

The Toby Foliage Engine

Version 2.1.0

(Player Interaction and Amplify Impostor)



Documentation

Toby Fredson

florianalexandru05@gmail.com

Table of Contents

Table of Contents	2
Product Overview	4
Optimization and Compatibility	4
Limitations	5
Terrain Requirement	6
(TTFE) Global Controller	7
Global Wind Parameters	8
Global Season Settings	9
Debug TTFE/Advanced Settings/Known Issues & Workarounds	10
Shaders Overview	11
How it Works Behind the Hood	12
Vertex Colors	13
Shader Settings	14
Texture Packing	14
Texture Settings	15
TTFE Translucency	16
Translucency Limitations	17
Shading Settings	18
Season Settings	19
Wind Settings	23
Wind Noise/Pivot/Trunk and Branch	24
Player Interaction	24
Cloth/Appendages	25
Wind Mask	26
Wind Mask Settings	27
Debug Wind Mask	28
Other Shaders	29
(TTFE) Terrain Snow	31
Additional Features	33

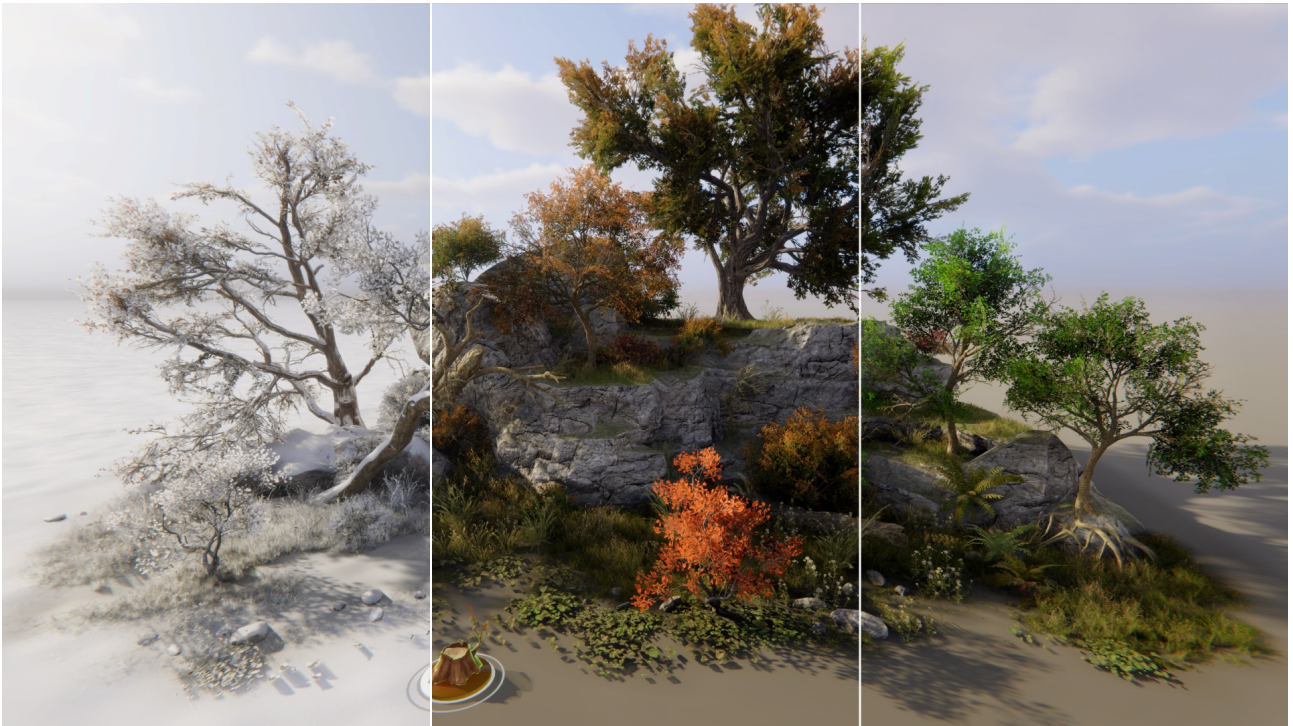
Material Matcher/Terrain Plants to Prefab Converter	33
Specular Fade	34
Optimization	35
GPU Instancing/Mobile and Low-End Shader Variants	35
Amplify Shader Functions	36
Original Assets	37
Release Notes	38
Credits/Acknowledgement	39

Product Overview

Introducing the Toby Foliage Engine (TTFE), an innovative single-pivot vegetation system that cleverly simulates multiple pivots per mesh, much like Pivot Painter. These Amplify shaders deliver a complete solution for achieving realistic wind animation and plant shading. Thanks to its seamless integration and intuitive interface, applying TTFE to any mesh is quick and straightforward. There is **no need for complex conversions or tedious setup** steps — the shaders handle everything for you.

TTFE works automatically alongside standard PBR shading. The wind physics are driven by internal mesh-based vertex position volumes, allowing you to easily control the size of the volume to mask wind effects directly in the material settings. You can also fine-tune the overall wind behavior through the same material parameters.

Built entirely in the Amplify Shader Editor, these shaders provide fully customizable features and feature clean, well-organized nodes that are easy to understand and learn from.



Optimization and Compatibility

✓ TTFE supports various third-party terrain and vegetation tools like [Gaia](#) and [Nature Renderer](#).
(Tested with **Nature Renderer**).

- Complete Package Optimization: Streamlined for performance across all supported platforms.
- Modern Shader Integration: Utilizes global variables for fast and flexible shader management.
- Full Compatibility: Supports Built-in, URP, and HDRP pipelines, with optional fixes for Built-in rendering.

Limitations

- ✗ TTFE is designed to work with instancing; static batching is not supported.
- ✗ While Unity's built-in instancing is supported, using a third-party instancing solution is recommended for optimal performance, as the default option may not yield significant performance improvements.
- ✗ TTFE is slightly more expensive for mobile, but it can be optimized by removing features or using the mobile version of the shaders.
- ✗ 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).
- ✗ **Static batching is not supported.**
- ✗ Only the latest Unity (LTS) versions are officially supported.

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

■ 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. Grass does not require any vertex colors.

■ TTFE is not a perfect alternative and may not produce results identical to those of Pivot Painter. The wind only relies on a single point of rotation!!

■ **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".

■ Setting your prefabs to static will break the shader due to a lack of support for static batching. If you want to bake lighting, first bake and then uncheck "static" after you're finished.

■ Subsurface Scattering bug in Built-in RP Deferred due to the "light direction node". Use the mobile version of the shaders if you encounter lighting glitches.

■ Player Interaction Limitations

- **Player Interaction (Shader-based)**

Shader interaction with touch bedding animation. Works with any prefab as well as with terrain plants. It can be switched between multiple players.

- ✗ Limitations: Supports only one interactor at a time.

- **Physics Interaction (Pivot bending)**

Works with an unlimited number of interactors at the same time.

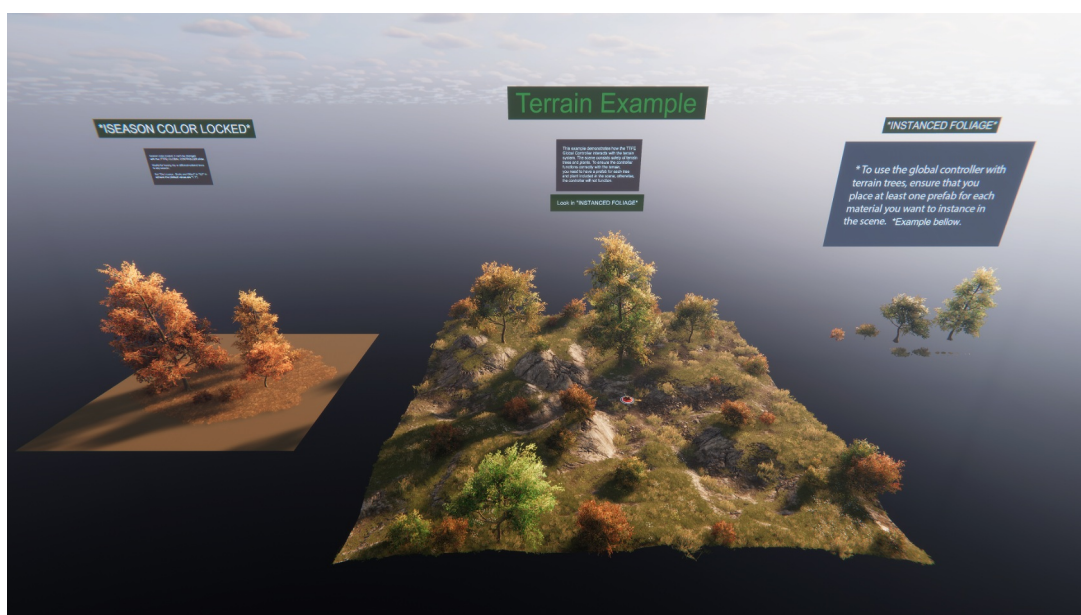
- ✗ Limitations: Doesn't work on terrain objects, as the terrain has its own collision system that is different from the default mesh. Use the "Terrain Plants to Prefab Converter" script as a workaround.

