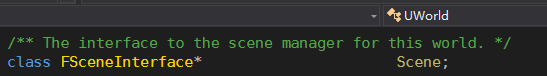
UE4分析

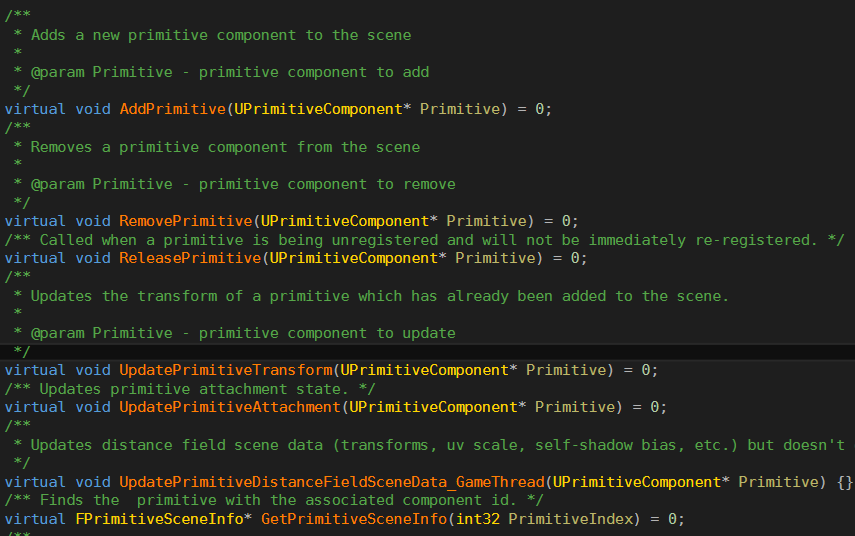
# 渲染

位于引擎中Engine\Source\Runtime\Engine\Public\SceneInterface.h的FSceneInterface接口类是一个UWorld对应的渲染接口：

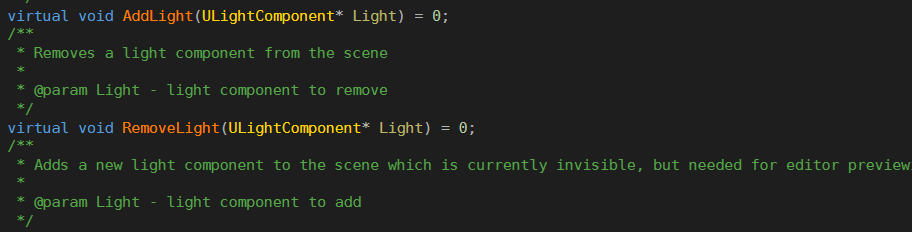


它主要有以下接口函数：

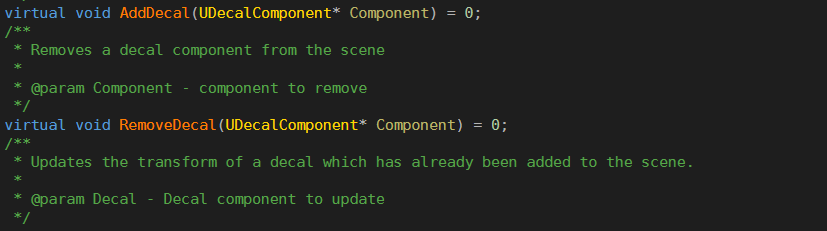
-相关Primitive组件：



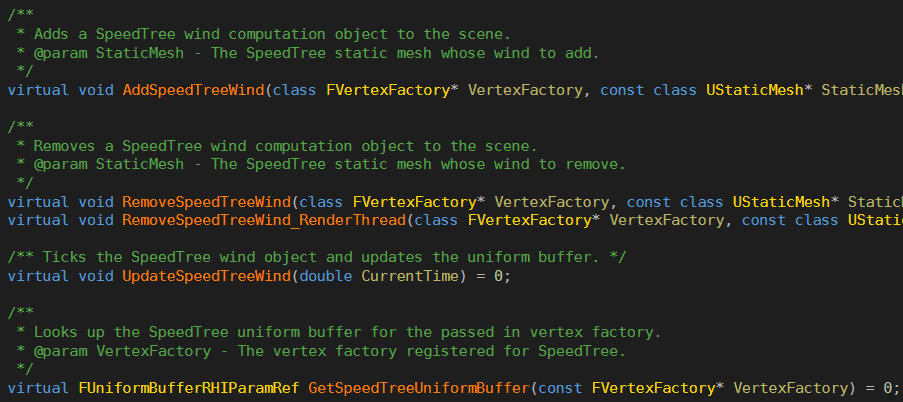
-相关Light组件：



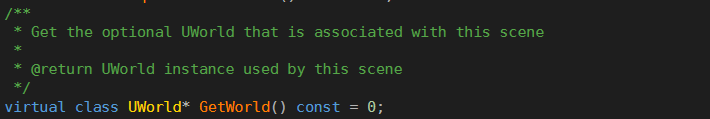
-相关Decal组件：



-等等其他的可渲染组件，其中也有SpeedTree的存在(SpeedTree是一款树的建模软件，有专门的UE4版本)

