

# The Toby Foliage Engine

Version 1.1.0



## Documentation

**Toby Fredson**

[florianalexandru05@gmail.com](mailto:florianalexandru05@gmail.com)

# Table of Contents

---

<b>Table of Contents</b>	2
<b>Product Overview</b>	3
Compatibility	3
<b>(TTFE) Global Controller</b>	4
Global Wind Parameters	4
Known Bugs	6
Known Issues	6
<b>Shaders Overview</b>	7
How it Works Behind the Hood	8
Limitations	8
Vertex Colors	9
<b>Shader Settings</b>	10
Texture Packing	10
Texture Settings	11
TTFE Translucency	12
Limitations	12
Known Bugs	12
High Definition Render Pipeline (HDRP)	13
<b>Lighting/Shading Settings</b>	14
<b>Season Settings (+Snow)</b>	15
<b>Wind Settings</b>	18
Global Wind Settings	19
Trunk and Branch	19
Wind Mask	21
<b>Other Shaders</b>	22
<b>Optimization</b>	25
GPU Instancing	25
<b>Amplify Shader Functions</b>	26
<b>Original Assets</b>	27
<b>Credits and Acknowledgement</b>	28

# Product Overview

Introducing the Toby Foliage Engine (TTFE), the innovative single-pivot vegetation system that attempts to simulate multiple pivots per mesh, similar to Pivot Painter. These Amplify shaders provide a comprehensive solution for achieving realistic wind and plant shading. Its seamless integration and user-friendly interface make applying it to any mesh easy. There is **no need to worry about conversion** or any cumbersome steps as the shaders streamline this process.

TTFE is an automatic feature that works in conjunction with PBR shading. The wind physics is affected by internal mesh-based vertex position volumes. You can control the size of your volume to mask out wind in your material settings. Similarly, you can modify the behavior of your wind through the material settings.

**Made entirely in the Amplify Shader Editor**, these shaders offer fully customizable features and well-organized nodes that are easy to learn from.



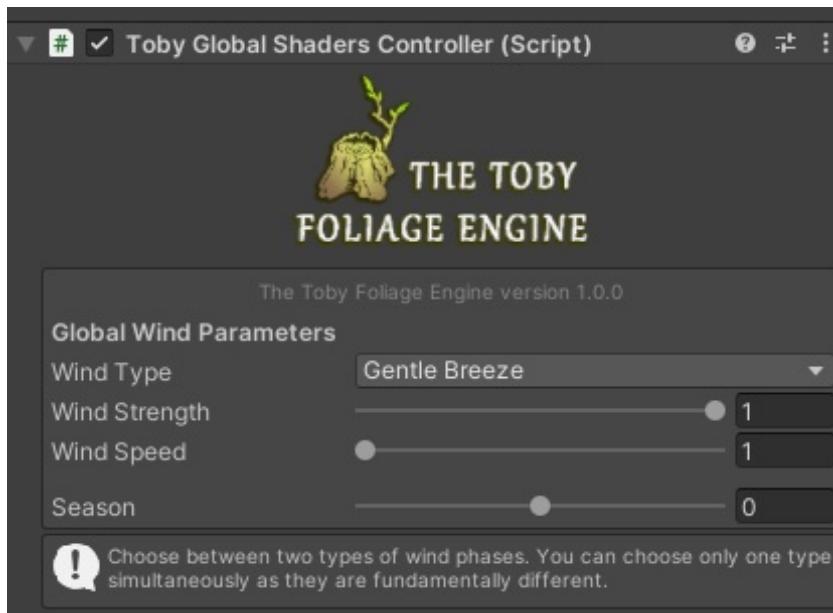
## Compatibility

- ✓ TTFE supports various third-party terrain and vegetation tools like [Gaia](#) and [Nature Renderer](#). (Tested with **Nature Renderer**)
- ✓ Supports motion vectors. ( It might be possible that motion vectors aren't fully supported in some HDRP builds).
- ✗ TTFE is not a perfect alternative and may not produce identical results as Pivot Painter. The wind only relies on a single point of rotation!
- ✗ Some features might differ slightly depending on the render pipeline used!

# (TTFE) Global Controller

This gizmo changes material settings for all shaders simultaneously. You can drag and drop the “(TTFE) GLOBAL CONTROLLER” prefab in the scene to do so.

You can find the prefab located in: `Assets\Toby Fredson\The Toby Foliage Engine\TTFE_Core\Resources\TTFE GLOBAL CONTROLLER`



## Global Wind Parameters

- **Wind Type**

Choose between two types of wind phases. You can choose only one type simultaneously as they are fundamentally different.

- **Gentle Wind**

Generate a gentle and realistic wind effect based on pivot rotation. This effect mimics the subtle rotations of plants growing in clusters, such as wheat fields. It will affect grass and trees but does not include directional trunk bending in trees. However, the branches will still be affected by the wind.

\*(Most performance friendly) [-6 FPS, 1.5 ms](#)

- **Strong Wind**

Intense wind pulse is driven by a 2D texture noise (3 vector RGB). Affects grass and trees.

\*(Most performance intensive) [-10 FPS, 2.5 ms](#)

- **Wind Off**

Turns off the wind completely via a static switch.

\*(Saves performance) [-0 FPS](#)

- **Wind Strength**

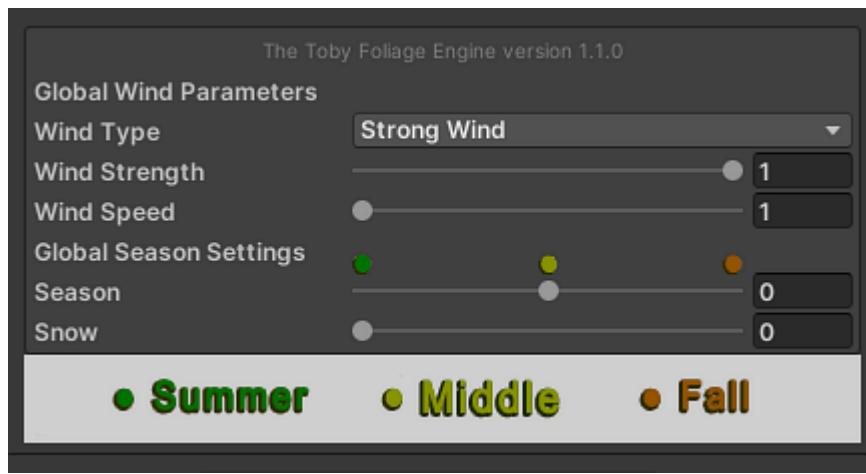
Wind intensity can be controlled and set to 0, but it's not the same as "Wind Off" and may affect performance.

- **Wind Speed**

Controls the wind speed.

- **Seasons**

Use the slider to adjust the season from summer to fall. You can customize more options in the material via the "Season Settings" of the shader. Adjusting these settings will largely influence the appearance of your leaves in different seasons.



The color change will be reflected in the global gizmo.



- **Snow**

Use the slider to adjust the snow accumulation. You can customize more options in the material via the "Season Settings" of the shader. Adjusting these settings will largely influence the appearance of your snow.

