Sagitta 场景管理设计

**1 概述**

一个场景对应于一个SceneManager， SceneManager保存有场景中所有的SceneObject：Node，Renderable，Camera，List等，并且这些Object的清理均由SceneManager来处理。

场景的组织由Node来进行。Node负责场景的所有变换，且场景的所有变换只能由Node来完成。任何属于场景的物体（SceneObject）必须要挂载（attach）到一个Node上才能够进行操作如平移旋转缩放等。整个场景靠一个Node的Tree组织起来。

场景消亡时，首先遍历所有节点，卸载（detach）掉其上的所有SceneObject，然后清空渲染列表（RenderQueue），再遍历SceneManager的各类型Object列表，删除所有SceneObject。

**2 SceneManager**

SceneManager统一管理所有的场景物体，包含有SceneNodeMap，RenderableMap，CameraMap，LightMap以及RenderQueue。另外也包含有一个RenderSystem，渲染的工作有它负责。

一个SceneManager对象被创建时，自动创建其根节点，以后一般调用SSceneManager::createSceneNode()来创建新的节点，节点被默认作为根节点的子节点创建，用户可自己设置其父节点，通过调用父节点的SNode::addChild()函数。要查找场景中的节点，可以调用SSceneManager::getSceneNode()函数。

渲染时遍历RenderQueue，逐一渲染其中的每个物体，渲染工作有RenderSystem完成。

**3 Node**

节点用于组织场景，以及实现场景中的各种变换，Sagitta的系统中，只有Node有变换操作（相机（Camera）也提供了旋转和平移的变换，但本质上是操作其所挂载的节点）。因此要实现对某个物体的变换必须将其挂载到一个节点上。也只有被挂载到节点上的物体（Object）才会被添加到SceneManager的相应物体队列中去，若是可渲染并可见的物体，则还会被添加到RenderQueue中。当一个物体从一个节点上卸载时（detach），根据选择挂载到原节点所在的场景的根节点上并设置为不可见（invisible）或者是被挂载到其他场景的某个节点上去。

**4 Camera**

相机用于生成相机变换矩阵（view matrix）和投影矩阵（project matrix），它们同样被挂载到节点上才能工作。在SceneManager创建时，会生成一个默认的相机，其位置为世界坐标的原点，方向指向世界坐标Z轴负方向，向上为世界坐标y轴正方向。每个相机创建时必须指点其所在的SceneManager，并将其挂载在该场景管理器根节点新生成的子节点上。Camera直接提供平移和旋转变换的接口，实质上是操作其所挂载的节点。一般创建一个新的相机可以使用SceneManager::createCamera()。

Camera的相机变换矩阵（view matrix）的生成按如下几个公式进行。

r = r \* q, u = u \* q, d = d \* q

(r为向右的向量，u为向上的向量，d为相机指向的方向，q为从绑定节点取得的绝对朝向的四元数)

r.x(), u.x(), d.x(), 0.0f,

view matrix = r.y(), u.y(), d.y(), 0.0f,

r.z(), u.z(), d.z(), 0.0f,

(-r).Dot(pos), (-u).Dot(pos), (-d).Dot(pos), 1.0f

对于OpenGL，该矩阵还需右乘如下矩阵

1, 0, 0, 0,

0, 1, 0, 0,

0, 0, -1, 0,

0, 0, 0, 1

才能传给glMultMatrixf()。

**5 Esse**

Esse是场景中除去Node的物体，包括Renderable、Light、Camera等，允许添加其他类型，但需要指明类别。Esse挂载在SceneNode上，所有者为SceneNode。在SceneNode上按类别保存一个map，每一个类别按ID保存为一个map。

Esse进入场景的方式有三种：

1 直接调用SSceneManager::\_addEsse()添加，不推荐这么做；

2 挂载到一个已经在Scene中的SceneNode上，SSceneNode::attachEsse()会调用SSceneManager::\_addEsse()将其添加到场景；

3 当调用SSceneManager::addSceneNode()时，该SceneNode上的所有Esse会被添加到Scene中。

Esse不能游离在场景中，即当它从场景中的节点中卸载时，需要从场景中去除——SSceneNode::detachEsse()调用SSceneManager::\_removeEsse()。

**6 SceneNode**

SceneNode是场景中的组织者，继承自Node。SceneNode允许挂载Esse。

SceneNode进入场景的方式有三种：

1直接调用SSceneManager::addSceneNode()；

2调用场景中某个节点的addChild()点添加为该节点的子节点。SSceneNode::addChild()会调用SSceneManager::addSceneNode()；

3 SSceneManager::addSceneNode()会将目标节点的子节点都添加到场景中。