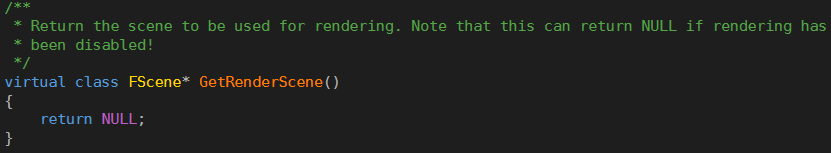


这个类是和UWorld相关联的

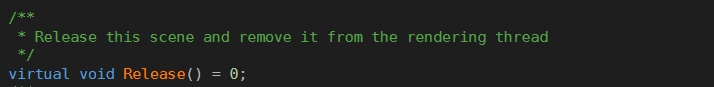




并且有相应的Scene



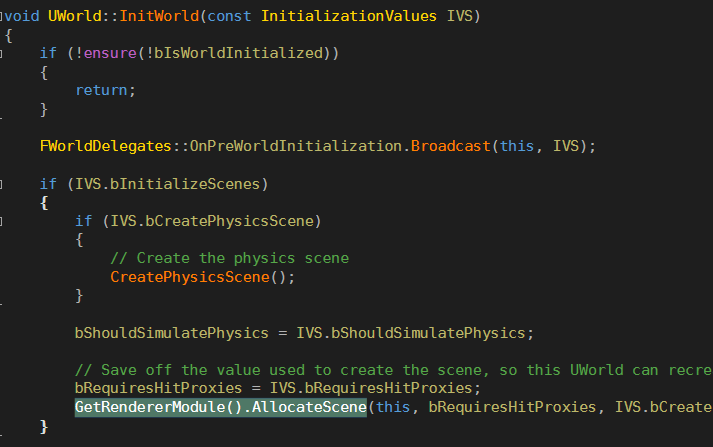
当然也和渲染线程相关（~只从接口类暂时不清楚如何关联~）



这个FSceneInterface接口类是一个实现了FSceneInterface的FScene的接口，而FScene通过GetRendererModule().AllocateScene构建；

其中找到的一个调用是在World.cpp中的InitWorld()初始化World时进行的：

从具体的实现中能够发现这个世界渲染与否和World的初始化参数中的bInitializeScenes参数相关。而Scene这个单词可以理解为是World在渲染模块中的概念

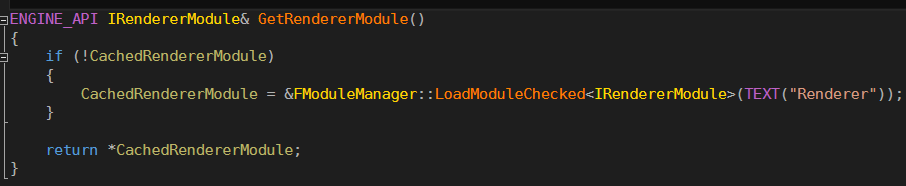


其中这个GetRendererModule()是UE4中的“ENGINE\_API”

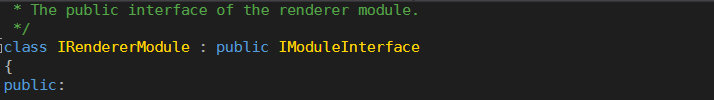
可能是UE4提供的引擎API

它是位于Engine\Source\Runtime\Engine\Public\EngineGlobals.h中

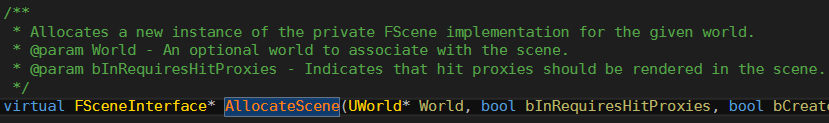
其功能一目了然：返回或加载返回渲染模块<IRendererModule>



IRendererModule是声明在Engine\Source\Runtime\RenderCore\Public\RendererInterface.h中的，继承自IModuleInterface（所有游戏引擎用到的模块都应当继承于此，它只有一些加载，卸载，等等的基函数）（注意: IRendererModule依然是一个接口类）