## Known Bugs

- If after adding new vegetation prefabs to the scene, the sliders don't seem to influence the wind, don't worry, this is just a default glitchy behavior. To fix this issue, copy and paste a new controller into the scene or reload the scene again. This should help resolve the problem.
- When opening a new scene, the wind might default to one phase or another, and the leaf and bark material will mismatch settings. To fix this, move the sliders around or change the wind type.
- When you save the scene, the wind reverts to the default settings of the materials, which is a glitch. It's best to save your scene first then change the gizmo sliders afterward, and refrain from saving again. This is only a scene glitch and won't reflect the settings from play mode or build as the wind settings are stored directly from the gizmo to materials.

## Known Issues

It has been observed that moving sliders with the global controller in a larger scene might appear laggy if using a **large amount of Unity prefabs**. It is worth noting that this issue might be resolved in the future and can now be avoided by using the **shaders with terrain trees** and **terrain details**. TTFE has improved integration with the terrain system so [using the terrain to paint plants is recommended](#).

# Shaders Overview

To effectively adjust TTFFE shaders to fit specific requirements, it is paramount to possess a comprehensive understanding of their strengths and limitations. The shaders are capable of handling various shapes of trees, however, the wind effect applied to the trees will behave realistically based on their size and height. It's possible that this may not produce the desired outcome in some cases, so you may need to make some adjustments yourself.

The main tree shapes are as follows:

1. Simple tube - Tall trees and pines.
2. Round - Broccoli-like trees that can also have a thick base.
3. Irregular - The most difficult type of tree, branch bending would be impossible without Pivot Painter.



When creating a tree in your 3D application, consider atlasing your textures based on the tree's shape, as different shapes require different settings and materials, because the wind mask must correlate with the trunk thickness.

The bottom image depicts how the wind mask can vary across different trees (Fig. A).

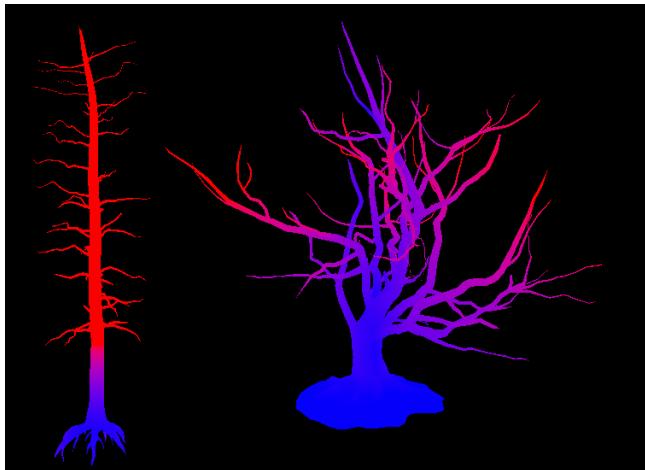


Fig. A

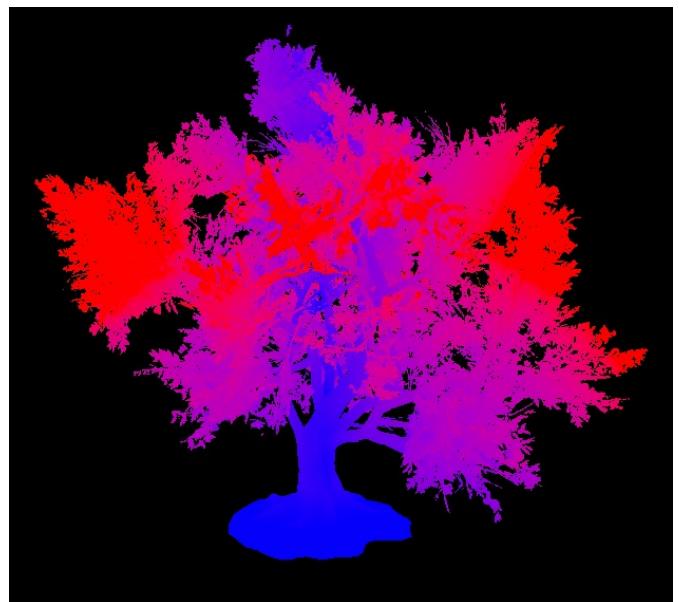


Fig. B

It's important to ensure that **bark and leaf materials** have **the same mask and wind** settings to avoid mismatch (Fig. B).

## How it Works Behind the Hood

TTFE uses a series of vertex position nodes multiplied by UV texture coordinates, vertex normals, and tangents to rotate the entire mesh based on its point of origin. It does this in more than a few phases giving the illusion of more complex bending.

Inside the shader, the wind system is divided into 3 sections; Trunk wind, branch wind, and leaf wind which contains all the various leaf noises for cards.

*Due to the employed method, **visible stretching** may occur in the trunk and branches. Adjust the material settings until the desired outcome is achieved.*

## Limitations

- TTFE is designed to work with instancing, static batching is not supported.
- TTFE is slightly more expensive for mobile but can be optimized by removing features.
- Shader Model 4.5+ capable devices are required (Desktop / High-End Mobile).
- While Mobile is supported, the scenes are intended for performance testing and will not be suitable for mobile devices (performance-intensive scenes).
- Only the latest LTS Unity versions are officially supported.

## Vertex Colors

The TTFE feature is functional without the implementation of vertex colors. However, green vertex color is necessary for the "flutter feature" to work on leaves but grass does not require any vertex colors. Vertex colors Red (self-shading/Ao and seasons) and green (leaf flutter) are used.



Note that other features, such as seasonal color changes and self-shading, work without vertex colors, but the presence of vertex colors enhances the overall effect.

You can debug vertex colors by enabling “[DEBUG] Compute Vertex Colors”.

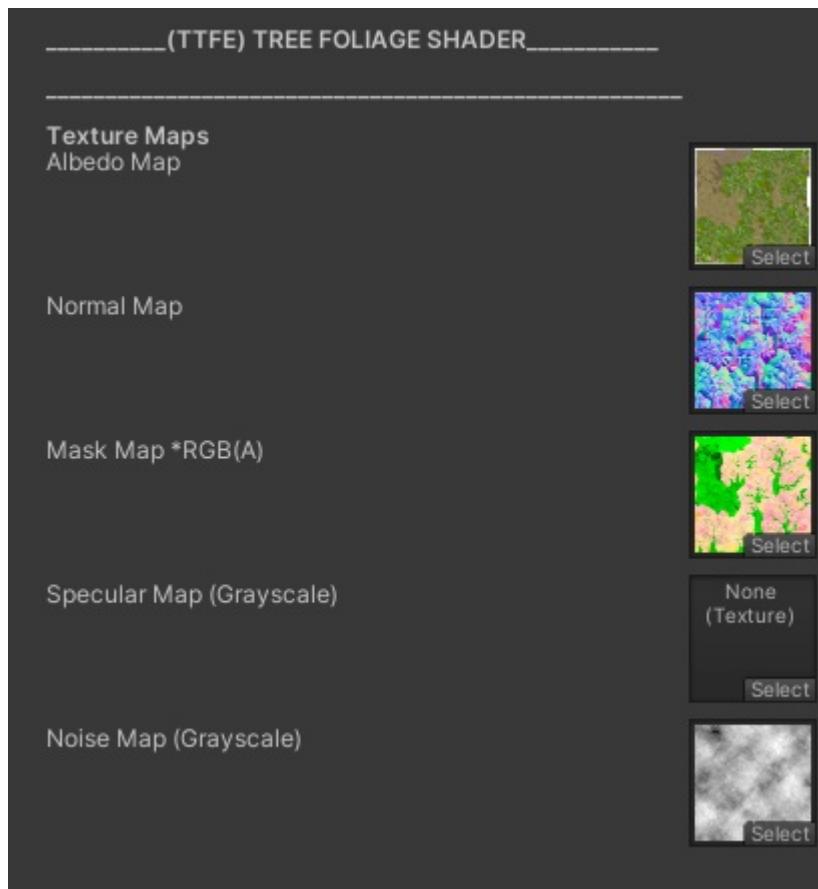
# Shader Settings

TTFE includes six shaders:

