**7 利用Direct3D绘制几何体(续)**

**7.1 帧资源**

我们的演示程序在绘制每一帧时都会讲CPU和GPU进行一次同步。这样做有两个原因：

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；

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。

这种解决方案效率低下，原因如下：

1，在每帧的起始阶段，GPU不会执行任何命令。

2，在每帧的收尾阶段，CPU会等待GPU完成命令的处理。

解决此问题的一种方案是：以CPU每帧都需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帧的时候，CPU将周而复始地从帧资源数组中获取下一个可用的(即没被GPU使用中的)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，并构建和提交对应的命令列表。

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。如果帧资源数组共有3个元素，则令CPU比GPU提前处理两帧，以确保GPU可持续工作。

(帧资源类的例子：FrameResource)

不难看出，这种解决方案还是无法完全避免等待情况的发生。

**7.2 渲染项**

随着场景中说绘制物体的逐渐增多，如果我们能创建一个轻量级结构来存储绘制物体所需的数据，那真是极好的。

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集称为渲染项。

(渲染项结构体RenderItem)

**7.3 渲染过程中所用到的常量数据**

(查看ShapesApp源码)

**7.4 不同形状的几何体**

本节中，我们将展示如何创建不同形状的几何体，例如椭球体，球体，柱体，圆台和圆锥等。

我们将程序性几何体的生成代码放入GeometryGenerator类中。

(GeometryGenerator类源码)

7.4.1 生成柱体网格

在定义一个柱体时，需要指定其顶底半径，高度，切片数量，以及堆叠层数。

柱体的侧面几何体：生成圆台的基本思路是遍历每个环，并生成列于环上的各个顶点。

(算法实现)

柱体的端面几何体:生成圆台端面的几何体，相当于在其顶面和底面的截面上切割出多个三角形，使之逼近一个圆形

(算法实现)

7.4.2 生成球体网络

欲定义一个球体，就要指定其半径，切片数量及其堆叠层数。

7.4.3 生成几何球体网格

上一小节中，构成球体的三角形面积并不相同，这在某些情况下并非我们所愿。相对而言，几何球体利用面积相同且边长相等的三角形来逼近球体。

(算法实现)

**7.5 绘制多种几何体演示程序**

(例子)

**7.6 细探根签名**

7.6.1 根参数

根签名是由一系列根参数定义而成。根参数有3个类型可选。

1，描述符表：描述符引用的是描述符堆中的一块连续范围，用于确定要绑定的资源。

2，根描述符：通过直接设置根描述符即可指示要绑定的资源，而且无需将它存于描述符堆中。但是，只有常量缓冲区的CBV，以及缓冲区的SRV/UAV才可以根描述符的身份进行绑定。这也就意味着纹理的SRV并不能作为根描述符来实现资源绑定。

3，根常量：借助根常量可直接绑定一系列32位的常量值。

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

根描述表：占1DWORD；

根描述符：(64位的GPU虚拟地址)占2DWORD；

根常量：每个常量32位，占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。

CD3DX12\_ROOT\_PARAMETER是对D3D12\_ROOT\_PARANETER的扩展。

**7.7 陆地与波浪演示程序**

(代码见书本)

7.7.1 生成栅格顶点

7.7.2 生成栅格索引

7.7.3 应用计算高度函数

7.7.4 根常量缓冲区视图

7.7.5 动态顶点缓冲区

(详见书本)