CTI Runtime Components URP 14plus

About this documentation

In case you want to use CTI trees along with the URP 14.0.4 or above you have to assign the *CTI URP SH* shader graph shaders and use the *CTI_SRP_CustomWind* script.
Both shaders and script are slightly different from the CTI Runtime Components for the built in RP.

Table of Content

Limitations

- Only basic LOD trees are supported. No tessellation.
- The shaders only accept wind from script.
- Shaders need URP 14+ Other versions of URP may need you to install the proper support package (if it exists).
- You can not author billboard textures using URP nor can you use the debug shader, so authoring should take place using the built in RP.

Changes

- Bark and Leaf shader only use two **textures**: The Albedo/Smoothness and Normal/Occlusion (bark) or the Albedo/Alpha and Normal/Smoothness/Translucency (leaves) textures. *→ You have to create new combined normal maps.*
- The shaders only accept **wind from script**. *→ The Tree component is not needed at all and should be removed.*

- The CTI LOD URP shaders need a slightly different input for the **wind from script**.
  → You have to use the CTI_SRP_CustomWind script instead of the old one.
- **Fade out Wind** has been dropped.
- **Fade out Translucency** has been dropped.
- **Tumbling** and **Turbulence** have slightly been reworked and optimized. *→ You may have to adjust their settings.*
- **Wind multipliers** for primary and secondary bending as well as edge fluttering have been added. *→ Now you can tweak the bending without editing the tree. Make sure multipliers in the bark material match those in the leaf material.*

CTI URP Shader Graphs

CTI URP SG Bark & CTI URP SG Bark Array shader

Shader Inputs

Base Inputs

Color Variation Trees will be slightly different tinted according to their position in world space. RGB defines the tint color, alpha the strength of the tint. *Always make sure that all shaders (leaves, bark and billboard) share the same color variation values.*

Albedo(RGB) Smoothness(A) Diffuse texture which contains **smoothness** (unlike the leaf shader which expects transparency) in the alpha channel.
The array shader here expects a texture array.

Smoothness Multiplier for the smoothness as sampled from the *Albedo (RGB) Smoothness(A)* map.

Normal Map(GA) Occlusion(B) contains the combined normal and occlusion map. *Red color channel should be set to pure black.*
The array shader here expects a texture array.

Normal Scale Lets you scale the normal

Specular Specular Color

Base Wind

Wind

Main(X) Multiplier for the Primary Strength. *Must match the value in the leaf material.*

Branch(Y) Multiplier for the Secondary Strength. *Must match the value in the leaf material.*

Flutter(Z) Multiplier for Edge Flutter. *Does not matter here.*

CTI URP SG Leaves shader

Shader inputs

Base Inputs

Color Variation Trees will be slightly different tinted according to their position in world space. RGB defines the tint color, alpha the strength of the tint. *Always make sure that all shaders (leaves, bark and billboard) share the same color variation values.*

Albedo(RGB) Alpha(A) Diffuse texture which contains transparency in the alpha channel.

Alpha Cutoff If the alpha channel of the Base texture contains different shades of gray instead of just black and white, you can manually determine the cutoff point by adjusting the slider.

Normal(GA) Smoothness(B) Trans(R) contains the combined normal, smoothness and translucency map.

Smoothness Multiplier for the smoothness as sampled from the *Normal (GA) Smoothness (B) Trans (R)* map. If this map is disabled *Smoothness* defines

Enable Normal Rotation if checked tumbling and turbulence will affect the normals.

Leaf Noise lets you adjust the time offset added when calculating leaf turbulence and tumbling based on the baked *edge flutter* (stored in vertex color green). Using values above 0.0 will most likely add some distortion to the leaf meshes – which in fact looks quite nice.

Enable advanced Edge Flutter Checking this will add a low frequency edge fluttering driven by vertex color blue and green and:

Strength(X) and Frequency(Y)

Transmission

Strength(X) acts as a factor which gets multiplied with the translucency value sampled from the "AO (G) Translucency (b) Smoothness (A)" map and lets you fine adjust final translucency.

Power(Y) or *View Dependency* determines when the translucent lighting effect will kick in depending on the view angle: Lower values will make translucent lighting appear already at rather flat viewing angles while high values will make it appear only if you look directly towards the sun.

Distortion(Z) Lets you distort the normal used to calculate the transmission. A value of 0.01 - 0.1 should be fine.

Ambient Scattering Lets you add backface ambient lighting. If set to 0.0 the shader will skip the ambient lighting evaluation.

CTI URP SG Billboard shader

Shader inputs

Blend Billboards If checked the shader will blend between two samples from the billboard atlas which gives you a nicer blending at slightly higher costs.

The alpha texture needs some noise in it to make this work which on the other hand means that we will lose the ability to add some ambient occlusion to the alpha as well. So set "Alpha Leak Suppression" to 1.0.

Enable Dithering If checked blending does not need any noise in the alpha. Instead a dither pattern will be calculated within the pixel shader. This results in a less organic blending but allows you to keep occlusion in the alpha.

Base Inputs

Color Variation(RGB) Strength(A) Make sure that the color fits the one you have added to the mesh trees.

Albedo(RGB) Alpha/Occlusion(A) This slot should contain the created albedo texture atlas.

Alpha Cutoff If the alpha channel of the Base texture contains different shades of gray instead of just black and white, you can manually determine the cutoff point by adjusting the slider. A value of 0.45 should just be fine.

Alpha Leak Suppression: As the alpha channel of the Albedo textures may store both: *Alpha* and *Occlusion* dark pixels from the alpha mask might leak into the occlusion texture (caused by bilinear filtering and mip mapping) which would end up in full occlusion at the outer parts of the billboard. But if you set it about 0.6 all pixels darker than that will be set to white so you will get simply no occlusion on outer pixels – which in fact makes much more sense.

Normal(AG) Translucency(R) Smoothness(B) This slot should contain the created texture atlas.

Normal Scale Scale of the normal.

Smoothness Multiplier for the smoothness as sampled from the *Normal (AG) Translucency (R) Smoothness (B)* map. If this map is disabled *Smoothness* defines the final smoothness value.

Thickness Remap Lets you remap the thickness

Specular Specular Color as a simple solid color which you most likely should set to the default value of dielectric materials which is RGB = 51,51,51.

Wind

Wind Strength As Billboards do not have any baked wind information you may use this parameter to make the bending of the billboard better match the bending of the mesh tree.

Wind Power Power value which drives the wind strength along the y axis. Should match the power value used on importing the tree. Default is 1.5.

Transmission

Strength(X) acts as a factor which gets multiplied with the translucency value sampled from the “AO (G) Translucency (b) Smoothness (A)” map and lets you fine adjust final translucency.

Power(Y) or *View Dependency* determines when the translucent lighting effect will kick in depending on the view angle: Lower values will make translucent lighting appear already at rather flat viewing angles while high values will make it appear only if you look directly towards the sun.

Distortion(Z) Lets you distort the normal used to calculate the transmission. A value of 0.01 - 0.1 should be fine.