AllocateScene的定义



到此一个渲染模块已经存在，并且通过渲染模块的AllocateScene分配了FScene的实例到UWorld上，也就是 < FSceneInterface \*>Scene，从World中的RecreateScene()中可以看到World中的< FSceneInterface \*>Scene与渲染模块IRendererModule是紧密相关的，而且RecreateScene的操作只可能在AllocateScene中进行，因此推断AllocateScene生成了相关这个World的FScene对象，而且为这个World中的成员变量< FSceneInterface \*>Scene赋予了初始值即生成的FScene对象



这里小结一下：

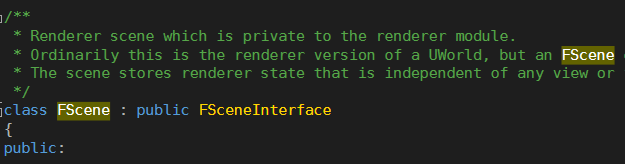
UWorld中存在一个成员变量<FSceneInterface\*>Scene【UWorld中主要UWorld::CreateFXSystem(),

UWorld::UpdateLevelStreamingInner(),

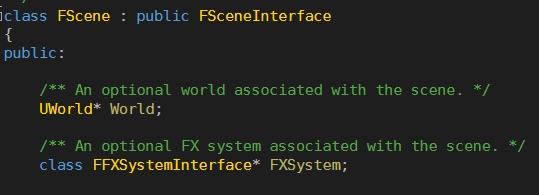
UWorld::UpdateParameterCollectionInstances()存在相关的调用】

而在UWorld的InitWorld()中，通过调用GetRendererModule().AllocateScene()生成了相关这个UWorld的FScene对象，且为这个UWorld中的成员变量< FSceneInterface \*>Scene赋予了这个初始值【注意：FScene是继承于FSceneInterface的，所以UWorld中的FSceneInterface\*变量命名是Scene】

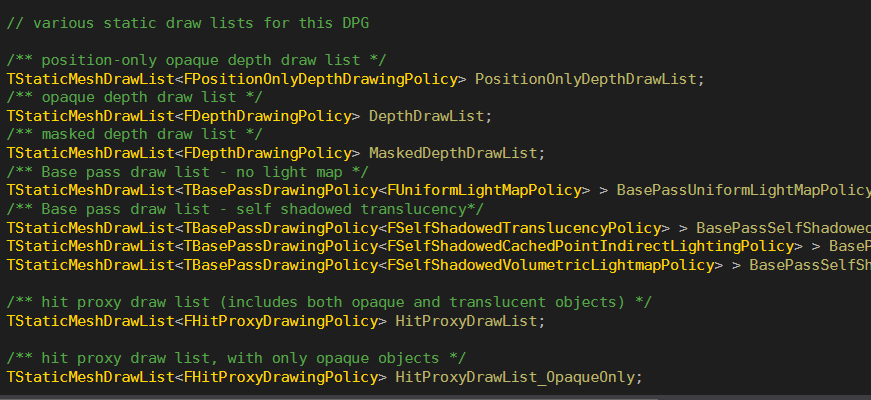
FScene声明于Engine\Source\Runtime\Renderer\Private\ScenePrivate.h中，继承自FSceneInterface



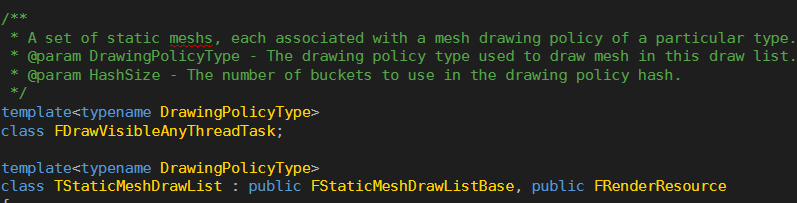
FScene中有当前相关世界的索引和FXSystem接口（FX在UE4中指一些特效）



