# 6.4 顶点着色器示例

下面是一个顶点着色器的示例，它的代码非常简单：

cbuffer cbPerObject

{

 float4x4 gWVP;

};

void VS(float3 iPosL : POSITION,

 float4 iColor : COLOR,

 out float4 oPosH : SV\_POSITION,

 out float4 oColor : COLOR)

{

 // 转换到齐次裁剪空间

 oPosH = mul(float4(iPosL, 1.0f), gWVP);

 // 把顶点颜色直接传到像素着色器

 oColor = iColor;

}

着色器使用一种称为高级着色语言（High-Level Shading Language，简称HLSL）的脚本语言来编写，它的语法与C++相似，很容易就能学会。附录B提供了一些有关HLSL的简要概述。在本书中，我们将采用一种基于示例的方式讲解HLSL及着色器编程。也就是，根据贯穿本书的每个演示程序所涉及的技术讲解相关的HLSL概念。着色器通常保存在一种称为effect文件（.fx）的纯文本文件中。我们会在本章随后的小节中讨论effect文件，而现在我们主要讨论顶点着色器。

这里，顶点着色器是一个称为VS的函数。注意，你可以为顶点着色器指定任何有效的函数名。该顶点着色器包含4个参数；前两个是输入参数，后两个是输出参数（由**out**关键字表示）。HLSL没有类似于C++的引用和指针，所以当一个函数要返回多个值时，我们必须使用结构体或输出参数。

前两个输入参数对应于我们在顶点结构体中定义的数据成员。参数语义“**:POSITION**”和“**:COLOR**”用于将顶点结构体的数据成员映射为顶点着色器的输入参数，如图6.4所示。

****

**图6.4 D3D11\_INPUT\_ELEMENT\_DESC数组为每个顶点元素指定了一个相关的语义，而顶点着色器的每个参数也都带有一个附加语义。语义用于建立顶点元素和顶点着色器参数之间的对应关系。**

输出参数也带有附加语义（“**:SV\_POSITION**”和“**:COLOR**”）。这些语义用于将顶点着色器的输出数据映射为下一阶段（几何着色器或像素着色器）的输入数据。注意，**SV\_POSITION**是一个特殊的语义（SV表示系统值，即system value的缩写）。它用于告诉顶点着色器该元素存储的是顶点位置。顶点位置的处理方式与其他顶点属性不同，因为它涉及到一些其他属性所没有的特殊运算，比如裁剪。若不是系统值，那么输出参数的语义名称可以是任何有效的语义名称。

该顶点着色器的代码非常简单。第一行通过与一个4×4矩阵**gWorldViewProj**相乘，将顶点位置从局部空间变换到齐次裁剪空间，矩阵**gWorldViewProj**是世界矩阵、观察矩阵和投影矩阵的组合矩阵：

// 转换到齐次裁剪空间

oPosH = mul(float4(iPosL, 1.0f), gWorldViewProj);

构造函数语法“float4(iPosL, 1.0f)”用于创建4D向量，它相当于“float4(iPosL.x, iPosL.y, iPosL.z, 1.0f)”；我们知道，顶点位置是一个点而不是一个向量，所以第4个分量应设为1（即w=1）。**float2**和**float3**分别表示2D和3D向量。矩阵变量**gWorldViewProj**定义在一个常量缓冲区中，我们会在下一节讨论对它进行讨论。内置函数**mul**用于实现向量-矩阵乘法，它为不同维数的矩阵乘法定义了多个重载版本；例如，该函数可以实现4×4矩阵乘法、3×3矩阵乘法、或者1×3向量与3×3矩阵的向量-矩阵乘法。最后一行是将输入的颜色赋值给输出参数，把颜色传递给管线的下一阶段：

oColor = iColor;

我们可以使用结构体来重写上面的顶点着色器，实现相同的功能：

cbuffer cbPerObject

{

 float4x4 gWVP;

};

struct VS\_IN

{

 float3 posL : POSITION;

 float4 color : COLOR;

};

struct VS\_OUT

{

 float4 posH : SV\_POSITION;

 float4 color : COLOR;

};

VS\_OUT VS(VS\_IN input)

{

 VS\_OUT output;

 output.posH = mul(float4(input.posL, 1.0f), gWVP);

 output.color = input.color;

 return output;

}

**注意**：当没有几何着色器时，顶点着色器至少要实现投影变换，因为当顶点离开顶点着色器时（在没有几何着色器的情况下），硬件假定顶点位于投影空间。当包含一个几何着色器时，投影工作可以转嫁到几何着色器中完成。

**注意**：顶点着色器（或几何着色器）不执行透视除法；它只完成投影矩阵部分。透视除法会随后由硬件完成。