**材质系统**

GB的材质系统有两种方式生成可用的Nishader,加法和减法

1. 加法（additive material system），顾名思义，就是根据渲染对像的状态、条件动态生成一套Nishader
2. 减法（subtractive system），A subtractive system will have an entire shader program and removes pieces from it using compile-time preprocessor commands.

NiMaterial和NiShader区别：

An NiShader contains the entire render state for how a specific geometry object will be drawn. It can span multiple render passes if necessary, but it is intended to describe to the renderer how to draw a certain object in a certain set of circumstances. A single NiShader instance can be shared among multiple geometry objects, if their structure and rendering circumstances coincide, but for the most part, an NiShader should not be able to easily handle different situations. For example, a single NiShader may be able to draw an unskinned object that has a base map and a dark map. If the same NiShader object is applied to a geometry that is missing a dark map, or is skinned, then the NiShader may fail. That is expected, because that NiShader was not written to be able to handle those circumstances.

NiMaterialResource： Uniform constants are encapsulated in the class NiMaterialResource. Nodes in the tree are encapsulated in the class NiMaterialNode.

NiMaterialConfigurator： a shade tree is encapsulated in an NiMaterialConfigurator object.

NiMaterialResourceBinding :Connections between the input and output resources of a given node and other nodes or uniform constants are encapsulated in the NiMaterialResourceBinding class.

在gamebryo产生shader步骤：

NiMaterialConfigurator封装shade tree

NiGPUProgramDescriptor的某个继承类通用设置的"位",确定哪些"统一常数"(在NiMaterialResource中)需要连接到树的结点(NiMaterialNode).连接用NiMaterialResourceBinding类实现

一旦所有连接完成，调用NiMaterialConfigurator的评估函数来产生NiGpuProgram和一些输入输出“统一常数”

NiMaterialFragmentNodes 构建shade tree的代码片段，根据不同的着色语言、平台、硬件选择给定的代码片段。

Fallbacks:

当创建shader失败,重建NiGPUProgramDescriptors回到第4步去重试，可能简化效果到多步完成(multiple passes)或者丢弃某些灯光或纹理。

NiMaterialInstance：与应用层接口，用来创建NiShader

NiMaterialLibrary：材质库，A material library contains a set of material descriptors and has the ability to generate materials as needed.

NiGPUProgramCache：The NiGPUProgramCache is a system designed to contain NiGPUPrograms that have been compiled with an additive material system.

一个对象可以使用多个材质，但是同时只有一个处在活动状态。举例：shadow map渲染的时候，需要shadow cast渲染和正常渲染两种方式，在GB中可以把两个shader都加入到该对象，由程序控制在什么时候该使用哪个shader.

**图片相关：**

NiImageRender

NiBMPReader

NiDebugVisualizationClick 在游戏中可以使cameras, lights, bounding volumes bones虚拟显示

**地形相关：**

地形渲染shaer name:TextureBlendMaterial 四层贴图混合方式

地形渲染方法：

Create nimesh

创建stream (AddStreamGetLock)

设置property

NiTerrainDecal 贴地效果

**18:13 2010-8-26**

win32文件操作例子：

NiTerrainSectorFile::Open

NiStreamLoadBinary

地形：

NiTerrainDataLeaf:地形的数据仓库，保存地图块贴图、顶点、meta数据等资源引用，本质上是个数据索引，真正的数据放在父sector上。

一个块最多8张贴图

NiTerrainSectorData:记录sector中叶节点的数据和变量，如块大小、LOD数量、顶点数据等。

动态管理TerrainBlock顶点缓冲区(NiDynamicStreamCache)

NiTerrainSectorFile: 读取/保存sector文件数据。如顶点高度、法线等。

NiTerrainBlock：负责地型可渲染的几何对象，

NiTerrainMaterial:材质