最重要的，它有各种DrawList，概念类似于渲染队列？

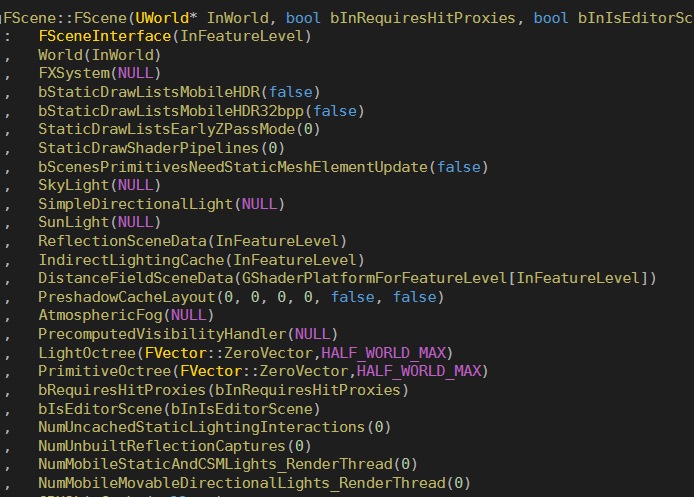


其注释如下：



渲染模块中的主要函数：

FScene的构造函数，有一堆的初始化列表项



关键的过程：

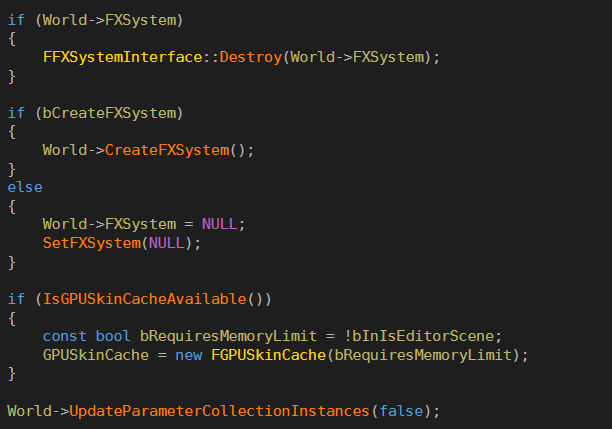
1.check World成员（初始化列表中初始过）并设置World中的接口引用自身



2.FScene中的FeatureLevel的赋值



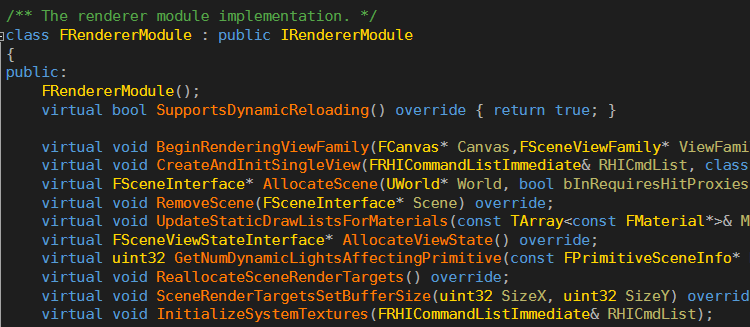
3.对World的设置（FXSystem相关等）



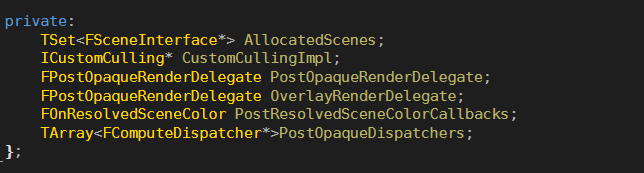
……

FRendererModule声明于Engine\Source\Runtime\Renderer\Private\RendererModule.h

继承于IRendererModule接口

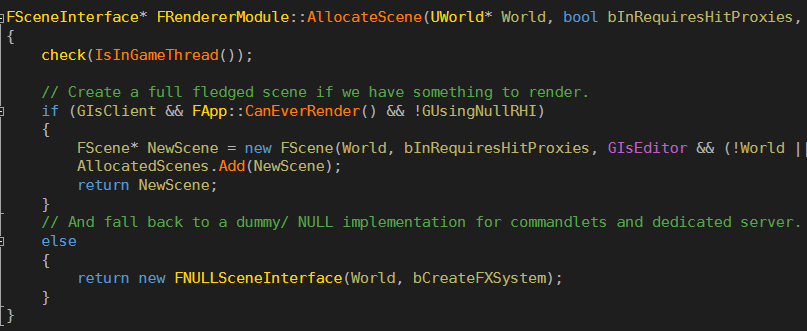


关键的成员变量：



AllocateScene的实现：

FScene与UWorld之间的过程主要由FScene的构造处理，AllocateScene中主要过程是FScene与自身的关系：AllocatedScenes.Add(NewScene);因此可以看出FRendererModule是相当于一个管理器（管理FScene）的存在，（~但独立的游戏只有一个World，只有一个FScene，那么这个FRendererModule中只有一个元素~）



# 物理