How To Add Vertex Colours Or A Second UV Channel

BigWorld Technology 2.1. Released 2012.

Software designed and built in Australia by BigWorld.

Level 2, Wentworth Park Grandstand, Wattle St
Glebe NSW 2037, Australia
www.bigworldtech.com

Table of Contents

Chapter 1. Introduction

BigWorld has a standard set of vertex formats used in the 3D engine to store the required information for rendering. This data storage covers the majority of usage cases, but some times more information is needed. This document describes a way of utlilising the new vertex colour and uv coordinate data stream to export vertex colours or extra texture coordinate information from 3dsmax or Maya and then load and use this data in a custom shader.

Chapter 2. Exporting vertex data

2.1. Setup

To set up BigWorld to use the extra vertex streams, make sure your chosen exporter (max/maya) is installed properly and your export pipeline is functioning (exporting a model from Max / Maya and loading it in Model Editor).

2.2. Authoring the content

The actual authoring of the extra vertex data is different depending on which authoring tool being used. Just ensure that only one extra vertex colour/uv set is created for each model. The extra vertex data will automatically be detected and appended to the model when it is exported.

Chapter 3. Using the vertex data in a shader

3.1. Using Vertex Colour

Once you have a model exported with the extra vertex colour data, you will need to make a few shader modifications to use it. The most important part is the modification of the vertex shader input structures to include the new parameter. The vertex colour is bound as an extra stream of vertex data. This stream is linked to the `COLOR0` input semantic. So to use the vertex colour in you vertex shader, the input should include something like:

```
float4 colour: COLOR;
```

The `VertexXYZNDUV` structure is used in the example shader. Once you have access to this data inside the vertex shader, you can do whatever use it however you want. The example shader uses the vertex colour to tint the result of the regular lightonly shader.

You can refer to the `lightonly_vertexcolour.fx` file for an example of using the new vertex colour stream.

3.2. Using A Second UV Channel

Using a second UV channel is very similar to way the vertex colour is used (described above). The new vertex UV coordinate is bound as an extra stream of vertex data and is linked to the `TEXCOORD1` input semantic. So in order to access this data, the vertex input structure should include:

```
float4 uv2: TEXCOORD1;
```

The example shader uses the extra texture coordinates to sample a second texture and combine it with another texture (sampled with the original coordinates) and combined via the alpha channel of the second texture.

You can refer to the `lightonly_dual.fx` file for an example of using the new secondary vertex uv stream.