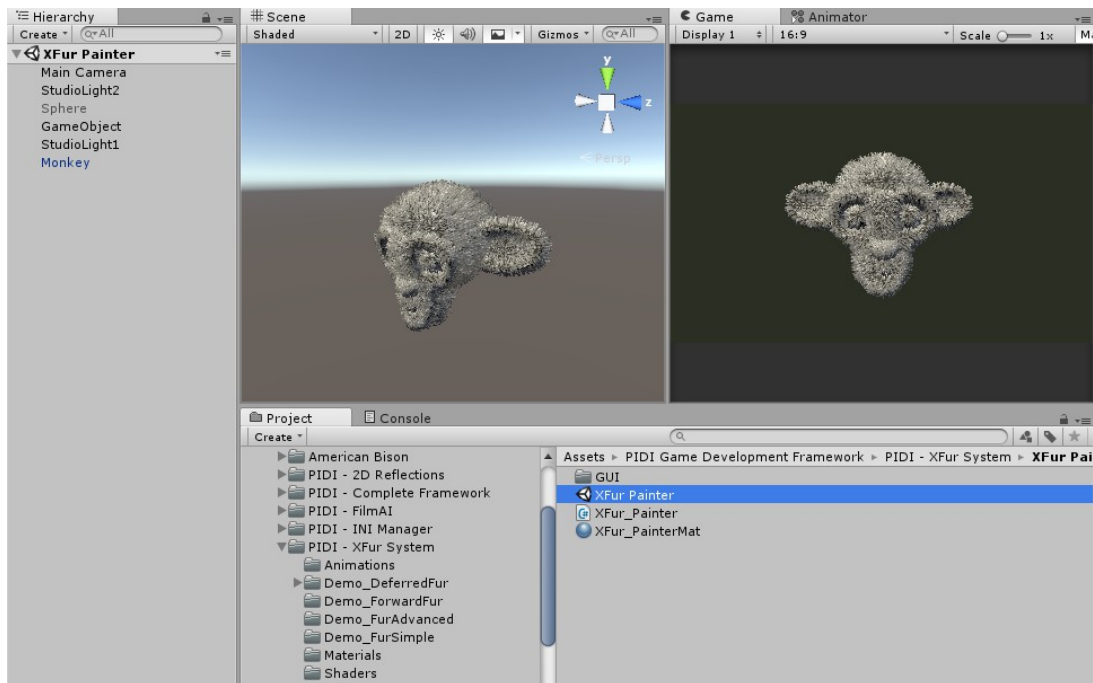
The component switches to a LOD level when the distance between the current camera and the object to render. Be careful with this, as it means that LOD may have unexpected behavior if several cameras are looking at it at once and all of them are displaying on the screen at once.



### 3. XFur Painter. Basic Guide



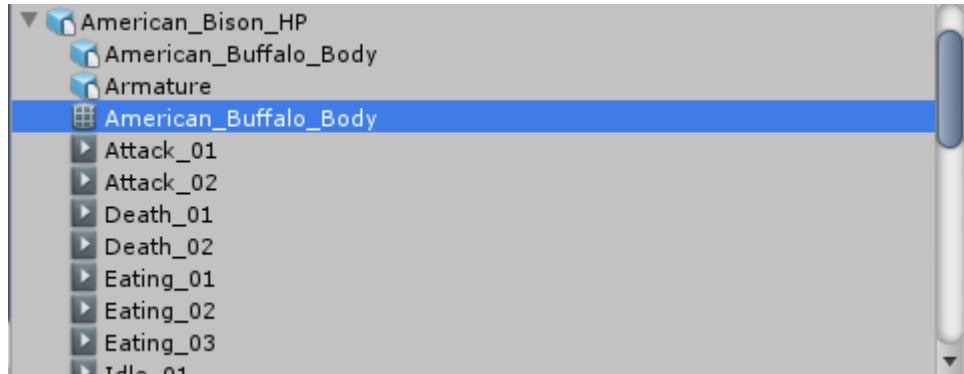
Fig. 10 – XFur Painter main screen



XFur Painter is an easy to use tool that allows you to create a Fur Data Map by directly painting on top of any 3D model. Using it is fairly easy as it involves just a couple of preliminary steps and hitting the play button on the Unity Editor.