■ Terrain Requirement

■ To ensure compatibility with the Global Controller (TTFE) when using prefabs as terrain trees or details on either Unity's default terrain or TTFE terrain, at least one instance of each prefab mesh with its associated material must be present in the scene. Without an instance of each material, the Global Controller will not function correctly.



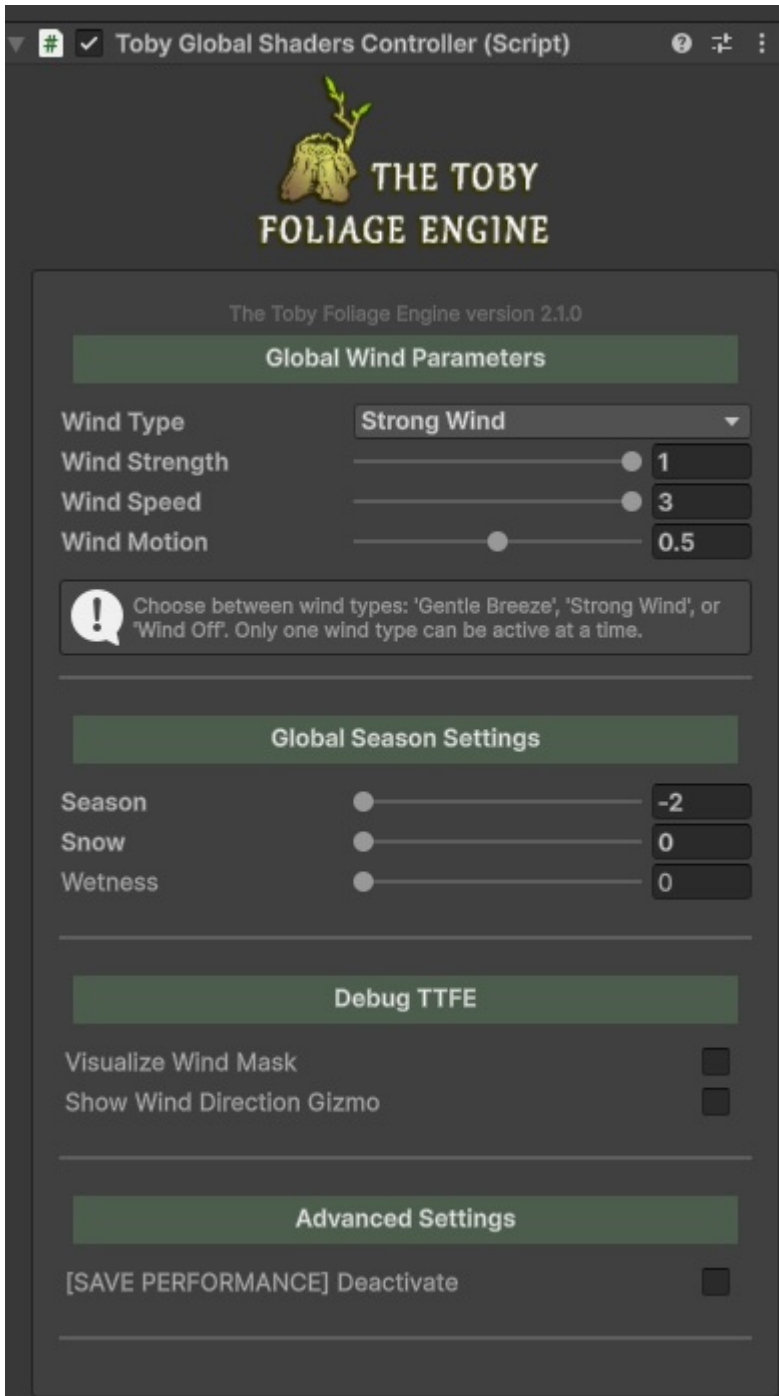
See “TTFEScene_Terrain Example”.



(TTFE) Global Controller

This tool adjusts material settings for all shaders at once. 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**

Dual Wind Systems: Choose between "Gentle Breeze" and "Strong Wind" modes, interchangeable in real time.

- **Gentle Wind**

Gentle and realistic wind effect based on pivot rotation. This effect mimics the subtle movements of plants growing in clusters, such as those found in wheat fields. It will influence grass and trees, but will not include bending of the tree trunks in any direction. However, the branches will still respond to the wind.

*(Most performance friendly) **-6/10 FPS, 1.5/3 ms**

- **Strong Wind**

Intense wind pulses are driven by a 2D texture noise (3-dimensional RGB vector). This affects grass and trees, incorporating "Wind Motion" in the direction of the motion vector. The "Gentle Breeze" type contains a weaker version of the "Wind Motion" effect.

*(Most performance intensive) **-10/20 FPS, 2.5/4 ms**

- **Wind Off**

Disables wind effects in the scene but keeps the shader active. The shader functions for wind remain operational even when not animating.

*(Saves no performance) **Null**

- **Wind Strength**

Wind intensity can be controlled and set to zero, but this is not the same as turning off the wind, and it may impact performance.

- **Wind Speed**

Controls the wind speed.

- **Wind Motion**

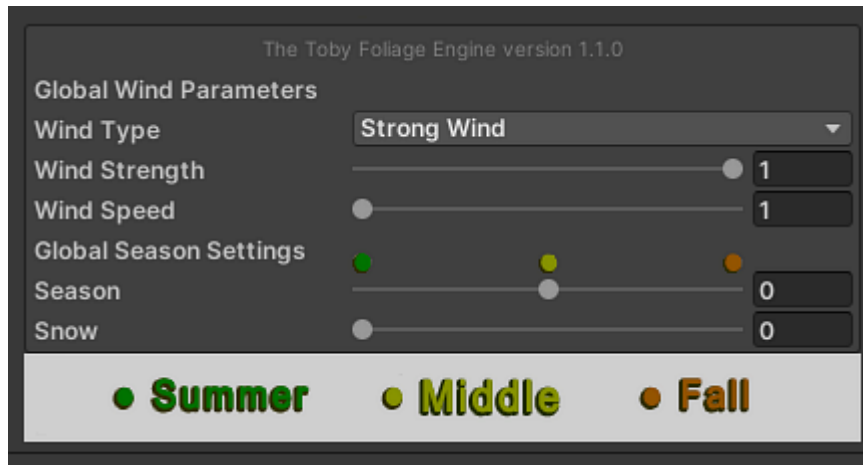
This affects grass and trees, incorporating "Wind Motion" in the direction of the motion vector. Rotate the Global Controller gizmo to change the direction of the wind vector.



Global Season Settings

- **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.

- **Wetness**

Use the slider to adjust the wetness accumulation.

Debug TTFE

- **Visualize Wind Mask**
View the wind mask volume of each plant globally using a color spectrum rather than by material (Better views in the shadeless debug view).
- **Show Wind Direction Gizmo**
Displays the directional vector gizmo for the wind. The controller model already has an arrow designed in.

Advanced Settings

- **[SAVE PERFORMANCE] Deactivate Wind**
Completely disable all window functions in the shader using static switch.

*(Saves performance) **-0 FPS**

Known Issues & Workarounds

- **Wind sliders not responding after adding new vegetation prefabs**
This is a known temporary issue. If the wind sliders cease to affect the vegetation after adding new plant prefabs, simply duplicate the Wind Controller in the scene or reload it. The problem is typically resolved immediately.
- **Wind phase and material mismatch when opening a scene**
When loading a scene, the wind may default to a specific phase, causing the leaf and bark materials to appear mismatched.

Fix: Simply move any wind slider or switch the wind type once. The materials will update correctly.

- **Wind settings reverting when saving the scene**
Saving the scene currently causes the wind settings to revert to the material defaults (a known editor glitch).

Recommended workflow:

1. Save your scene first.
2. Then adjust the wind gizmo sliders to your desired values.
3. Avoid saving the scene again afterward.

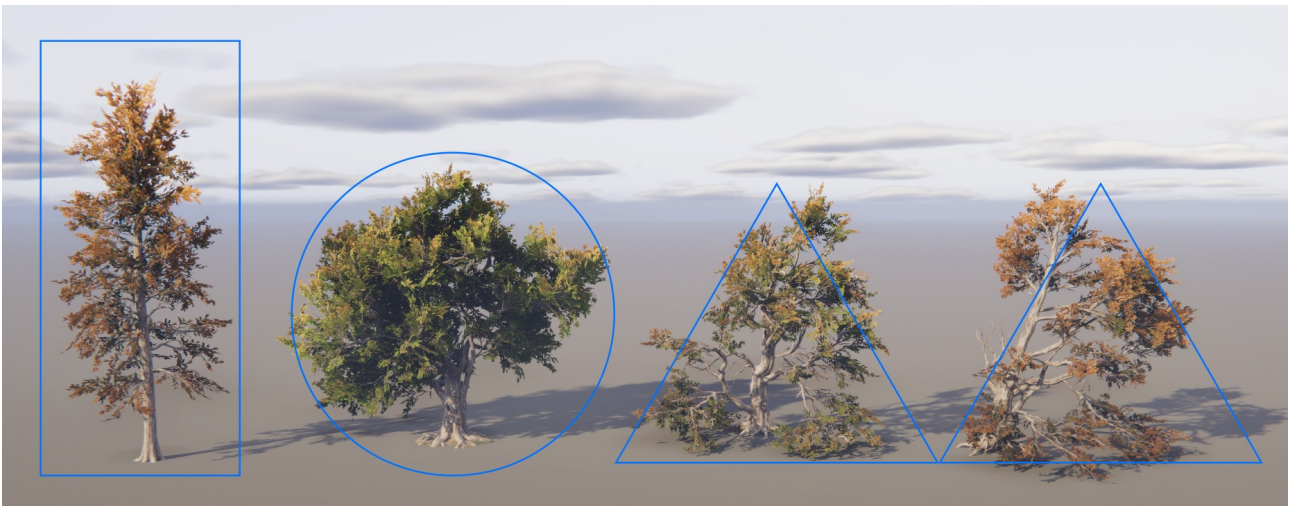
Note: This is purely an editor/scene glitch. Your wind settings are correctly stored from the gizmo directly into the materials, so they will work properly in **Play Mode** and in the final **Build**.

