Smooth Style Readme

So so smooth!

Overview

Online Version

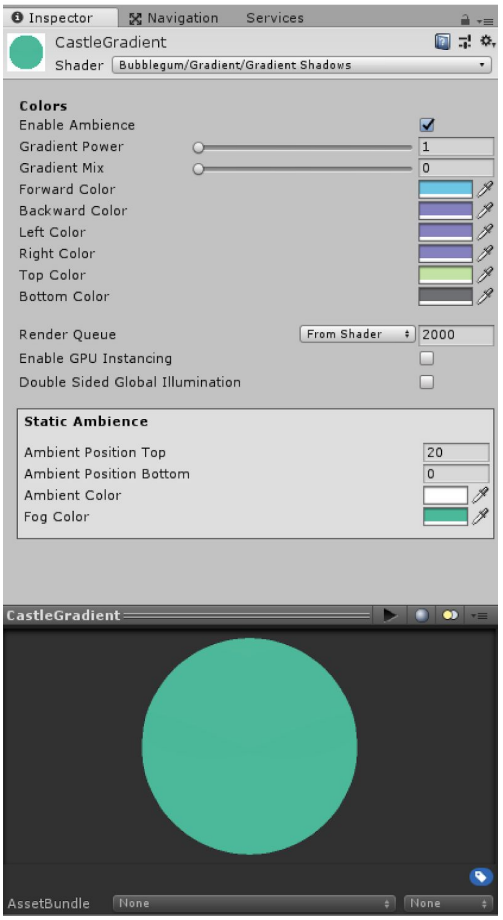
This package is designed as a solution for a good looking high performance game. The shaders are written in a way that makes them expandable and can be easily tweaked for even better performance. This document is designed in such a way to teach a beginner programmer how they might edit the shaders for themselves.

Support

Please email info@turbokiwi.com for all support queries. We will aim to respond within 24 hours.

Quick Start

The package contains four sample scenes that demonstrate how to work with the assets. Otherwise follow these quick guidelines:



1. Create a material
2. Select the shader for the material: **Bubblegum** > **Gradients** > **Gradient Simple**
3. Input colors for each of the sides (these are world space values)
4. Use the **Gradient Power** and **Gradient Blend** sliders to achieve the desired look
5. Toggle **Ambience** on/off. The static values are shared between all materials (see ambience workflow for more).
6. Drag the new material onto your 3D object.

Shader Inner Workings

The gradient is applied through the vertex pass, this means that we will only get color blending on round/curved objects.

Vertex Pass

The directional colors are applied during the vertex pass. There are two options for blending, **Multiply** or **Additive.** Additive being the quicker of the two, but does not include all of the features such as controlling the mix amount.

Fragment Pass

The ambience is applied during the fragment pass so that we can have a smooth vertical fog gradient. Removing this pass will improve performance (remember to remove from both lightmapped and normal passes).

Enabled fragment pass:

```
118         fixed4 frag(vertexOutput o) : COLOR
119         {
120             fixed4 mainColor = tex2D(_MainTexture, o.uv0) * ComputeFragmentGradient(o.color);
121             mainColor.rgb *= DecodeLightmap(UNITY_SAMPLE_TEX2D(unity_Lightmap, o.uv1));
122             UNITY_APPLY_FOG(o.fogCoord, mainColor);
123 
124             return mainColor;
125         }
126 
127         ENDCG
```

Disabled fragment pass:

```
118         fixed4 frag(vertexOutput o) : COLOR
119         {
120             fixed4 mainColor = tex2D(_MainTexture, o.uv0);// *ComputeFragmentGradient(o.color);
121             mainColor.rgb *= DecodeLightmap(UNITY_SAMPLE_TEX2D(unity_Lightmap, o.uv1));
122             UNITY_APPLY_FOG(o.fogCoord, mainColor);
123 
124             return mainColor;
125         }
```

Ambience Workflow

Basic Control

Control the values via the inspector on the material, please note that changing the values on one material will affect them all.

Dynamic Control

Looking into the scene **GradientCastle,** select the **Sun** object. There are four **ShaderPropertyEditing** components attached. These are used to set the static ambience values when the scene is loaded. This workflow is required to have different values in each scene.

Lightmapping

The shaders also include a pass written for lightmapped objects. This exception is the **GradientShadows** shader, which is written to support realtime shadows, the lightmapped equivalent is the **GradientSimple** shader.

Shader Variants

Gradients.cg includes all of the shared functions and variables between shaders

Gradient Fast

Additive based coloring shader which ignores the ambience and mixing control.

Gradient Simple

Very basic coloring shader, this acts as the base for the below shaders.

Gradient Shadows

Using the simple shader but with an added lighting pass to receive shadows, this is therefore slower than the simple shader.

Gradient Textured

Using the simple shader but with an added texture pass, this can be used with a color map to batch draw calls.

Gradient Skybox

A generic gradient skybox.

Post Processing

The package screenshots included a heavy post processing stack, this has been left out of the package but can easily be re-implemented if desired.

Just copy the toggle boxes in this image and you can achieve the same look.

Bubblegum

The Bubblegum framework is a flexible framework designed for Unity3D. The shaders in this package are included in Bubblegum for free. So are they not free on the asset store? The small fee covers basic support and request, and the shaders are not guaranteed to always be included with Bubblegum. https://bitbucket.org/alexisrabadan/bubblegum

Credits

Castle Model by Shrugz http://hiddencreative.org
Shader based on Minimalist https://www.assetstore.unity3d.com/en/#!/content/91366

THE END