1. (TTFE) Grass Foliage
2. (TTFE) Roots
3. (TTFE) Tree Bark
4. (TTFE) Tree Foliage
5. (TTFE) Tree Billboard
6. (TTFE) Uber Cliff

We'll go over some of the settings below. For the sake of simplicity, only the relevant settings will be documented.

## Texture Packing



- **Mask Map \*RGB(A)**
  - R - Branch mask (masks out non-leaves in the texture and prevents them from changing color with seasonal change).
  - G - Ambient occlusion.
  - B - Translucency.
  - (A) - Smoothness.

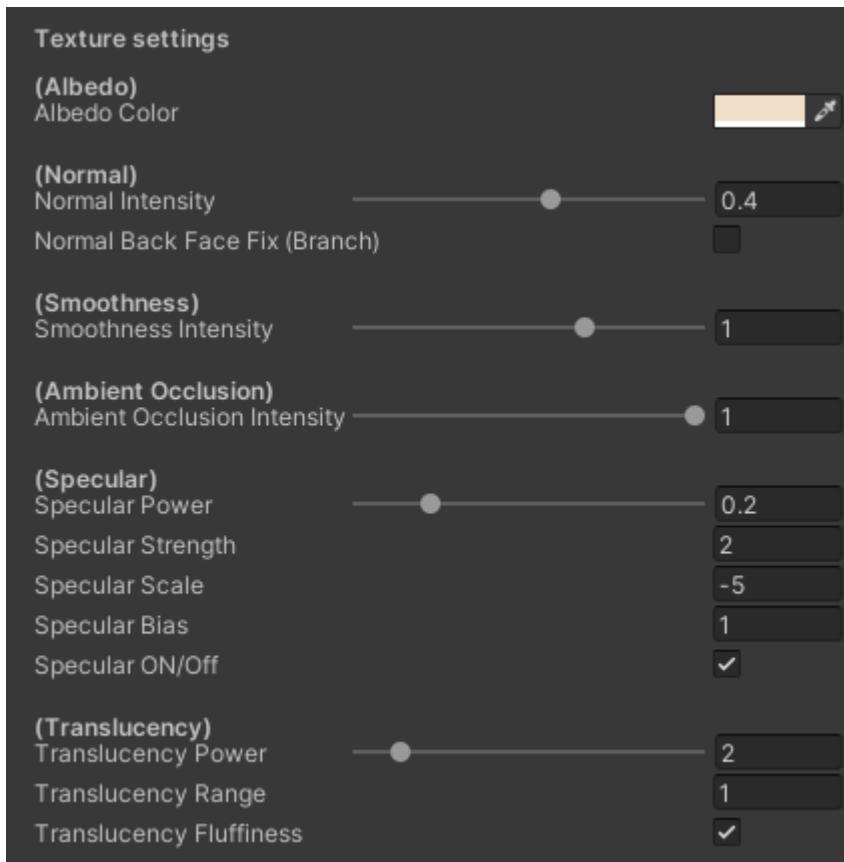
- **Specular Map**

(Grayscale) - White is specular, black is not specular (Works only with "Specular Backface Occlusion" on).

- **Noise Map**

(Grayscale) - Used as a texture mask for season color.

## Texture Settings



- **Normal Back Face Fix (Branch)**

This branch shading is used for cards with dead or dry branch textures. It corrects the normal map on the backside by flipping the Z channel.

- **Specular**

Attempts to remove the specular from the backside by recalculating the fresnel. An additional specular map can be added for more control (works with the "Specular Map" channel).

Previously known as "Specular Backface Occlusion (deprecated)", this new method uses vertex tangents and fresnel calculation, making it very efficient and cost-effective.

\* *It's designed to work together with "Mobile Shading (World Up)" and should only be used once this toggle is on.*

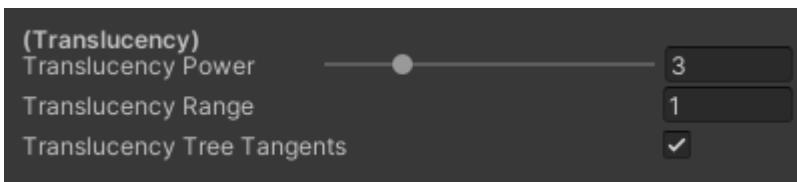
- **TTFE Translucency**

New and Innovative deferred translucency that gets automatically occluded in shadow by a feature called “Translucency Ao”. Works in Built-in, URP, and HDRP (only additional to the diffusion profile In HDRP).

- **Translucency Power**  
Adjust translucent intensity.
- **Translucency Range**  
Adjust translucent thickness offset.
- **Translucency Fluffiness**  
Used on grass to make it appear more lumpy.



- **Translucency Tree Tangents**  
Used on trees for better and more encompassing lighting.

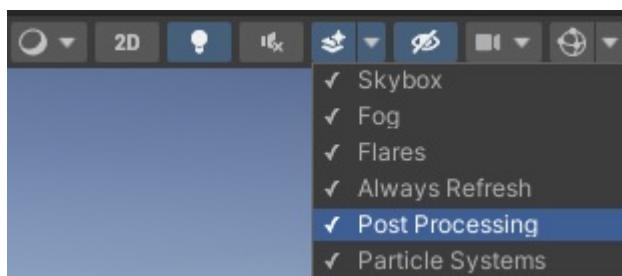


## Limitations

TTFE translucency requires one directional light to work as it's "light direction based". Omni lights are only partially supported as they will be locked to the sun's direction.

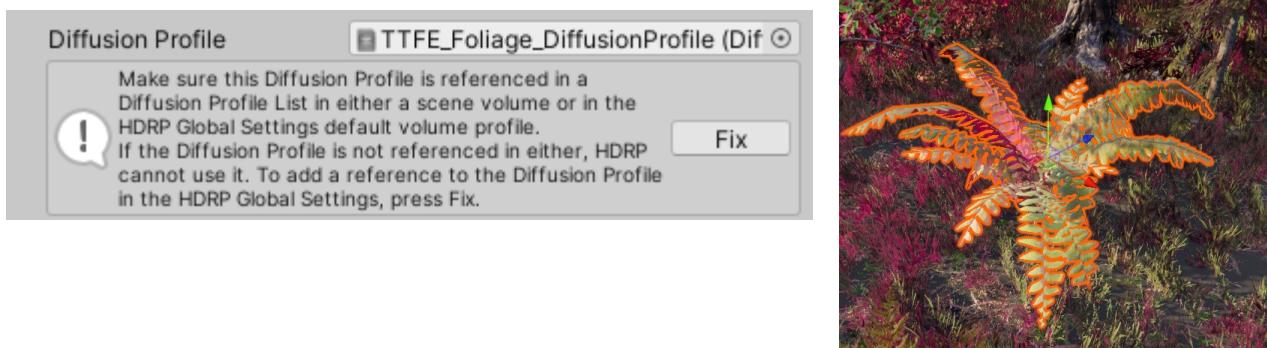
## Known Bugs

If you notice glitching, toggle the "sky and fog" button in the editor. This is just an editor glitch.