Shaders Overview

To effectively adjust TTFE 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).

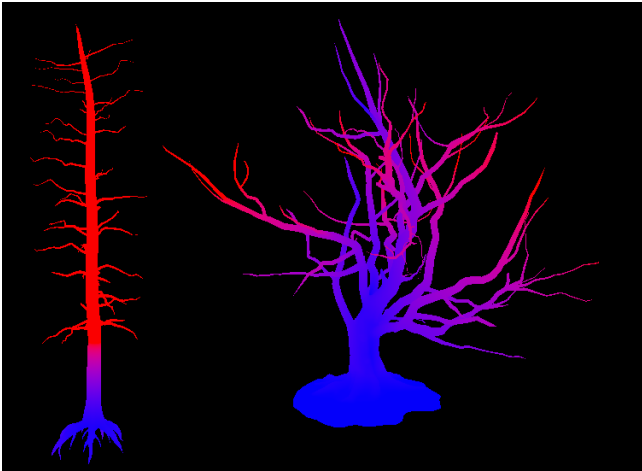


Fig. A

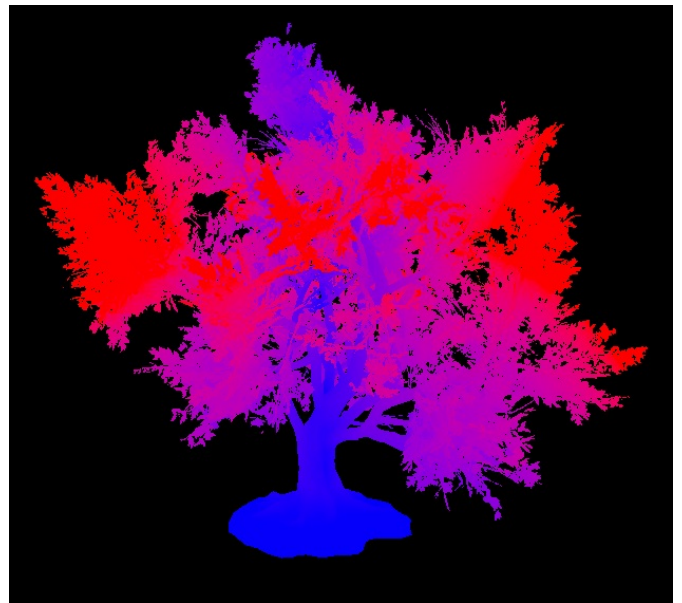


Fig. B

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

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.*

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".

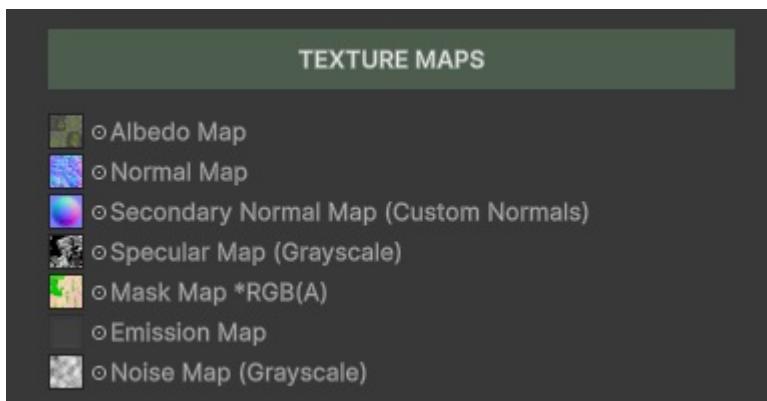
Shader Settings

TTFE includes 14 shaders:

1. (TTFE) Cloth
2. (TTFE) Grass Foliage
3. (TTFE) Grass Impostor
4. (TTFE) Mesh
5. (TTFE) Mesh Impostor
6. (TTFE) Roots
7. (TTFE) Simple Cliff Masked
8. (TTFE) Tree Bark
9. (TTFE) Tree Billboard
10. (TTFE) Tree Foliage
11. (TTFE) Tree Foliage Impostor
12. (TTFE) Uber Cliff Optimized
13. (TTFE) Water Beta
14. (TTFE) Terrain Snow

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

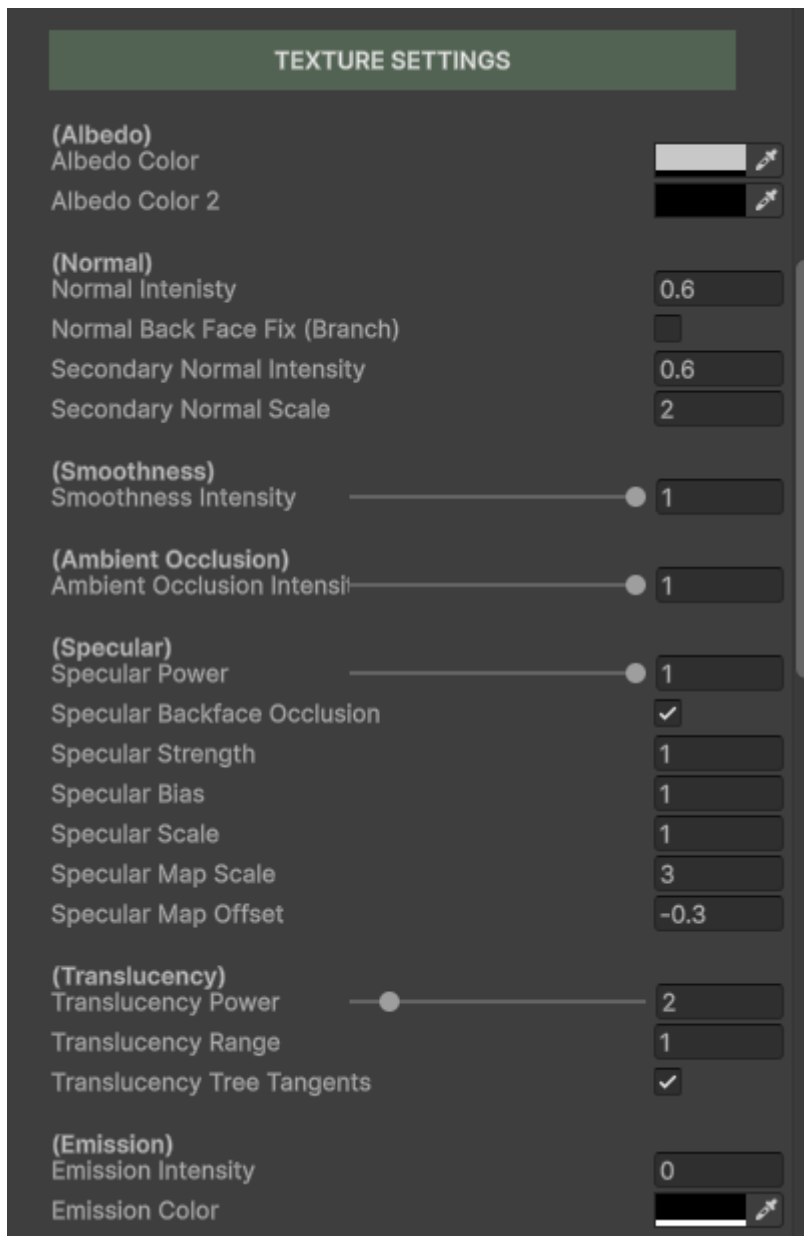
Texture Packing



- **Secondary Normal Map (Custom Normals)**
Assign custom normals to individual branch cards rather than utilizing custom vertex normals. It can be used for making branches appear lumpy.
- **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 seasonal color and random contrast.

Texture Settings



- **Dual Color**
Utilize Albedo Colors 1 and 2 to create a more vivid albedo effect, enhancing the intensity of your plants, particularly for fantasy and alien varieties.

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

This branch shading is used for cards with dead or dry branch textures as the obscuration of the backface renders it unsuitable for translucency. It corrects the normal map on the backside by flipping the Z channel.

- **Secondary Normal Intensity and Scale (Custom Normals)**

Use a secondary normal map. Good for adding custom normals to branches.

