# 8.10 地形纹理演示程序

在本例中，我们要为地形和水体添加纹理。首先，我们要在地形上平铺一幅草地纹理。 由于地形网格很大，如果我们直接拉伸纹理，那么每个三角形只能得到很少的几个纹理元素。换句话说，这里无法为表面提供足够高的纹理分辨率；我们会受到倍增问题的影响。所以，我们要在地面网格上平铺草地纹理，进而获得较高的分辨率。其次，我们要通过一个时间函数对水体纹理进行平移，使水体显得更真实一些。图8.15是该演示程序的屏幕截图。

****

**图8.15 地形纹理演示程序的屏幕截图。**

## 8.10.1 生成网格纹理坐标

图8.16是一个建立在*xz*平面上的*m*×*n*网格以及一个在规范化纹理空间[0,1]2中的对应网格。可以看到，*xz*平面上的第*ij*个网格顶点的纹理坐标对应于纹理空间中的第*ij*个网格顶点的坐标。第*ij*个顶点对应的纹理空间坐标为：

*u*ij = *j* ∙ ∆*u*

*v*ij = *i* ∙ ∆*v*

其中，∆*u* = ，∆*v* = 。

****

**图8.16 空间中的网格顶点vij的纹理坐标等于*uv*空间中的第*ij*个网格顶点Tij的坐标。**

因此，我们可以使用如下代码为地面网格生成纹理坐标：

void GeometryGenerator::CreateGrid(float width, float depth, UINT m, UINT n, MeshData& meshData)

{

 UINT vertexCount = m\*n;

 UINT faceCount = (m-1)\*(n-1)\*2;

 //

 // 创建顶点

 //

 float halfWidth = 0.5f\*width;

 float halfDepth = 0.5f\*depth;

 float dx = width / (n-1);

 float dz = depth / (m-1);

 float du = 1.0f / (n-1);

 float dv = 1.0f / (m-1);

 meshData.Vertices.resize(vertexCount);

 for(UINT i = 0; i < m; ++i)

 {

 float z = halfDepth - i\*dz;

 for(UINT j = 0; j < n; ++j)

 {

 float x = -halfWidth + j\*dx;

 meshData.Vertices[i\*n+j].Position = XMFLOAT3(x, 0.0f, z);

 meshData.Vertices[i\*n+j].Normal = XMFLOAT3(0.0f, 1.0f, 0.0f);

 meshData.Vertices[i\*n+j].TangentU = XMFLOAT3(1.0f, 0.0f, 0.0f);

 // 在网格上拉伸纹理

 meshData.Vertices[i\*n+j].TexC.x = j\*du;

 meshData.Vertices[i\*n+j].TexC.y = i\*dv;

 }

 }

 …

}

## 8.10.2 纹理平铺

前面提到，我们希望在地形网格上平铺一幅草地纹理。但是，目前计算出来的纹理坐标是在单位区间[0,1]2中的，无法产生平铺。所以，我们要指定重复寻址模式并通过一个纹理变换矩阵将纹理坐标扩大5倍。这样，纹理坐标会被映射到[0,5]2区间内，使纹理在地形网格表面平铺5×5次：

XMMATRIX grassTexScale = XMMatrixScaling(5.0f, 5.0f, 0.0f);

XMStoreFloat4x4(&mGrassTexTransform, grassTexScale);

…

Effects::BasicFX->SetTexTransform(XMLoadFloat4x4(&mGrassTexTransform));

…

activeTech->GetPassByIndex(p)->Apply(0, md3dImmediateContext);

md3dImmediateContext->DrawIndexed(mLandIndexCount, 0, 0);

## 8.10.3 纹理动画

我们要通过一个位于**UpdateScene**方法中的时间函数在纹理空间中平移纹理坐标，使水体纹理在网格上移动。我们为每帧提供一个很小的位移量，以得到一个平滑动画。我们同时使用无缝纹理和重复寻址模式，以使纹理坐标在平移时不出现间断。下面的代码示范了如何为水体纹理计算位移量，并生成和设定水体的纹理矩阵：

// 平铺水面纹理

XMMATRIX wavesScale = XMMatrixScaling(5.0f, 5.0f, 0.0f);

// 根据时间平移纹理

mWaterTexOffset.y += 0.05f\*dt;

mWaterTexOffset.x += 0.1f\*dt;

XMMATRIX wavesOffset = XMMatrixTranslation(mWaterTexOffset.x, mWaterTexOffset.y, 0.0f);

// 组合缩放和平移

XMStoreFloat4x4(&mWaterTexTransform, wavesScale\*wavesOffset);

…

Effects::BasicFX->SetTexTransform(XMLoadFloat4x4(&mWaterTexTransform));

…

activeTech->GetPassByIndex(p)->Apply(0, md3dImmediateContext);