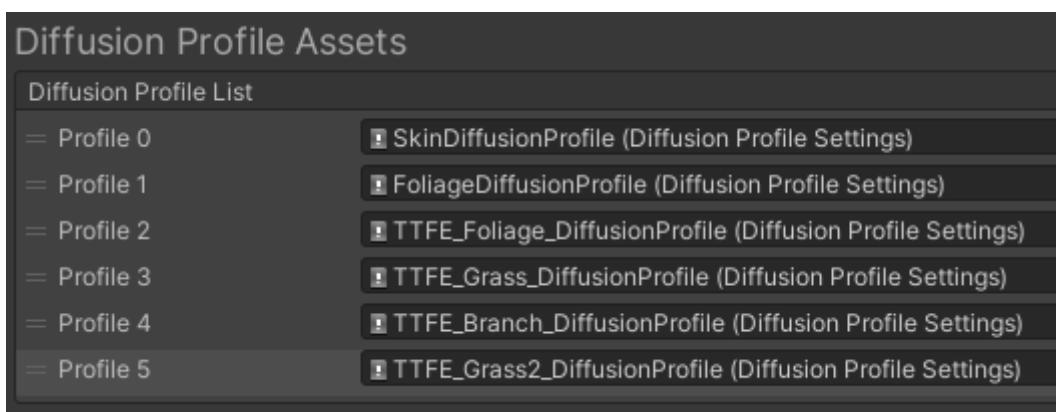


## High Definition Render Pipeline (HDRP)

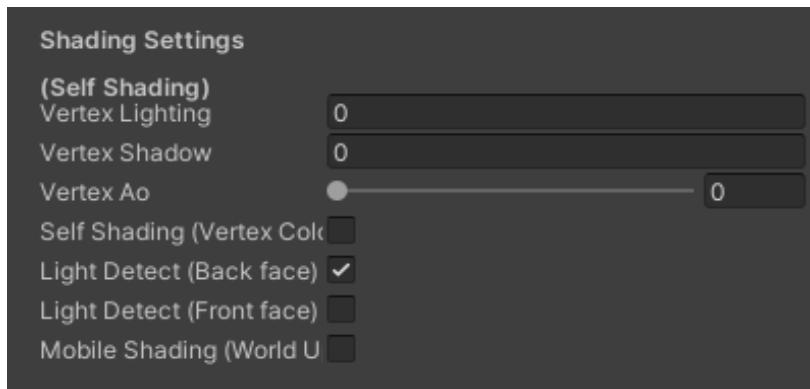
The models in the scene appear unnaturally green or pink. This issue is caused by a diffuse profile yet to be added to the HDRP asset. You can quickly fix it by selecting the material in question. When selected, you will see a warning window under the diffusion profile field in the inspector, simply click the “fix button” to add the supplied profiles to your project.



You can manually assign all the diffusion profiles in the "HDRP Global Settings" by going in to:  
**"Edit -> Project Settings -> Graphics -> HDRP Global Settings -> Diffusion Profile Assets"**

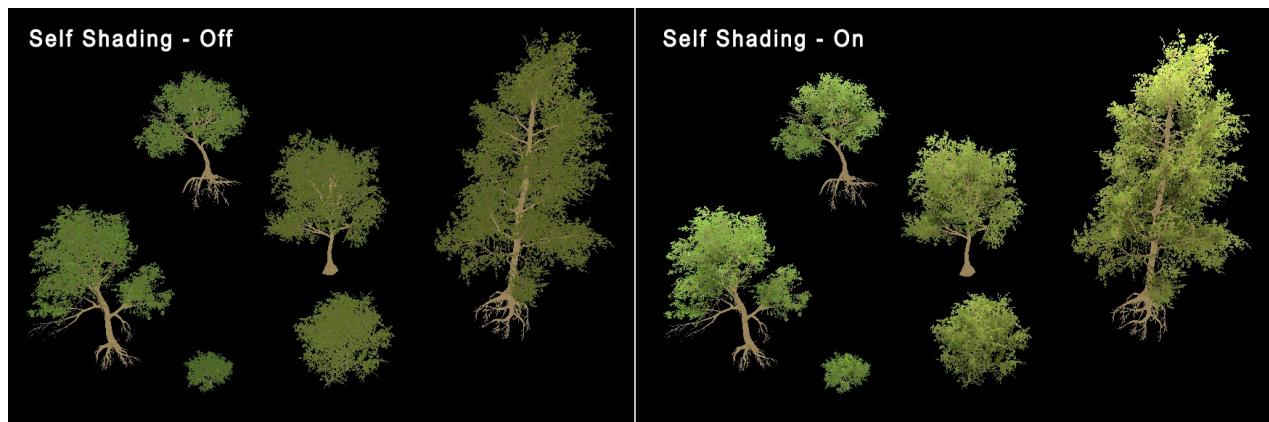


## Lighting/Shading Settings



- **Self Shading**

Get some extra shading out of your foliage for a more stylized look. Vertex position scale and offset are used to shade and light the tree, in addition, you can use vertex color R for ambient occlusion.



- **Vertex Lighting**  
Adds vertex position-based lighting.
- **Vertex Shadow**  
Adds vertex position-based shading.
- **Vertex Ao**  
Uses baked ambient occlusion from the mesh vertex color R for extra shading. Bake vertex color R in your 3d modeling program.
- **Self Shading (Vertex Color)**  
Turns the self-shading On/Off.
- **Mobile Shading (World Up)**  
Foliage with simple "vertex position Y shading" ("world up" toggle). Shading conforms to surfaces such as the terrain. It's usually used on grass and is more optimal for performance.

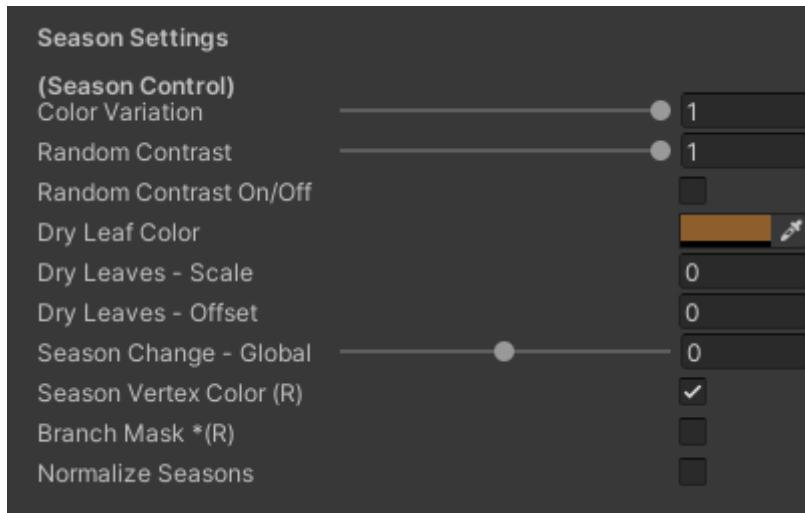
- **Light Detect (Back face)**

Tackles the age-old problem of dark back faces in Unity by re-calculating vertex normals. This option is enabled by default.