- **Specular**

Attempts to eliminate the specular highlight from the backside involve recalculating the Fresnel effect. You can also add a specular map for greater control, which works with the "Specular Map" channel. Previously referred to as "Specular Backface Occlusion (deprecated)," this updated method utilizes vertex tangents and Fresnel calculations, making it 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**

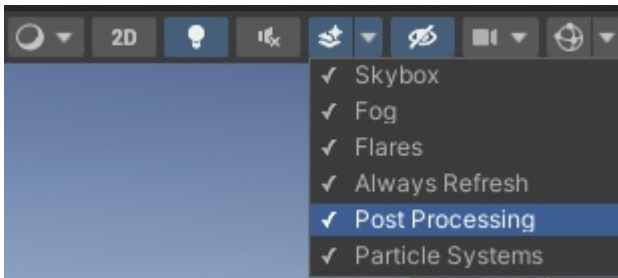
Used on trees for better and more encompassing lighting.

Translucency 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.

Translucency Known Bugs

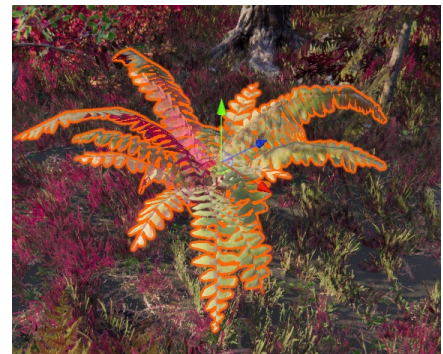
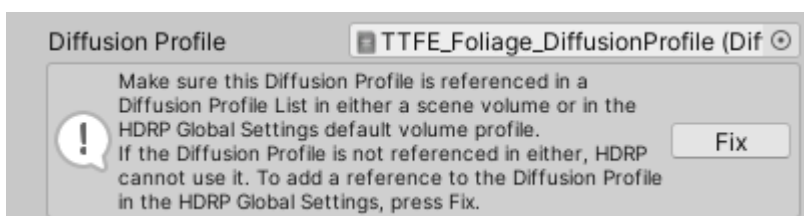
If you notice glitching, toggle the "sky and fog" button in the editor. This is just an editor glitch.



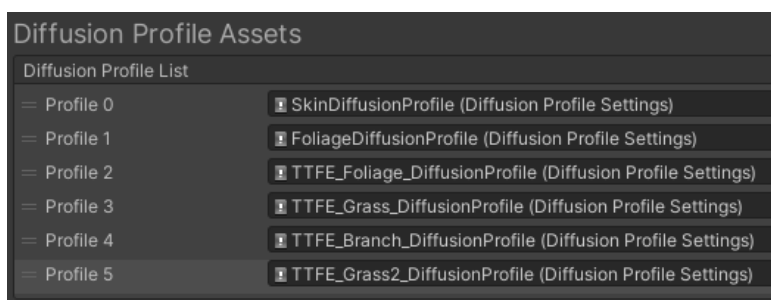
High Definition Render Pipeline (HDRP)

The models in the scene appear unnaturally green or pink. This issue occurs because the required diffuse profile has not yet been added to your HDRP asset. To fix it quickly:

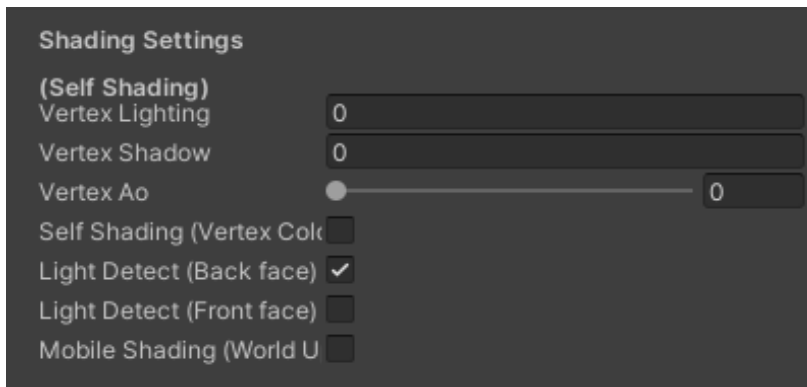
1. Select the affected material in the Project window or Hierarchy.
2. In the Inspector, look for the warning message displayed below the Diffusion Profile field.
3. Click the Fix button to automatically add the supplied diffuse profiles to your project.



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"** or adding the profiles to your post-processing volume.

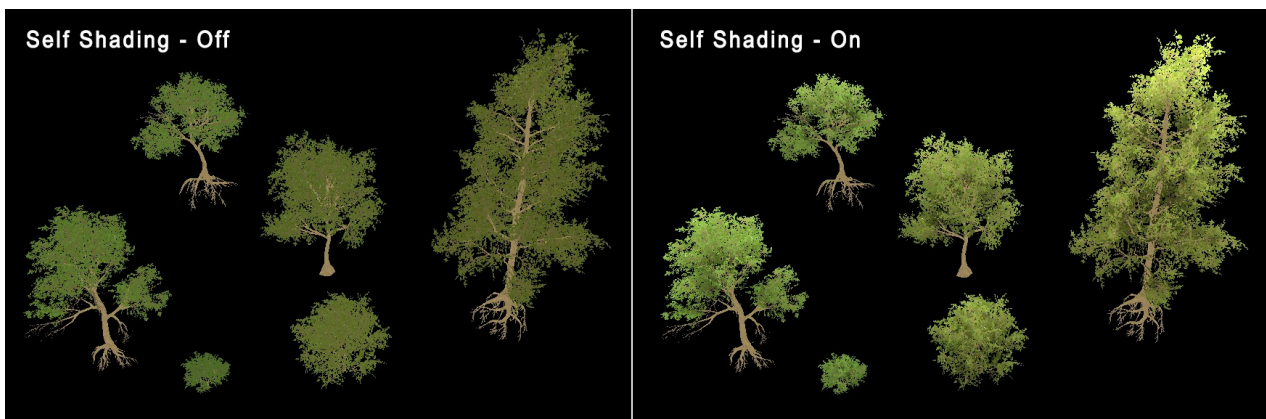


Shading Settings



- **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 is set to match the terrain tangents. It's usually used on grass and is more optimal for performance. It also works well on trees, but keep the Y rotation locked at 90 degrees. Rotating the mesh will completely break the shading.

- **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. More accurate front and specular shading.

Season Settings

SEASON SETTINGS

(Color Variation)

Color Variation

Random Color Scale

(Texture Based Color Variation)

TBCV Map Intenisty

TBCV Wold Scale

TBCV Map Offset

TBCV On (Uses Noise Map) ☒

(Random Contrast)

Random Contrast

(Texture Based Random Contrast)

TBRC Map Intenisty

TBRC Wold Scale

TBRC Map Offset

TBRC On (Uses Noise Map) ☒

(Season Controll)

Dry Leaf Color

Dry Leaves - Scale

Dry Leaves - Offset

Season Vertex Color (R) ☒

Branch Mask *(R) ☒

(Snow)

Snow Scale

Snow Offset

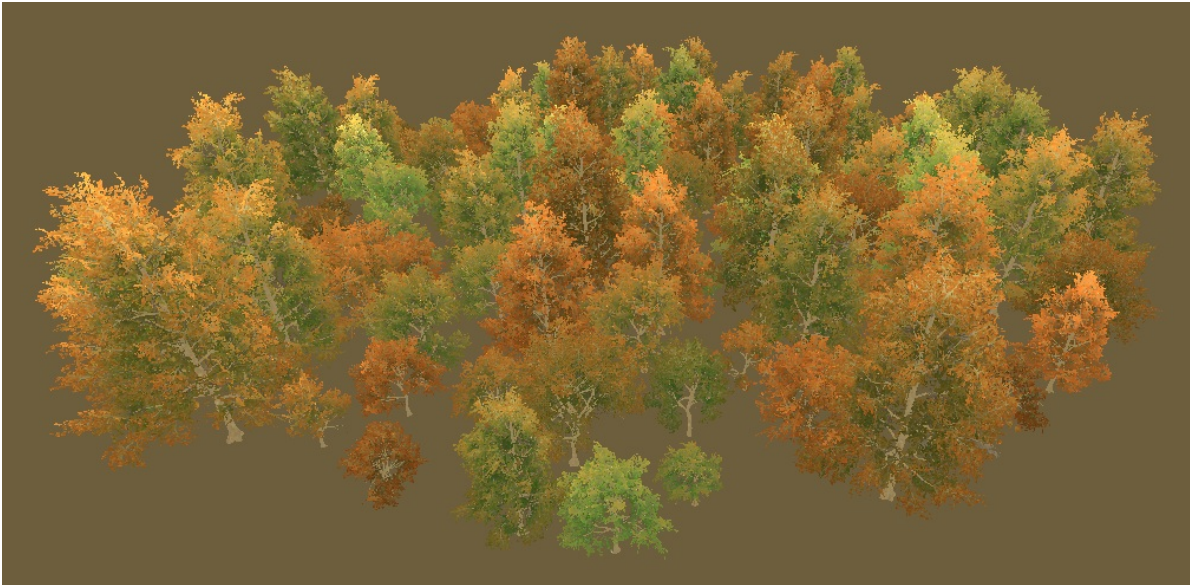
Normal Map Based Snow ☒

Snow Mask Vertex (R) ☐

Snow Value

- **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 via a static switch).
- **Texture Based Color Variant and Random Contrast (TBCV and TBRC)**
Uses a "noise map" texture to create the randomised color effects on a grid, rather than relying on world and object positions. This method is recommended for Pivot-based physics interactions because the object position causes an unwanted color change when the pivot is moved.
- **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.

