# 6.7 渲染状态

从本质上讲，Direct3D是一个状态机（state machine）。在我们改变它的状态之前，驻留在状态机内的当前状态是不会改变的。例如，我们在6.1节、6.2节和6.3节中看到，当顶点缓冲和索引缓冲绑定到管线的输入装配阶段时，如果我们不绑定其他缓冲，那么它们就会一直驻留在那里；同样，在没有改变图元拓扑之前，当前的图元拓扑设置会一直有效。另外，Direct3D将配置信息封装在状态组中，我们可以使用如下3种状态组配置Direct3D：

**1．ID3D11RasterizerState**：该接口表示用于配置管线光栅化阶段的状态组。

**2．ID3D11BlendState**：该接口表示用于配置混合操作的状态组。我们将在有关混合的章节讨论这些状态；默认情况下，混合处于禁用状态，所以我们可以先不考虑这方面的问题。

**3．ID3D11DepthStencilState**：该接口表示用于配置深度测试和模板测试的状态组。我们将在有关模板缓冲的章节讨论这些状态；默认情况下，模板是禁用的，所以我们可以先不考虑这方面的问题。而默认的深度测试是我们在4.1.5节描述的标准深度测试。

目前，我们唯一需要关心的状态块接口是**ID3D11RasterizerState。**我们可以通过填充一个**D3D11\_RASTERIZER\_DESC**结构体并调用如下方法来创建**ID3D11RasterizerState**对象：

HRESULT ID3D11Device::CreateRasterizerState(

const D3D11\_RASTERIZER\_DESC \*pRasterizerDesc,

ID3D11RasterizerState \*\*ppRasterizerState);

第1个参数是一个指向**D3D11\_RASTERIZER\_DESC**结构体的指针，该结构体用于描述所要创建的光栅化状态块；第二个参数用于返回创建后的**ID3D11RasterizerState**对象。

**D3D11\_RASTERIZER\_DESC**结构体的定义如下：

typedef struct D3D11\_RASTERIZER\_DESC{

D3D11\_FILL\_MODE FillMode; // Default:D3D11\_FILL\_SOLID

D3D11\_CULL\_MODE CullMode; // Default:D3D11\_CULL\_BACK

BOOL FrontCounterClockwise; // Default:false

INT DepthBias; // Default:0

FLOAT DepthBiasClamp; // Default:0.0f

FLOAT SlopeScaledDepthBias; // Default:0.0f

BOOL DepthClipEnable; // Default:true

BOOL ScissorEnable; // Default:false

BOOL MultisampleEnable; // Default:false

BOOL AntialiasedLineEnable; // Default:false

} D3D11\_RASTERIZER\_DESC;

这里面的大部分成员是高级选项或者不常用的选项；因此，我们在这里只讲解前3个成员的含义，其他成员的详情请参见SDK文档。

**1．FillMode**：当指定为**D3D11\_FILL\_WIREFRAME**时，表示以线框模式渲染几何体；当指定为**D3D11\_FILL\_SOLID**时，表示以实心模式渲染几何体，这是默认值。

**2．CullMode**：当指定为**D3D11\_CULL\_NONE**时，表示禁用背面消隐功能；当指定为D3D11\_CULL\_FRONT时，表示消隐朝前的三角形；当指定为**D3D11\_CULL\_BACK**时，表示消隐朝后的三角形，这是默认值。

**3．FrontCounterClockwise**：当设为**false**时，表示按顺时针方向环绕的三角形（相对于观察者）是朝前的，而按逆时针方向环绕的三角形（相对于观察者）是朝后的，这是默认值。当设为**true**时，表示按逆时针方向环绕的三角形（相对于观察者）是朝前的，而按顺时针方向环绕的三角形（相对于观察者）是朝后的。

在创建**ID3D11RasterizerState**对象之后，我们可以使用个新的状态块来更新设备：

void ID3D11DeviceContext::RSSetState(ID3D11RasterizerState \*pRasterizerState);

下面的代码示范了如何通过创建一个光栅化状态块来禁用背面消隐：

D3D11\_RASTERIZER\_DESC rsDesc;

ZeroMemory(&rsDesc, sizeof(D3D11\_RASTERIZER\_DESC));

rsDesc.FillMode = D3D11\_FILL\_SOLID;

rsDesc.CullMode = D3D11\_CULL\_NONE;

rsDesc.FrontCounterClockwise = false;

rsDesc.DepthClipEnable = true;

HR(md3dDevice->CreateRasterizerState(&rsDesc,&mNoCullRS));

**注意**：因为没有设置的属性的默认值是0或false，所以使用ZeroMemory可以正常初始化这些属性。但是，若有些属性默认值不是0或是true，那么你就必须显式地设置这些值。

注意，对于一个应用程序来说，你可能会用到多个不同的**ID3D11RasterizerState**对象。所以，你应该在初始化时把它们都创建出来，然后在应用程序的更新/绘图代码中切换些状态。例如，场景中有两个物体，你希望先以线框模式绘制第一个物体，然后再以实心模式绘制第二个物体。那么，你就应该创建两个**ID3D11RasterizerState**对象，当绘制物体时，切换这两种不同的状态：

// Create render state objects at initialization time.

ID3D11RasterizerState\* mWireframeRS;

ID3D11RasterizerState\* mSolidRS;

...

// Switch between the render state objects inthe draw function.

md3dDeviceContext->RSSetState(mSolidRS);

DrawObject();

md3dDeviceContext->RSSetState(mWireframeRS);

DrawObject();

注意，Direct3D不会从一种状态自动恢复到先前状态。所以，当绘制物体时，你应该根据需要手工指定状态对象。错误地假设设备的当前状态必然会导致错误的渲染结果。

每个状态块都有一个默认状态。我们可以通过在调用**RSSetState**方法时指定空值来恢复默认状态：

md3dDeviceContext->RSSetState( 0 );

**注意**：应用程序无需在运行时创建额外的渲染状态组。所以，应该在初始化时就定义并创建所有需要用到的状态组。而且，因为无需在运行时修改状态组，你可以在渲染代码中对这些状态组提供全局只读访问。例如，你可以将所有状态组对象放置在一个静态类中，通过这个方法，你就无需创建重复的状态组，渲染代码的不同部分都能共享这个渲染状态组对象。