- **Light Detect (Front face)**

Corrects the vertex shading of the front face to be identical to the default Unity shading. Can be used after “Light Detect (Back face)” has been turned on.

## Season Settings



- **Season Control**

Color and seasonal changes are based on vertex and world position.



- **Color Variation**

Each plant will have a random color based on world position.

- **Random Contrast**

Each plant will have a random contrast based on world position (Expensive feature hence it can be toggled off).

- **Random Contrast On/Off**

Turn on or off the random contrast.

- **Dry Leaf Color**

Leaf color in fall.

- **Dry Leaves – Scale and Offset**

Adjust the radius of the dry leaves.

To lock the season color so it can't be changed with the (TTFE) GLOBAL CONTROLLER, set "Dry Leaves - Scale and Offset" to "0,0" (default values are "1,1"). Is useful for having dry or different-colored trees in any season.

- **Season Change - Global**

Adjust the intensity of the seasonal change. Correlates directly to the "season" slider of the global gizmo.

- **Season Vertex Color (R)**

Uses the vertex color red to drive the seasonal change, instead of the vertex position.

- **Branch Mask \*(R)**

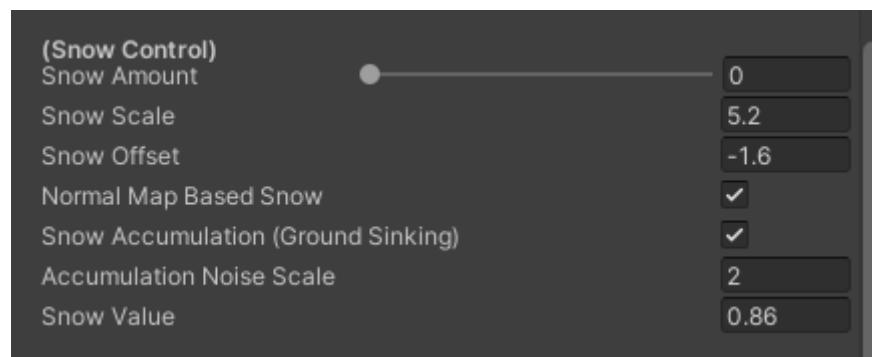
Prevent seasonal color changes in texture atlas branches with masking (R-channel).

- **Normalize Seasons**

It prevents season colors from glitching with world position.

- **Snow Control**

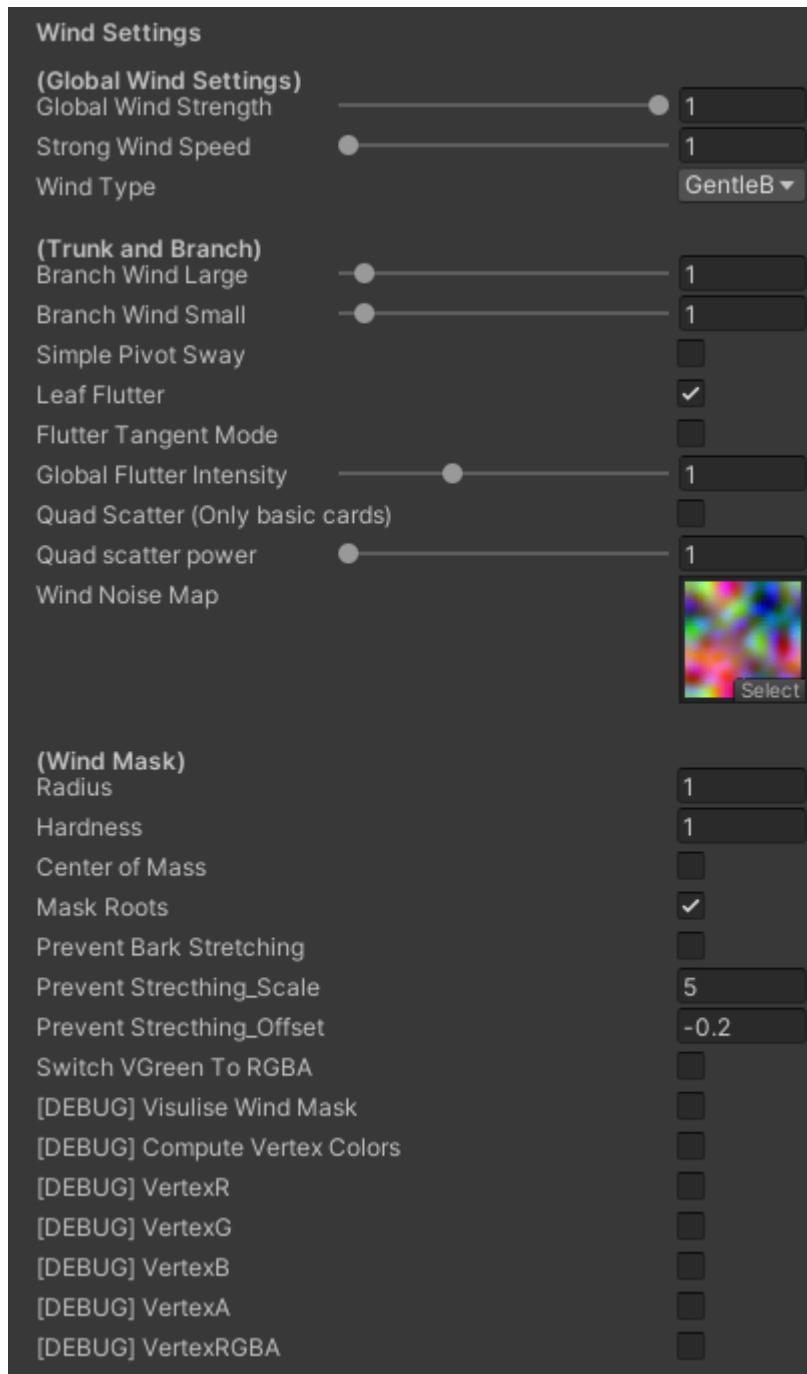
Snow accumulation is based on the vertex and world position. The shader uses a simple color on foliage to save performance costs and a procedural snow material made internally from noise on non-foliage meshes. All TTFE shaders now support the snow feature.



- **Snow Amount - Global**  
Adjusts the global snow accumulation amount (used by the global controller). Correlates directly to the "snow" slider of the global gizmo.
- **Snow Scale and Offset**  
Adjust the radius of the snow.
- **Normal Map Based Snow**  
Snow accumulation is driven by the normal map (map channel G).
- **Snow Accumulation (Ground Sinking)**  
Control the amount of snow covering the grass. Internal noise is used and multiplied in the alpha to create the effect that the grass is buried beneath the snow.