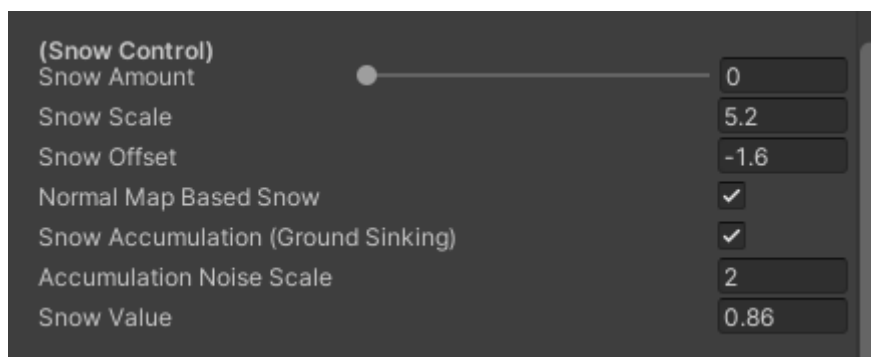
To lock the season color so it can't be changed with the (TTFE) GLOBAL CONTROLLER, set "Dry Leaves - Scale and Offset" to "0,0" (default values are "1,1"). Is useful for having dry or different-colored trees in any season.

- **Season 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).
- **Season Change - Global** (*Global variable, not visible in material inspector*)
Adjust the intensity of the seasonal change. Correlates directly to the "season" slider of the global gizmo.

- **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 including terrain.



- **Snow Amount – Global** (*Global variable, not visible in material inspector*)
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

WIND SETTINGS

(Wind Texture Maps)

 Wind Noise Map

(Pivot)

Grass Sway Power (Gentle Wind)1

Motion Bending Randomness (Gentle Wind)0.1

Motion Bending Downward Strength-0.5

Pivot Sway☒

Pivot Sway Power2

Wind TypeStrongW▼

(Noise)

Motion Flutter Intensity3

Motion Flutter Scale10

Quad Scatter Intensity1.4

Quad Scatter (Only basic cards)☒

Quad Scatter World Normals☐

Wind Motion Scale Mask☒

Grass Interaction☒

Physics Interaction☒

Physics Wind☒

(Perspective)

Perspective Intensity0.8

Perspective Correction☒

(TTFE) Grass Foliage

WIND SETTINGS

(Wind Texture Maps)

 Wind Noise Map

(Pivot)

Grass Sway Power (Gentle Wind)1

(Trunk and Branch)

Pivot Randomness Strength0.5

Pivot Randomness1

Wind TypeStrongW▼

Branch Wind Large1.5

Branch Wind Small1.5

Branch Sway Power4

Branch Fold Strength0.5

(Leaf Noise)

Motion Flutter Intensity1

Large Noise Intensity2

Flutter Size50

Flutter Power3

Flutter Tangent Mode☒

Quad Scatter Intensity2

Quad Scatter Scale10

Quad Scatter (Only basic cards)☒

Quad Scatter World Normals☐

Motion Bending Gentle Random0.1

Downward Strength-0.5

Leaf Interaction☒

Physics Interaction☒

Physics Wind☐

(TTFE) Tree Foliage

(Appendages)

Appendage ID Intensity0

Appendage ID Wind Direction☐

Appendage ID Random Direction☐

Appendage Motion Intensity0

Appendage Motion (Y) Intensity0

Appendages☐

(TTFE) Tree Bark

- **Wind Noise Map**

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.

- **Pivot**

Extra pivot motion settings for the grass shader. Motion bending works with the global controller motion vector. The direction vector can be rotated.

- **Trunk and Branch**

Adjust and customize branch and leaf settings.

- **Pivot settings**

Adjust the strength and randomness of the swaying pivot.

- **Branch Wind Large and Small**

Adjust the large and small wind noise of your branches. You should adjust the intensity of branch bending based on your preference, as sometimes the bending may not be very visible.

- **Bench Folding**

Inward fold of branches. Works with the wind vector and can be rotated using the wind Gizmo.

- **Leaf Noise**

Turns on leaf fluttering noise.

- **Flutter Tangent Mode**

Leaf flutter is based on vertex tangent rather than vertex position.

- **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.

- **Player Interaction**

Settings for player interaction contain 2 elements. Leaf Interaction (Shader) and Physics Interaction (Pivot bending).

- **Leaf Interaction**

Shader interaction with touch bedding animation. Works with any prefab as well as with terrain plants. It can be switched between multiple players. Supports only one interactor at a time.

- **Physics Interaction**

Pivot bending, works with an unlimited number of interactors at the same time. Doesn't work on terrain objects, as the terrain has its own collision system that is different from the default mesh. Use the "Terrain Plants to Prefab Converter" script as a workaround.

- **Physics Wind**

Extra animation added by wind physics, usually used on grass.

- **Cloth**

Similar to the foliage shader, it uses vertex animation instead of actual physics, but includes additional settings for cloth simulation.

- **Appendages**

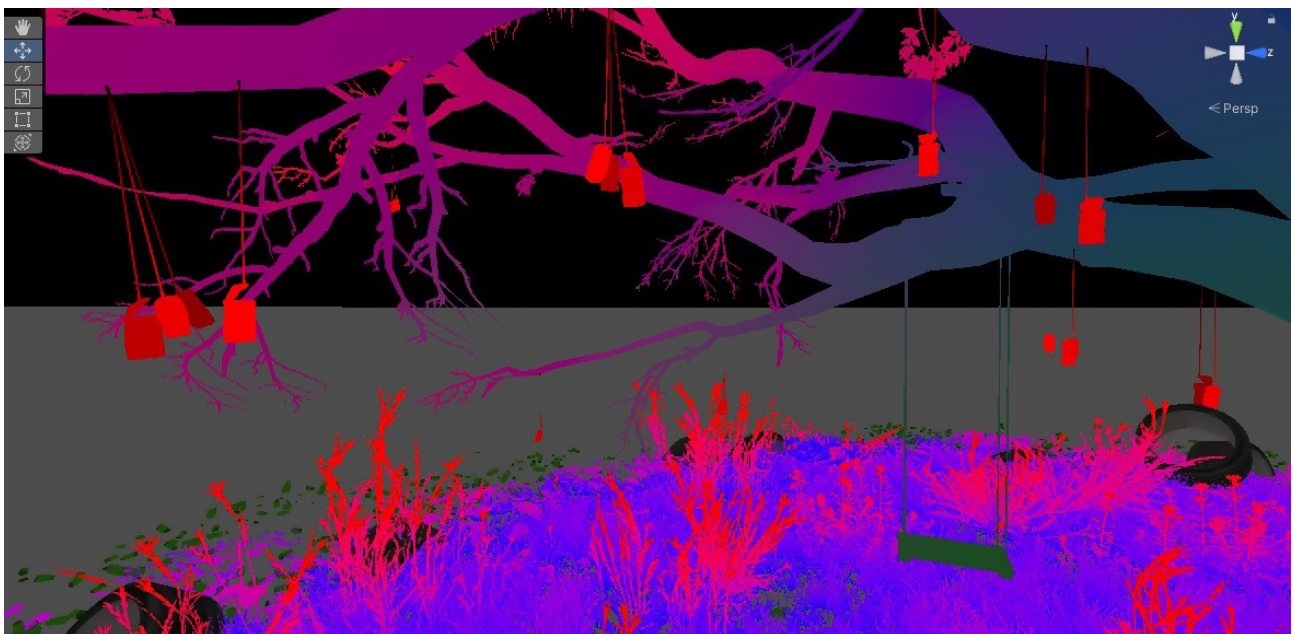
Enables attachment of hanging objects to main bodies or pivots for dynamic effects.

- In your 3D software:

1. Paint black where they connect to the tree.
2. Paint the objects with vertex color red.
3. Add a random value to the vertex colors for wind variation (works on a single merged mesh).

- In Unity:

1. Apply the Bark shader to the appendage objects.
2. Enable the Appendages feature in the material.
3. Copy the settings from the example mesh if needed.