Fig. 11 – XFur Painter main scene

To start, open the XFur Painter scene included in the package. Delete the Monkey object already in the scene and replace it with the model you want to paint. For this example, we will use another asset, an American Bison model.



*Fig. 12 – Identifying the mesh inside a rigged and animated 3d model*

This model is rigged, so we will go through its hierarchy in the Project view and drag the mesh object onto the scene. Some animated models (especially those that come from software such as Blender) may present an incorrect rotation after being dragged onto the scene. Set the mesh in the correct rotation and place it at the position 0, 0, 0.



*Fig. 13 - The model placed at 0.0.0 in the XFur Painter scene*

Once the model is placed and ready, add a Mesh Collider component to it. This is necessary for the XFur Painter application to detect it and to allow you to paint over it.

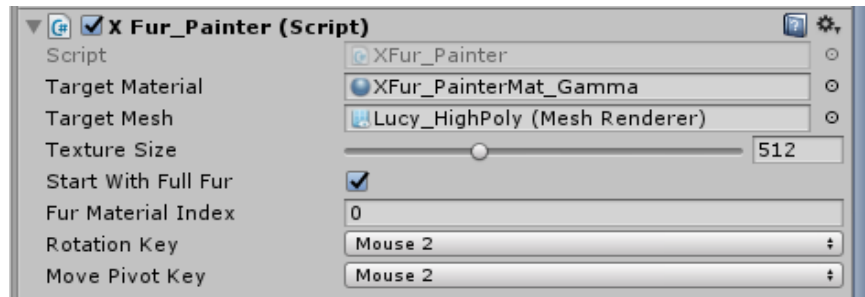


Fig. 14– The XFur Painter component configured in the scene.

Once you have finished doing this, you just need to assign the mesh in the scene (in this case the American Buffalo model ) to the **Target Mesh** variable and decide if you want to start with the mesh fully covered in fur and shave it as needed or with a shaved mesh and paint in the fur as needed.

For most 3<sup>rd</sup> party models it is recommended to start with a clean mesh and paint the fur in, as this will allow you to avoid leaving difficult to reach places (such as the inside of the mouth or the space between fingers) covered in long fur.

The other parameter you can control is the size of the texture that XFur Painter will work with. Currently, a size of up to 1024x1024 is supported. Arbitrary non power of 2 sizes are also supported, but it is recommended to use values of either 256x256, 512x512 or 1024x1024 for your textures. Start with full fur decides whether the mesh will start covered in fur or completely clean. The Fur material index is used for models with more than 1 material to specify which material is the Fur enabled one. The last two variables allow you to configure the inputs to rotate and move while in XFur Painter

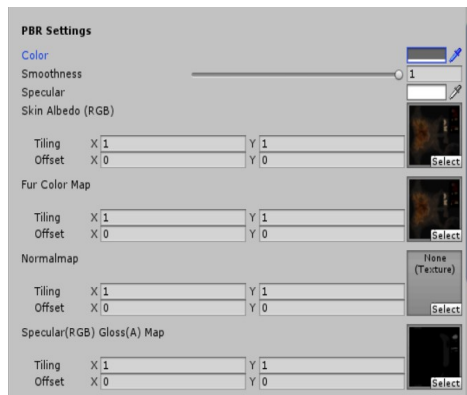


Fig. 15 – The Xfur\_PainterMat set up to use with this mesh

After this step, you can just hit play and the XFur Painter tool will start working immediately but for even better results and an easier to manage workflow it is recommended to also prepare the XFur\_PainterMat material with the textures that you will need to use for the model. In the image above we have assigned the diffuse map, specular map and the fur color map to the XFur\_PainterMat material. If you assign a fur data map or a fur direction map to the material, these will be copied and used as a starting point for the painting process. This allows you to, for example, continue a previous painting session by assigning previously exported textures.

You should also set the Fur Direction vector to a value higher to 0 so that the fur grooming done inside XFur Painter is visible. For best results while working with 3<sup>rd</sup> party models or models that don't have a secondary UV map designed explicitly for fur mapping, we recommend using a Triplanar XFur shader. Once all of this is set up we can hit play and start painting the fur with ease.