- **Accumulation Noise Scale**  
Adjust the noise scale of the accumulated snow.
- **Snow Value**  
Adjusts the albedo value of the snow.

## Wind Settings



- **Global Wind Settings**

Correlates directly to the (TTFE) Global Controller sliders.

- **Global Wind Strength**  
Adjust the intensity of the main wind.
- **Strong Wind Speed**  
Adjusts the wind speed.
- **Wind Type**  
Use the dropdown to change between wind phases.

- **Trunk and Branch**

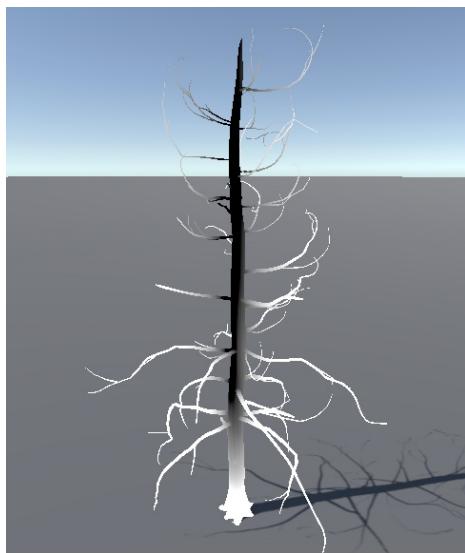
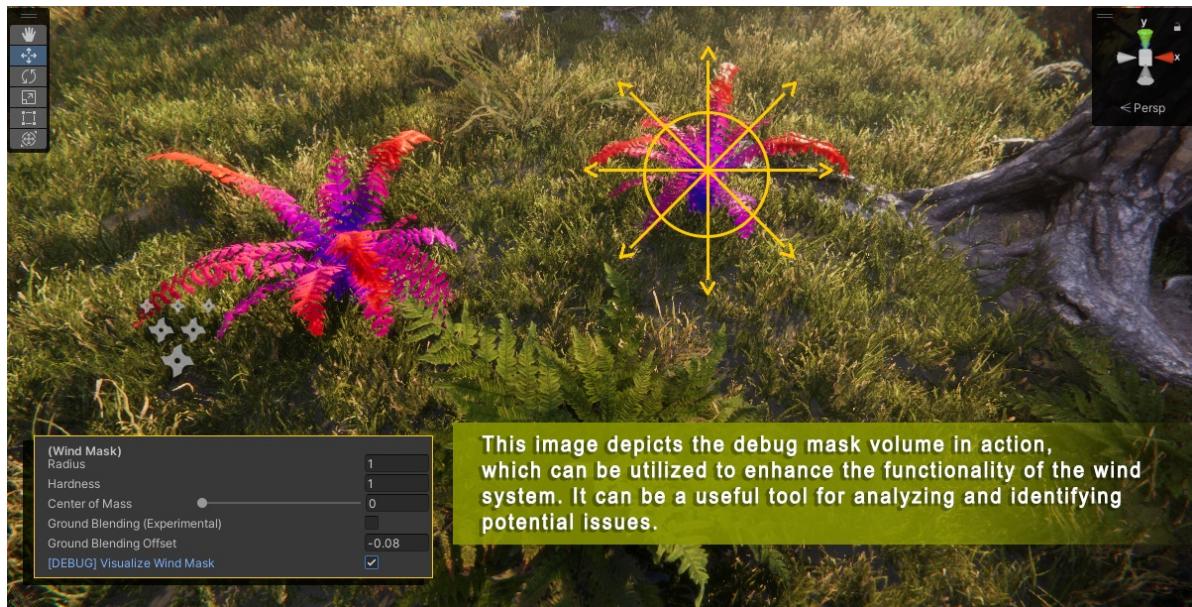
Adjust and customize branch and leaf settings.

- **Branch Wind Large and Small**  
Adjust the large and small wind noise of your branches. You will need to adjust your branch bending intensity according to your preference because in some cases the branch bending won't be very visible.
- **Simple Pivot Sway**  
Best fitted for tall trees and pines with cylindrical shaped bark (prevents strange bark wobble).
- **Leaf Flutter**  
Turns on leaf fluttering.
- **Flutter Tangent Mode**  
Leaf flutter is based on vertex tangent rather than vertex position.
- **Global Flutter Intensity**  
Adjust the intensity of the flutter.
- **Quad Scatter (Only Basic Cards)**  
Random pivot motion based on vertex tangents. This is only suitable for simple cards as intersecting planes will move independently from each other.
- **Quad Scatter Power**  
Adjust the intensity of the quad scatter.
- **Wind Noise Map**  
Internal 2D texture noise (3 vector RGB), generates an intense wind pulse that affects grass and trees (only available for "Strong Wind" type). The effect will be turned off if this texture is missing.

- **Wind Mask**

The meat and potatoes of the Toby Foliage Engine, the wind mask is essential for customizing how your trees behave in different wind conditions. A vertex-position volume sphere, masking out the wind on your trunk and leaves. Its radius can be controlled via the material settings.

Using vertex color as a mask might become available in future releases.



\* Due to its rudimentary and limited nature, the mask isn't perfect and never will be, therefore additional options are available.

- **Radius**  
Adjusts the volume size to mask wind in your material settings.
- **Hardness**  
Adjusts the intensity of the volume mask.
- **Center of Mass**  
Creates a center of gravity making the mesh less affected by wind in its center.
- **Mask Roots**

It prevents the roots from being affected by the wind.

- **Prevent Bark Stretching**

Helps to reduce stretching on the sides of the bark where branches connect.

- **Prevent Stretching Scale and Offset**

Adjust settings to reduce the stretching amount.

- **[DEBUG] Visulise Wind Mask**

Tick "[DEBUG] Visualize Wind Mask" to visualize wind physics.

- **[DEBUG] Compute Vertex Colors**

Visualize all vertex colors available on the mesh.

# Other Shaders

## (TTFE) Tree Billboard

Identical to the tree foliage shader but with fewer features.

## (TTFE) Roots

Basic root shader.

## (TTFE) Uber Cliff

Triplanar cliff shader with grass/snow coverage and correct edge slope angle.

It contains three tri-planar material layers; “Rock Layer”, “Dirt Layer”, “Coverage Layer” and additional “Detail Texture” and Edge Normal”. The new version was updated with better slope function and snow. Snow settings are the same as the foliage shaders with the additional benefit of “snow slope” options.