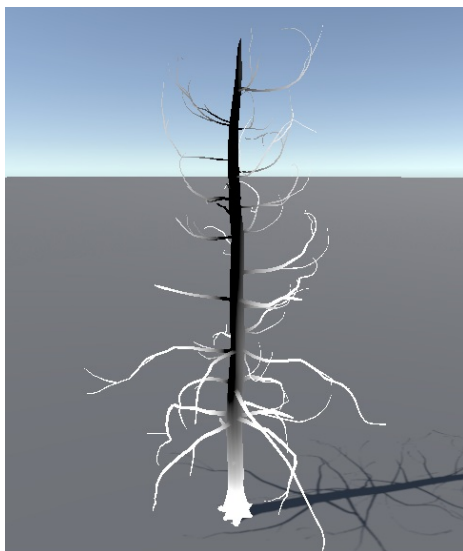
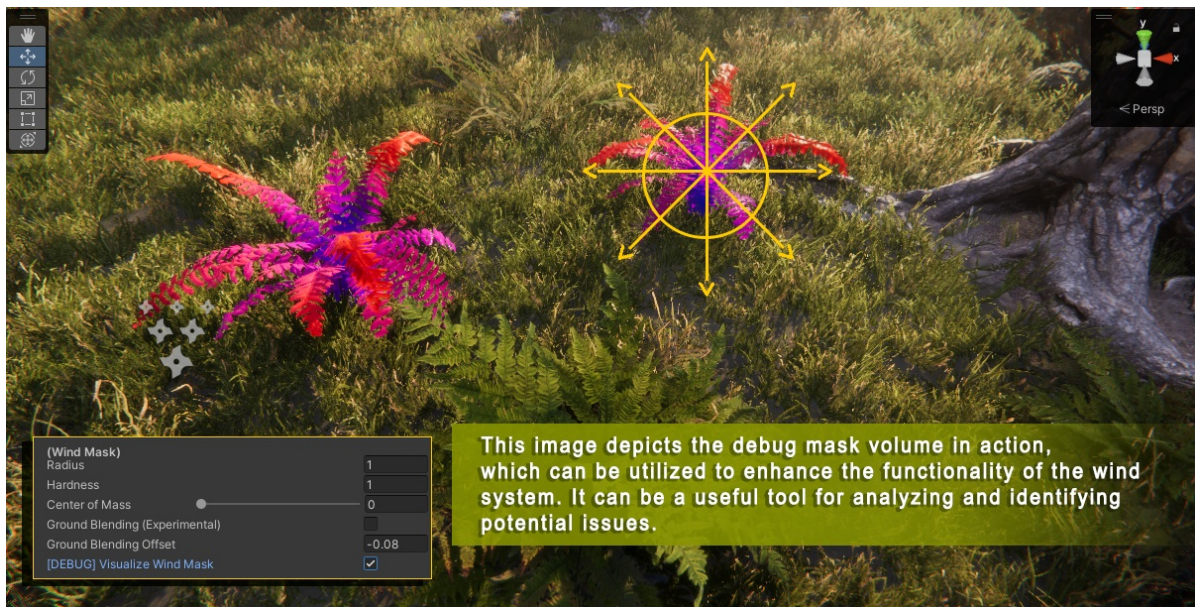


- **Perspective Correction**

Adjusts grass angles for optimal viewing from top-down perspectives.

Wind Mask

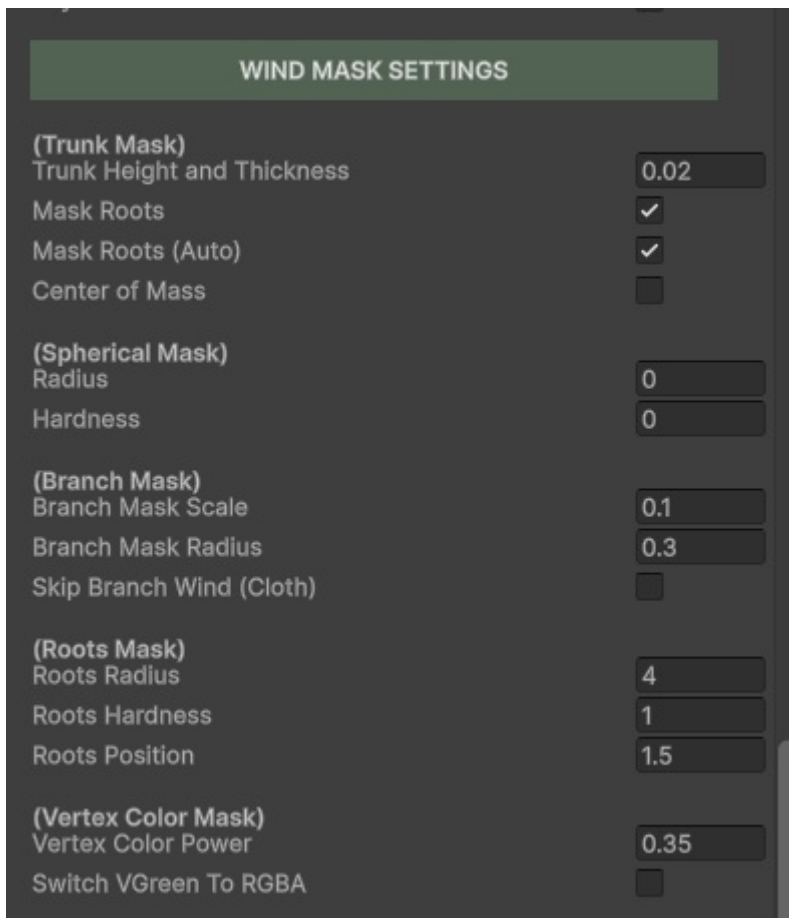
The meat and potatoes of the Toby Foliage Engine, the wind mask is essential for customizing how your trees behave in different wind conditions. A vertex-position volume sphere, masking out the wind on your trunk and leaves. Its radius can be controlled via the material settings.



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

Wind Mask Settings

The Wind Mask has been completely updated since version **2.0** and now includes four elements: **Trunk Mask**, **Spherical Mask**, **Branch Mask**, and **Root Mask**. Each mask layer is designed for a specific tree part and shape.



- **Trunk Mask**

Used for the trunk and basic pivot masking.

- **Trunk Height and Thickness**

The taller the trunk, the stronger the swaying effect will be. The wider the trunk, the less it will sway.

- **Mask Roots**

It prevents the roots from being affected by the wind.

- **Center of Mass**

Masking from the center simulates a center of gravity, reducing wind impact on the mesh.

- **Spherical Mask**

Creates a spherical volume from the center outwards. This technique is typically used for the central part of bark when you have an unconditional bark shape, which is usually bent.

- **Radius**

Adjusts the volume size to mask wind in your material settings.

- **Hardness**
Adjusts the intensity of the volume mask.

- **Branch Mask**

Masks only the branches.

- **Skip Branch Wind (Cloth)**
Used on flags or cloth that are hung by a pole.

- **Roots Mask**

Masks the lower part of the trunk, preventing it from disconnecting from the ground. The position of the mask volume can be adjusted along the Y axis (use Roots Position).

- **Vertex Color Mask**

Controls the intensity of the vertex color "green" on leaves. This will affect how your flutter noise and card bending will look.

Debug Wind Mask

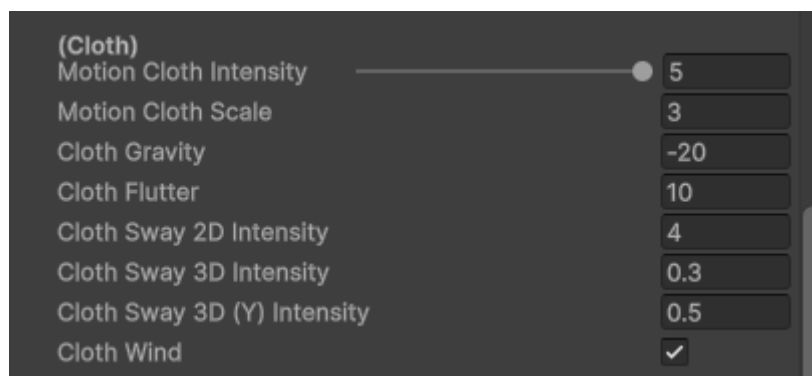
Visualize the wind mask volume using a color spectrum from red to blue. Red indicates strong wind, while blue signifies no wind. Green is used to represent the roots masking.

- **[DEBUG] Visualise Wind Mask**
Tick "[DEBUG] Visualize Wind Mask" to visualize wind physics mask.
- **[DEBUG] Compute Vertex Colors**
You can view the wind mask on all plants in the scene at once by clicking the toggle "Visualize Wind Mask" in the Global Controller. However, vertex colors can only be displayed by toggling the material option (compute vertex colors). This option only works per material.

Other Shaders

(TTFE) Cloth

Specifically designed for cloth and flags, this shader is similar to the vegetation shaders. It utilizes vertex animation enhanced with additional options for simulating cloth movement in the wind.



(TTFE) Mesh

Basic and performant PBR shader with crossfade. Also supports snow and wetness. Used on static meshes.

(TTFE) Tree Billboard

Identical to the tree foliage shader but with fewer features.

(TTFE) Amplify Impostor

3D/PBR billboard shader, it features custom impostor shaders that utilize the default texture packing used by TTFE. Supports all the season effects and works with the Global Controller.

(TTFE) Roots

Basic root shader. Can be used for dead leaves and ground cover.

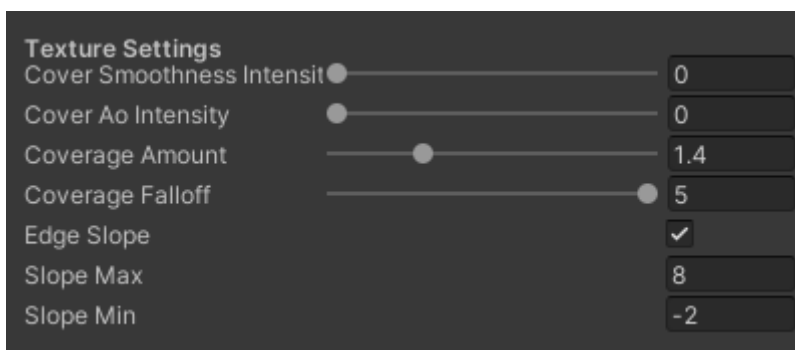
(TTFE) Simple Cliff Masked

Default cliff shader that is more optimized than the Uber cliff shader. It supports snow, wetness, and specular editing. Additionally, HDRP provides support for specular fade.

