**11 模块**

模块缓冲区是一种“离屏”缓冲区，我们可以利用它来实现一些特殊效果。

模块缓冲区，后台缓冲区以及深度缓冲区都有着相同的分辨率，这样一来这三者相同位置上的像素就能一一对应起来。

在指定一个模板缓冲区时，要将它与一个深度缓冲区配合使用。

这种缓冲区所起的作用就如同印刷过程中使用的模板一样，我们可以用它来组织特定的像素片段渲染至后台缓冲区中。

要设置模板缓冲区(以及深度缓冲区)状态，就需要填写D3D12\_DEPTH\_STENCIL\_DESC结构体实例，并将其赋予PSO的DepthStencilState字段。

**11.1 深度/模板缓冲区的格式及其资源数据的清理**

深度/模板缓冲区其实也是一种纹理，因而必须用下列特定的数据格式来创建它。

深度/模板缓冲可用的格式如下：

1,DXGI\_FORMAT\_D32\_FLOAT\_S8X24\_UINT(格式说明)

2,DXGI\_FORMAT\_D24\_UNORM\_S8\_UINT(格式说明)

我们可以在绘制每一帧画面之初，用以下方法来重置模板缓冲区中的局部数据：

ClearDepthStencilView

**11.2 模板测试**

我们可以通过模板缓冲区阻止对后台缓冲区特定区域的绘制行为，这项操作实则由模板测试来决定的。

模板测试会随着像素的光栅化过程而执行(即在输出合并阶段进行)。若模板功能呈开启状态，则需经过下面两处运算：

1，在左运算数由程序中定义的模板参考值StencilRef与程序内定义的掩码值StencilReadMask通过AND(与)运算来加以确定；

2，右运算数由正在接受模板测试的特定像素位于模板缓冲区中对应值Value与程序中定义的掩码值StencilReadMask经过AND(与)运算来加以确定；

可以发现左运算数和右运算数中的StencilReadMask是同一个值。接下来，模板测试用程序中所选定的比较函数对左运算数与右运算数进行比对，从而得到布尔类型的返回值。

如果测试结果为true，就将当前受检测的像素写入后台缓冲区，如果结果为false，就禁止此像素向后台缓冲区的写操作。

运算符是D3D12\_COMPARISON\_FUNC枚举类型所定义的比较函数之一。

(D3D12\_COMPARISON\_FUNC枚举)

11.3 描述深度/模板状态

要描述深度/模板状态，就需要填写D3D12\_DEPTH\_STENCIL\_DESC实例。

(D3D12\_DEPTH\_STENCIL\_DESC结构体)

(D3D12\_DEPTH\_STENCILOP\_DESC结构体)

(D3D12\_STENCIL\_OP枚举)

通过ID3D12GraphicsCommandList::OMSetStencilRef方法设置模板参考值。

11.4 实现平面镜效果

11.4.1 镜像概述

11.4.2 定义镜像的深度/模板状态

11.4.3 绘制场景

11.4.4 绕序与镜像

(代码见书本)

11.5 实现平面阴影

11.5.1 平行光阴影

11.5.2 点光阴影

11.5.3 通用阴影矩阵

11.5.4 使用模板缓冲区防止双重混合

11.5.5 编写阴影部分代码

(代码见书本)