Fig. 16– XFur Painter in action, ready to start painting the American Bison model

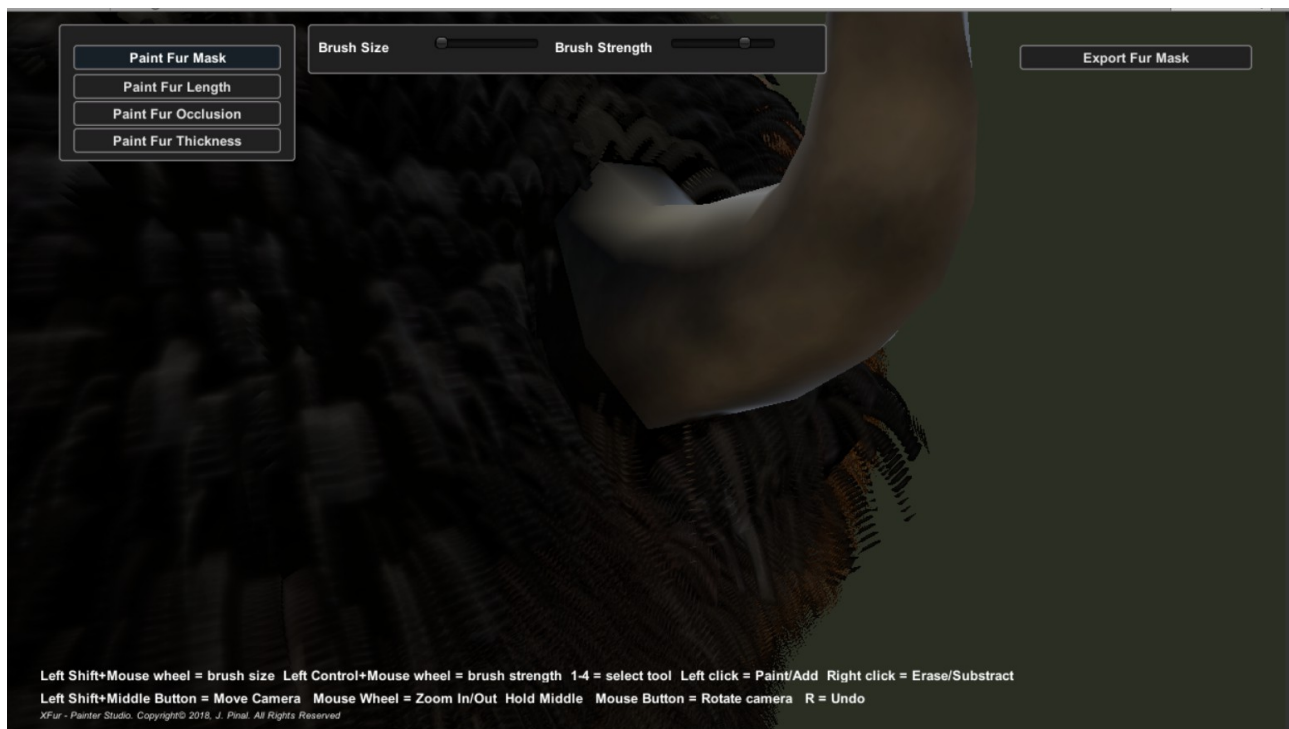


Fig. 17 – Shaving the fur from the model's horns

In this example we have decided to start with the mesh completely covered in fur since most of it will indeed need fur. We will just remove it from unwanted places such as the horns, eyes and hooves.

While in Paint Fur Mask mode (option selected in blue on the upper left corner or by pressing the **1** key) we paint over the horns with the right mouse button to remove the fur strands. The Fur Data maps are rendered at a 512x512 resolution to keep a good performance and also to be able to send the data to the vertex stage of the shader without issues. To paint finer details such as the eyes and horns we need to reduce the brush size to the minimum.

Once you finish painting your Fur Data Maps, click on the Export Textures button and save the resulting files anywhere on your disk.

You have now authored your first Fur Data Maps ready to be used with the XFur shaders on any material and on the 3D model, even when it is animated.

If you need a more in-depth tutorial about how to paint with XFur Painter there are two small video tutorials included with the tool in zip files that shows how to setup a mesh, paint it and export the resulting maps.

If you wish to author your own Fur Data Maps without the tool (mostly to get a higher resolution result or even more precision) the channel configuration is the following for the main fur data map : Red channel = Fur mask, Green channel = Fur length, Blue channel = Fur shadowing/occlusion, Alpha channel = Fur thickness.