(TTFE) Uber Cliff (Legacy)

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) Water

Full PRB shader, the settings include various options for refraction and reflection, support for mirror reflections, and real-time reflection probes.

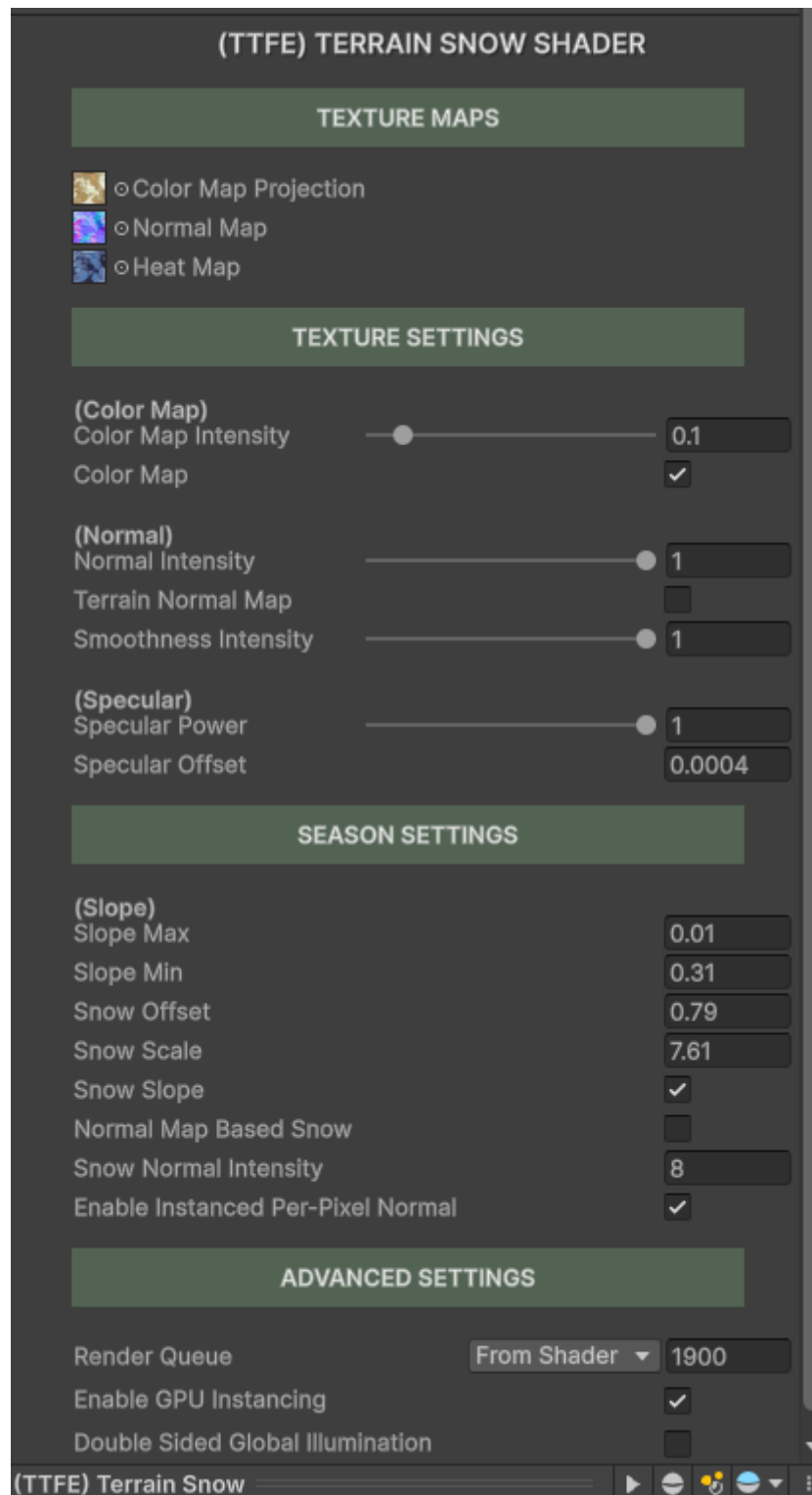
(TTFE) Terrain Snow

Enhanced terrain shader with support for snow slopes and wetness, similar to the mesh and cliff shaders. Includes additional maps like a color map, custom normal map, and heat map. Enable "**specular fade**" support in HDRP for realistic Fresnel.



Terrain wetness

Terrain with snow



Additional Features

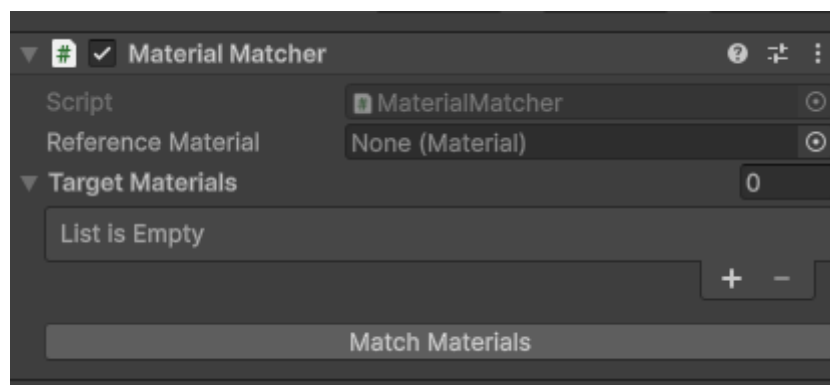
"TTFE includes a set of extra tools designed to streamline your workflow:

- **Material Matcher** – Aligns material properties across assets for perfect consistency.
- **Terrain Plants to Prefab Converter** – Non-destructively converts terrain plants into prefabs.
- HDRP **Specular Fade** support.

Material Matcher

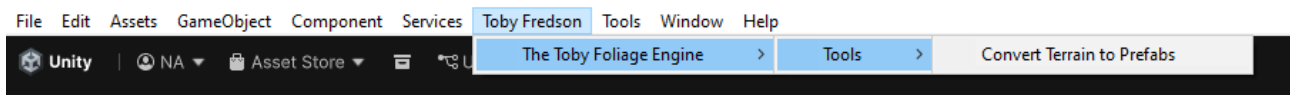
Create an empty game object and attach the "Material Matcher" script to it. Copy the source material into the "reference material" slot and add as many target materials as needed. Click the "Match Material" button.

For best results, use this tool on projects with many trees — particularly those that use separate materials for the bark and leaves, with the Tree Foliage shader on the leaves and the Trunk shader on the bark. You can avoid this by adding only one shader and material for the entire tree.



Terrain Plants to Prefab Converter

Because the physics/pivot interaction doesn't work on terrain objects, use the "Terrain Plants to Prefab Converter" script as a workaround. **Note** that the process cannot be canceled and may take a long time on large terrain as it uses CPU resources!



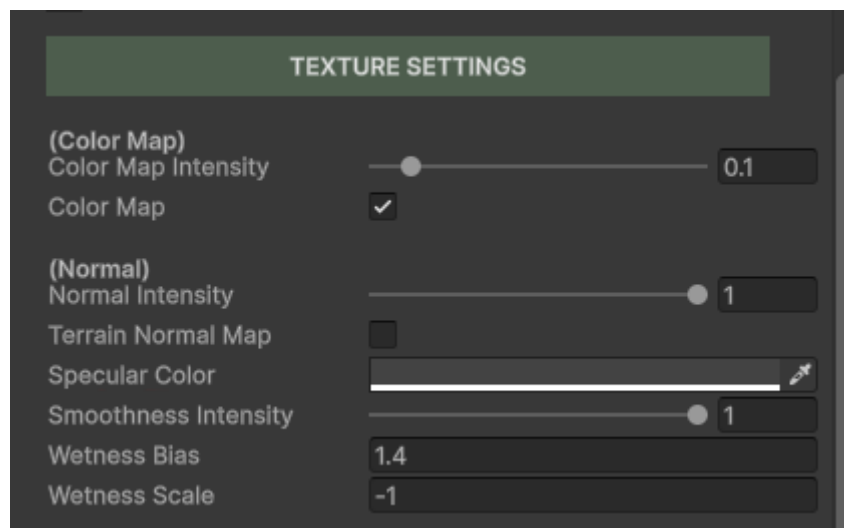
Specular Fade

Specular fade allows fine-tuning of the Fresnel effect using color properties or a texture map. This option can help eliminate the harsh Fresnel appearance on terrain. The “(TTFE) Mesh” shader includes this capability as well.

To enable this property, open the Project Settings window, select **HDRP Global Settings** and go to **Miscellaneous > Specular Fade**.



To adjust the Fresnel, change the “Specular Color” value.



Optimization

Rest assured that 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 that may require optimization for mobile devices.

- **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 to be surprisingly costly, likely due to the "blend node". This option is best to be avoided if possible.
- **Ground Blending (Experimental)**
This is an experimental feature that 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.