## (TTFE) Mesh

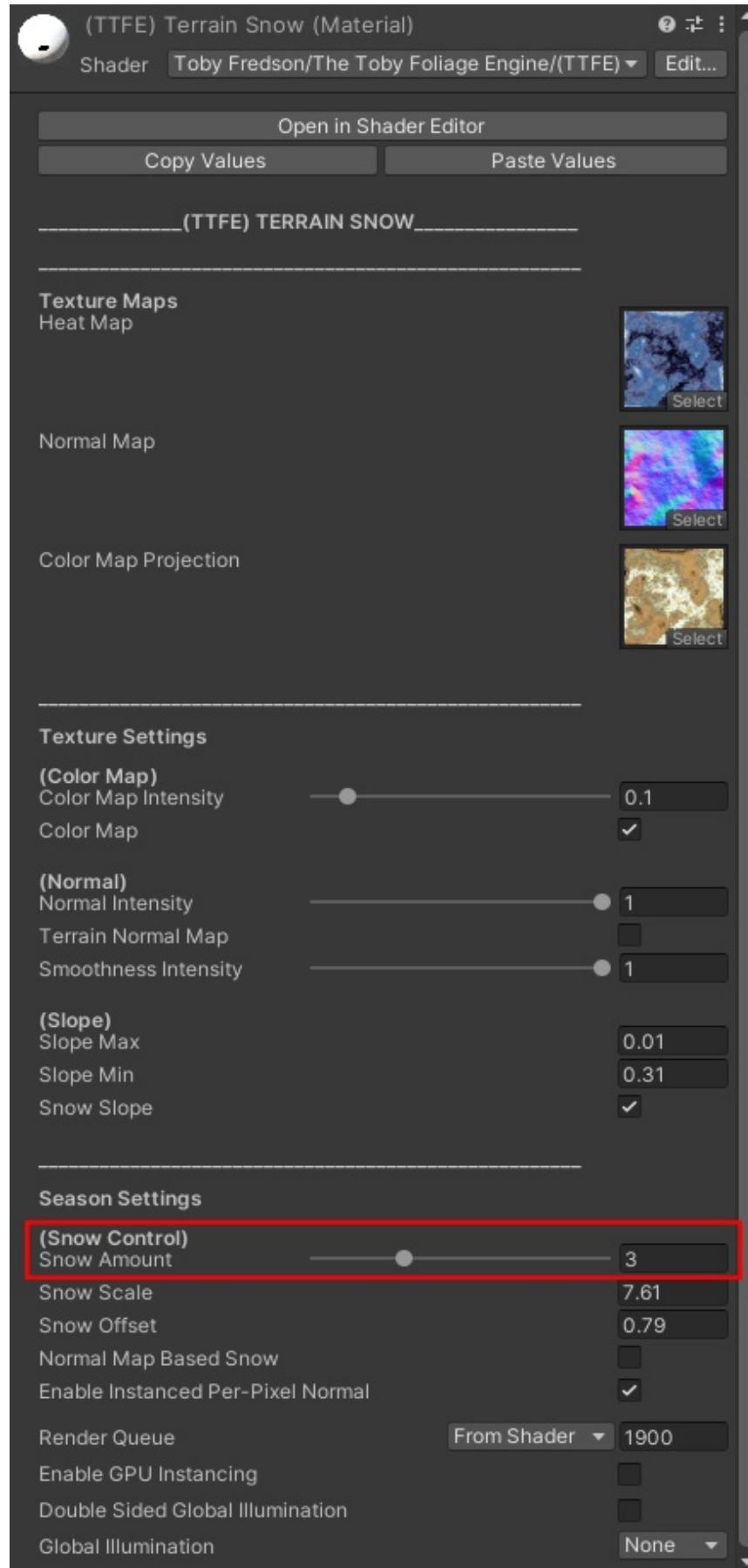
Basic and performant PRB shader with crossfade. Also supports snow.

## (TTFE) Terrain Snow

Basic terrain shader with added support for snow slopes, similar to the mesh and cliff shader.



■ Global Controller and Terrain Shader! The current version of TTFE is designed to work with foliage and mesh shaders. It also includes a terrain shader that is still in development, so the global controller does not yet support the snow feature on terrain. However, you can manually adjust the snow.



# Optimization

Rest assured that each and every node in the shaders has been meticulously tested and optimized to perfection. While the scenes and models may not be ideal for mobile devices, they serve as an excellent benchmark for testing TTFE against other shaders.

Here's a list of expensive features for mobile devices that may require optimization.

- **Light Detect (Back face)**

This option is enabled by default and could be expensive. "Mobile Shading (World Up)" is recommended for the best performance.

- **Random Contrast**

It has proven surprisingly expensive, most likely, because it uses a "blend node". Another one best left avoided.

- **Ground Blending (Experimental)**

This is an experimental feature, it enables the bottom vertices to be stretched downwards. Initially conceived to fix floating grass on slopes, its performance cost shouldn't be taken lightly. Avoid this feature unless you want to use it. ***Not suitable for games!!***

## GPU Instancing

Is supported in Built-in but. In URP and HDRP you will have to turn off the SRP batcher.

TFFE runs very well without instancing, so manually turn on instancing on your materials as it is disabled.

\* There seems to be a bug that breaks instancing when "cross-fading" is enabled. If you experience issues make sure your prefab contains a low triangle count and "LOD crossfading" is turned off otherwise it might break instancing.