For the fur grooming map, any value above 0.5 adds positive displacement of the fur on the X, Y or Z axes (red, green and blue channels) while any value below 0.5 adds negative displacement.



## 4. Final notes and Performance Tips

XFur is a highly complex system for fur in Unity and as such it has not been designed to be used on low end PCs nor on mobile. To use it at its fullest a dedicated graphics card is heavily recommended.

The system has been tested on integrated IntelHD graphics cards and on GeForce GTX 6xx level cards. Dedicated graphics card had great results and high FPS counts even with 100+ animated characters while integrated graphics cards had great results with up to 5 characters in scenes with a moderately complex lighting setup.

However, there are limits on the system. To get the maximum performance out of your scenes and characters we recommend you follow the tips below :

- Use deferred rendering whenever possible. Deferred rendering greatly reduces the amount of drawcalls done by the shader. A 16 samples shader will add just 1 drawcall for each additional point light after the first one is rendered in deferred mode.
- Do not light the fur enabled mesh with more than a few point lights plus a directional light in forward mode. If several lights are needed set the pixel light count to a low number.
- Do not affect a fur enabled model with many shadow casting lights. Each shadow casting light will re-draw the whole fur effect so a 16 samples shader affected by 2 shadow casting lights will make up to 54 draw calls even in deferred mode.
- Use the XFur\_AdvancedLOD script whenever possible. This script will dynamically adjust the number of samples based on the distance to the camera for maximum performance.
- Keep the fur enabled objects small if you are rendering in forward mode. Forward mode uses full transparency and an object that covers most of the screen with fur will cause a high level of overdraw and reduce the performance of the application.

The XFur\_LOD component (**Now Obsolete, use XFur\_AdvancedLOD instead**) is designed to help with managing the performance of the XFur shaders when added to a fur enabled object in the scene. Setting up this component is quite easy.

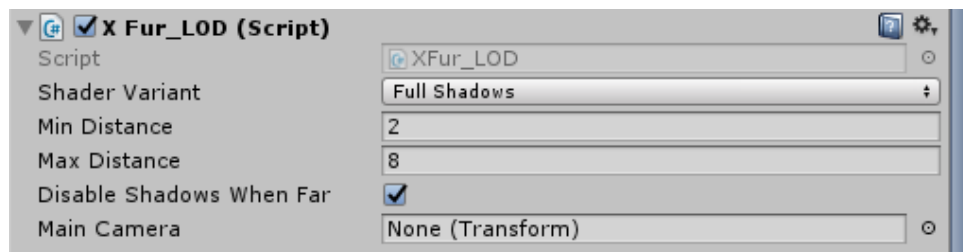


Fig. 18– XFur LOD component interface.

**Shader variant** controls what kind of fur shaders will be used in the 3D model. When the object is closer than **Min. Distance** to the Main Camera, the material will use the 16 samples variant of the shaders. When the object is between the **Min. Distance** and **Max. Distance** values the 8 samples version will be used instead. When the object is further away than the **Max. Distance** value the 4 samples variant will be used and, if **Disable Shadows When Far** is enabled, shadows will be disabled for this object as well.

If no Main Camera transform is assigned to the component it will automatically detect and assign the Scene's main camera.

These tips are just a general advice and can be ignored. In more modern PCs the performance cost of all these situations is not a heavy issue.

Despite all this we are always looking for new and better ways of improving XFur's performance and future updates will include shader collections geared towards more modest hardware as well as to improve the performance cost of the current shaders.

If you have any troubles with our software or need any assistance, don't hesitate in contacting us at [support@irreverent-software.com](mailto:support@irreverent-software.com) and we will get back at you with help.

Thank you very much for purchasing our software, we hope our tools and the PIDI Game Development Framework will help you make awesome games!

The Irreverent Software Team.

*Copyright© 2015-2018, Jorge Pinal Negrete. All Rights Reserved.*

*The monkey head 3D model "Lucy" included with this package was designed and modelled by Jorge Pinal Negrete and can be used freely in any demo, application or video as long as proper credit is given to Jorge Pinal Negrete and the following links are provided : <https://assetstore.unity.com/publishers/14062> & <https://www.irreverent-software.com>*