7 利用Direct3D绘制几何体(续)

7.1 帧资源

到目前为止，我们的演示程序在绘制每一帧时都会将CPU和GPU进行一次同步。这样做的原因有两个：

1. 在GPU未结束命令分配器中所有命令的执行之前，不能将它重置。如果不进行同步，那么GPU完成当前第n帧的处理之前，CPU可能会继续执行下一帧(第n+1帧)的相关工作：如果CPU在第n+1帧中重置了命令分配器，那么GPU当前还未处理的命令就会被清除掉；
2. 在GPU未完成与常量缓冲区相关的绘制命令之前，CPU不可更新这些常量缓冲区；

所以，我们在每帧绘制的结尾都会调用D3DApp::FlushCommandQueue函数，确保GPU在每一帧都能正确完成所有的命令的执行。

解决此问题的一种方案是：以CPU每帧都需要更新的资源作为基本元素，创建一个环形数组。我们称这些资源为帧资源，而这种循环数组通常由3个帧资源元素所构成。该方案的思路是：在处理第n帧时，CPU周而复始地从帧资源数组中获取下一个可用的帧资源。趁着GPU还在处理此前帧之时，CPU将为第n帧更新资源，并构建和提交相应的命令列表。随后，CPU会继续针对第n+1帧执行同样的工作流程，并不断重复下去。

不难看出，这种方案还是无法完全避免等待情况的发生。

7.2 渲染项

绘制一个物体需要设置多种参数，我们把单次绘制调用过程中，需要向渲染流水线提交的数据称为渲染项。

7.3 渲染过程中所用到的常量数据

基于资源的更新频率对常量数据进行分组。

7.4 不同形状的几何体

7.4.1 生成柱体网格

7.4.2 生成球体网格

7.4.3 生成几何球体网格

7.5 绘制多种几何体演示程序

7.5.1 顶点缓冲区和索引缓冲区

7.5.2 渲染项

7.5.3 帧资源和常量缓冲区视图

7.5.4 绘制常量

7.6 细探根签名

根签名定义了：在绘制调用之前，需要绑定到渲染流水线上的资源，以及这些资源应如何映射到着色器的输入寄存器中。

7.6.1 根参数

根参数有3个类型可选。

1. 描述符表；
2. 根描述符；
3. 根常量；

考虑到性能因素，可放入一个根签名的数据以64DWORD为限。3种根参数类型占用空间的情况如下。

1. 描述符表：每个描述符表占用1DWORD；
2. 根描述符：每个根描述符占用2DWORD；
3. 根常量：每个常量32位，占用1DWORD。

我们可以创建出任意组合的根签名，只要它不超过64DWORD的上限即可。

7.6.2 描述符表

7.6.3 根描述符

7.6.4 根常量

7.6.5 更复杂的根签名示例

7.6.6 根参数的版本控制

7.7 陆地与波浪演示程序