## Performance Intensive Scenes

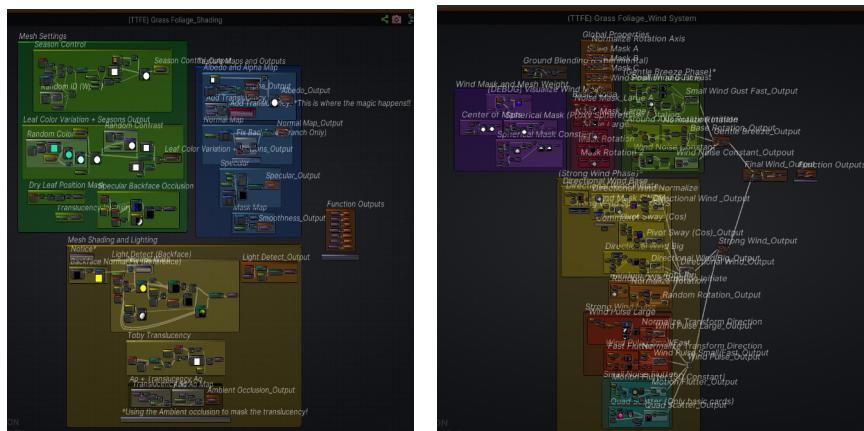
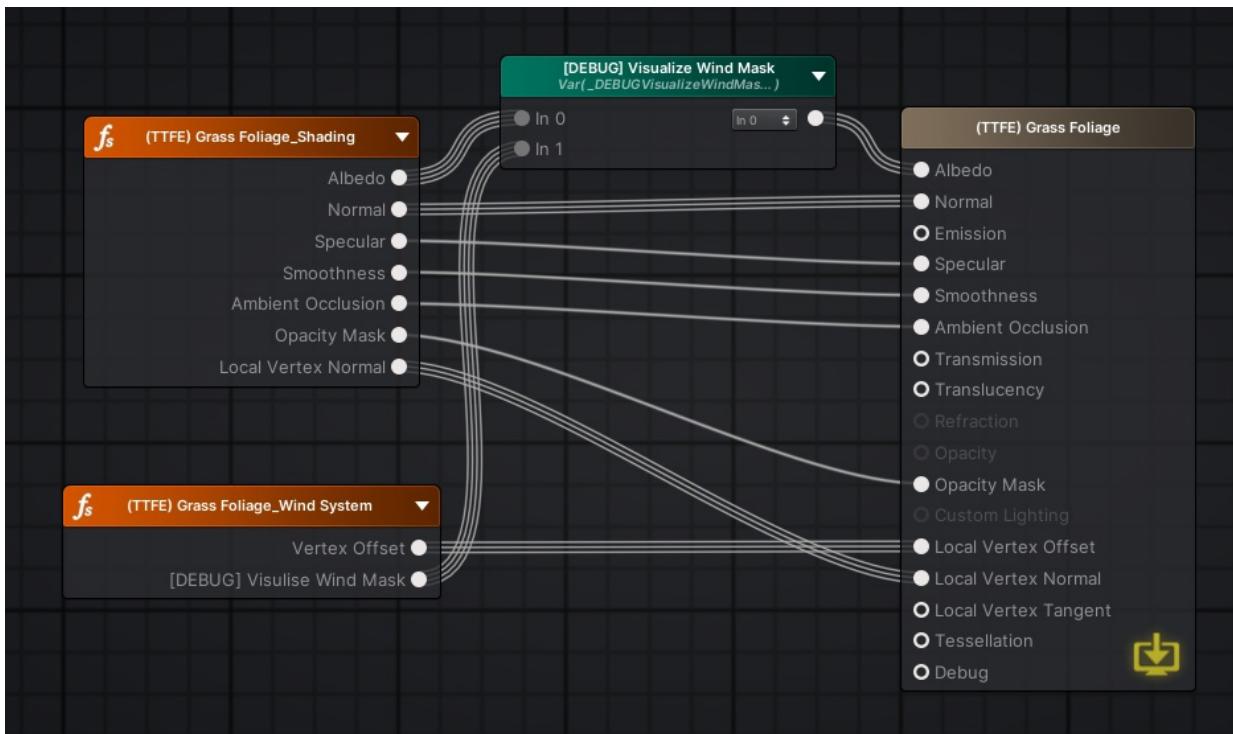
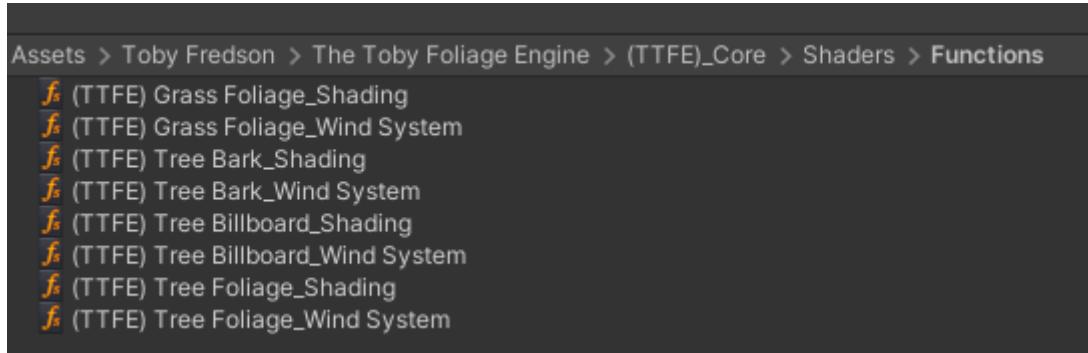
The demo scenes were created to test the performance of the TTFE shaders, which can be compared with other shaders or used for testing. For lower-end devices, you can set the "LOD bias to 1" in your "Preferences -> Quality settings".

In HDRP the settings are defaulted to 1. If you want good visual quality set "LOD bias" to 2.

# Amplify Shader Functions

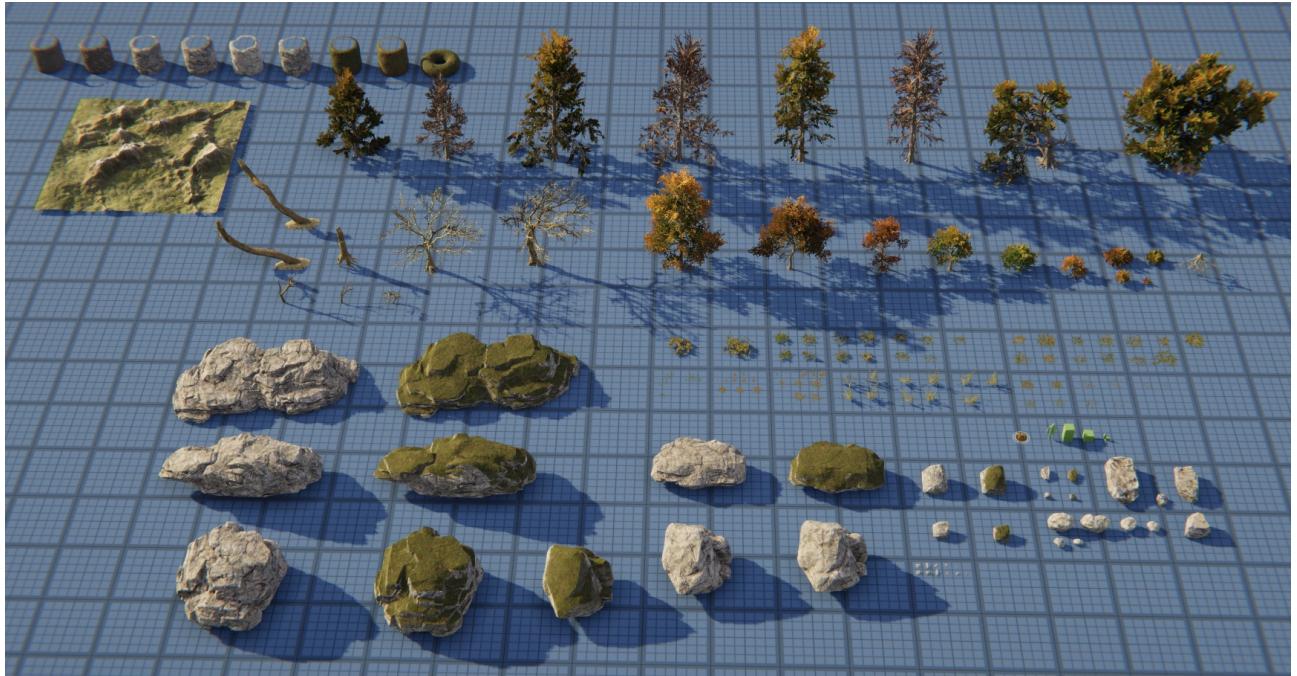
The node systems have been methodically arranged into Amplify Shader functions, rendering them easily accessible and comprehensible to integrate with your shaders. The system is well-organized and user-friendly, providing an efficient and effective workflow for users to optimize their performance.

Foliage shaders contain 2 types of nodes, "\_Shading" and "\_Wind System".



# Original Assets

The Toby Foliage Engine includes a range of unique art assets that are well-optimized and game-ready, all of which are original.



# Credits

Credits for the “*Specular Backface Occlusion*” and “*Light Detect (Back face)*” systems go to [Hybris Factory](#).

[https://www.artstation.com/ivan\\_janik](https://www.artstation.com/ivan_janik)

Special thanks for helping me develop the “Translucency Occlusion” system.

# Acknowledgement

Inspired from [The Visual Engine](#) By [BOXOPHOBIC](#).