TTFE runs very well without instancing, so manually turn on instancing on your materials, as it may be disabled.

There appears to be a bug that disrupts instancing when "cross-fading" is enabled. If you encounter issues, ensure that your prefab has a low triangle count and that "LOD Crossfading" is turned off; otherwise, it may interrupt 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.

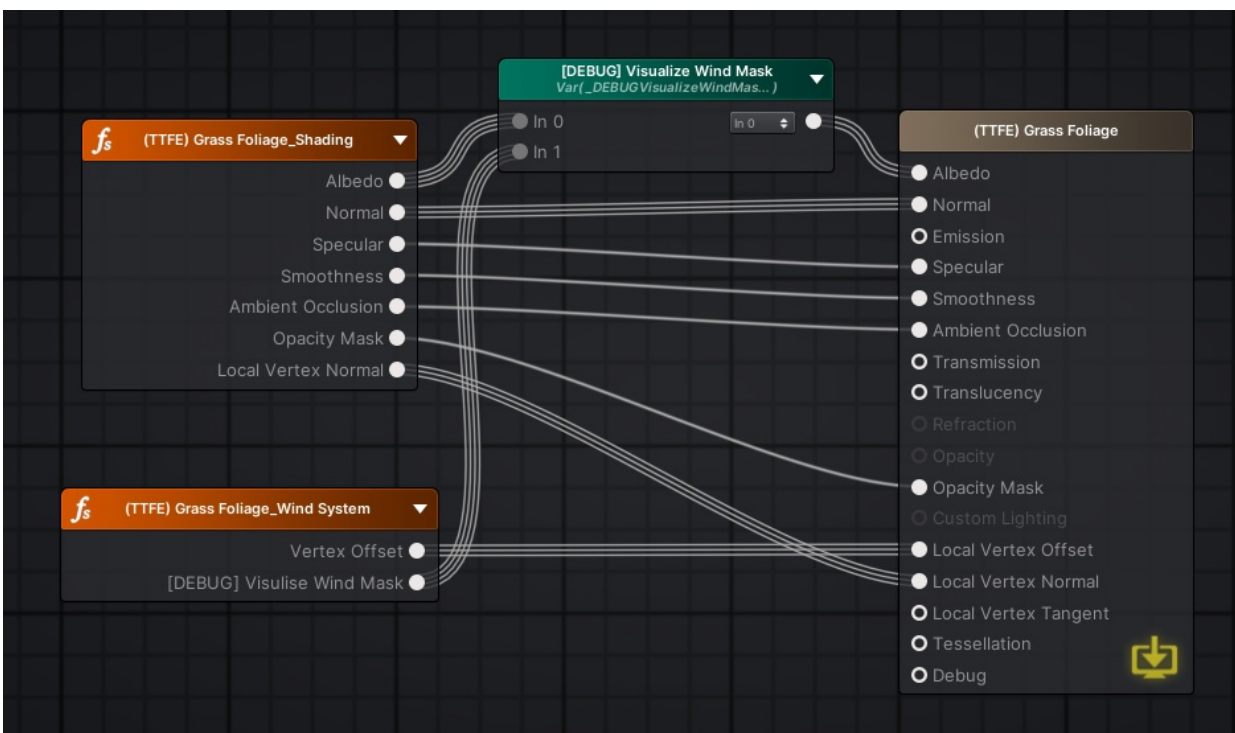
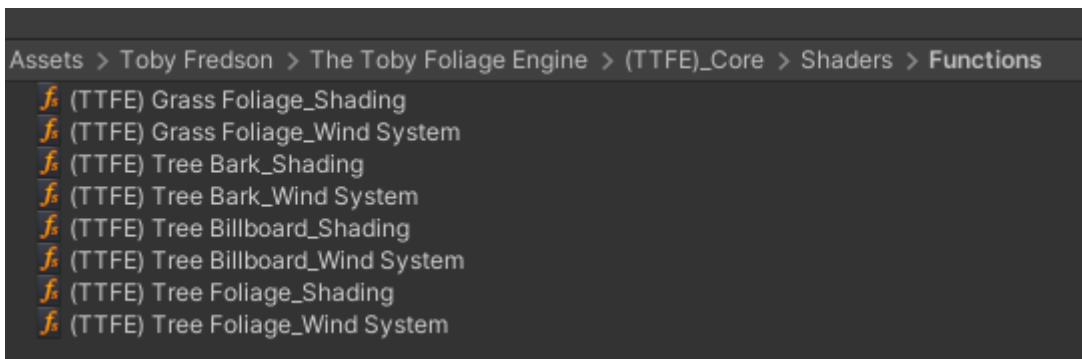
Mobile and Low-End Shader Variants

TTFE now includes mobile-friendly shaders. Simply switch to these variants to improve performance. The mobile version has eliminated unnecessary features, maintaining core functionality and reducing bloat. The low-end version includes only simple shading and lacks seasonal features, but it is compatible with the wind system.

Amplify Shader Functions

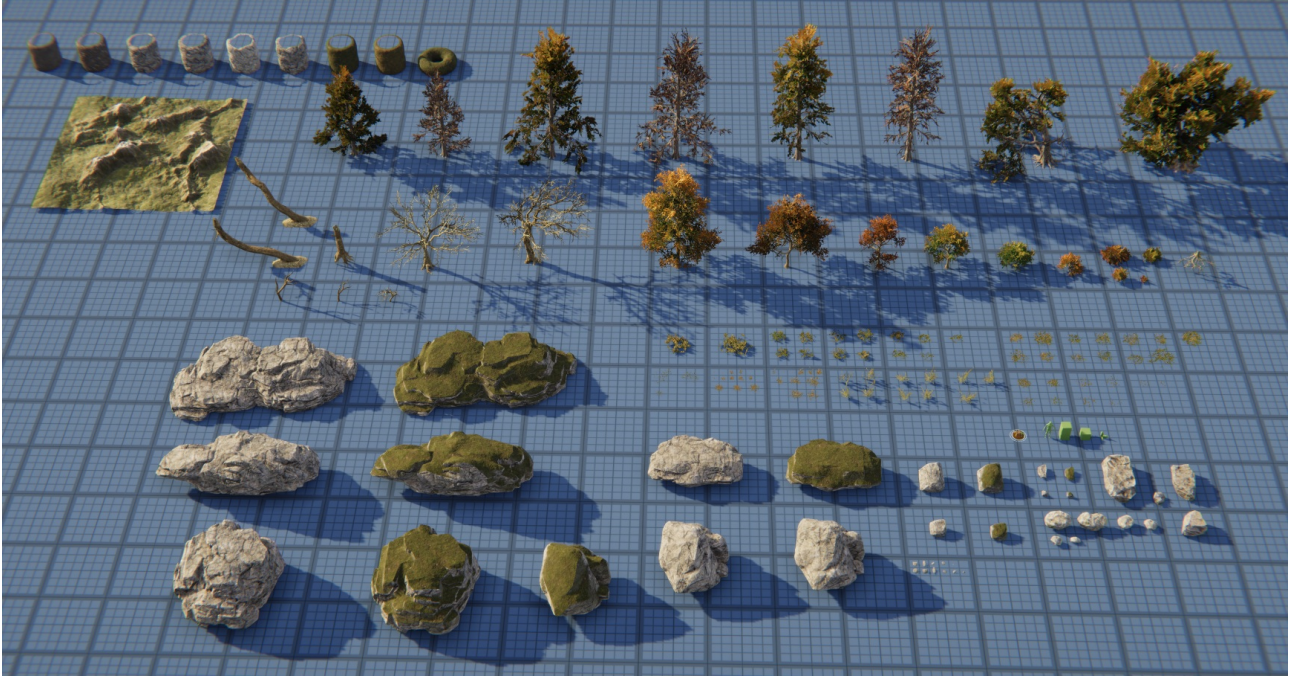
The node systems have been methodically arranged into Amplify Shader functions, rendering them easily accessible and reprehensible to integrate with your shaders. The system is well-organized and user-friendly, providing an efficient and effective workflow for users to optimize their performance.

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.



Release Notes

Version 2.0

- Complete package overhaul and optimization.
- Complete UI overhaul.
- Fixed Global Controller lag issues.
- Fixed season color bugs.
- Improved and more realistic wind.
- Additional options for wind and shading.
- Additional tools like “Material Matcher” and “Terrain Plants to Prefab Converter”.
- Additional nature prefabs.
- Additional learning example scenes.
- New grass and plant features.
- Optional shaders and fixes for Built-in.
- Fast and modern shader integration via global variables.
- Full Terrain Integration.
- Full mesh shaders integration with the TTFE controller.
- Player interaction with plants.
- Physics interaction with plants.
- Rotation of wind direction using the controller gizmo.
- Cloth physics features.
- Appendices (A useful feature for hanging objects attached to a main body or pivot).
- Perspective correction (Adjusts the angle of the grass as seen from above).
- Mobile and low-end shader variants.
- Wetness.

Version 2.1

- Amplify Impostor integration.
- Dual color.
- Specular Fade support.
- (TTFE) Water shader.
- Updated Global Controller.
- Added snow feature to (TTFE) Roots shader.
- Minor shader changes.
- Minor code updates.

Credits

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

https://www.artstation.com/ivan_janik

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

Acknowledgement

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