**雪人场景作业说明**

**运行环境为windows 32位，基于vs2017和directx11,使用Unicode字符集**

1. 场景种类说明

场景分为6部分，分别为天空盒、立方体、两个雪人、基于高度图的地形、公告牌之树、雪花粒子系统。除了天空盒以及雪花例子，其他的场景都有雾化功能，200开始雾化，500场景完全消失；且都简单的设置了碰撞检测；相机漫游地形不会出界，会固定在地形之内；地形应用了曲面细分和纹理混合功能；箱子纹理应用了两张纹理图，且相机照射上去会有聚光灯；W、S、A、D键用于场景漫游，鼠标持续左击并左右摇摆用于相机左右旋转视角，鼠标持续左击并上下摇摆用于相机上下旋转视角。

图1-1是类的关系说明图。

SecenBase

SnowParticle

Tree2D

Sky

Box

SnowMan

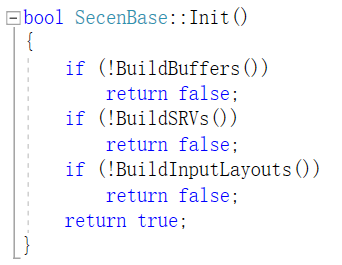
Terrain

图1-1 场景类关系图

以下讲解各个场景实现

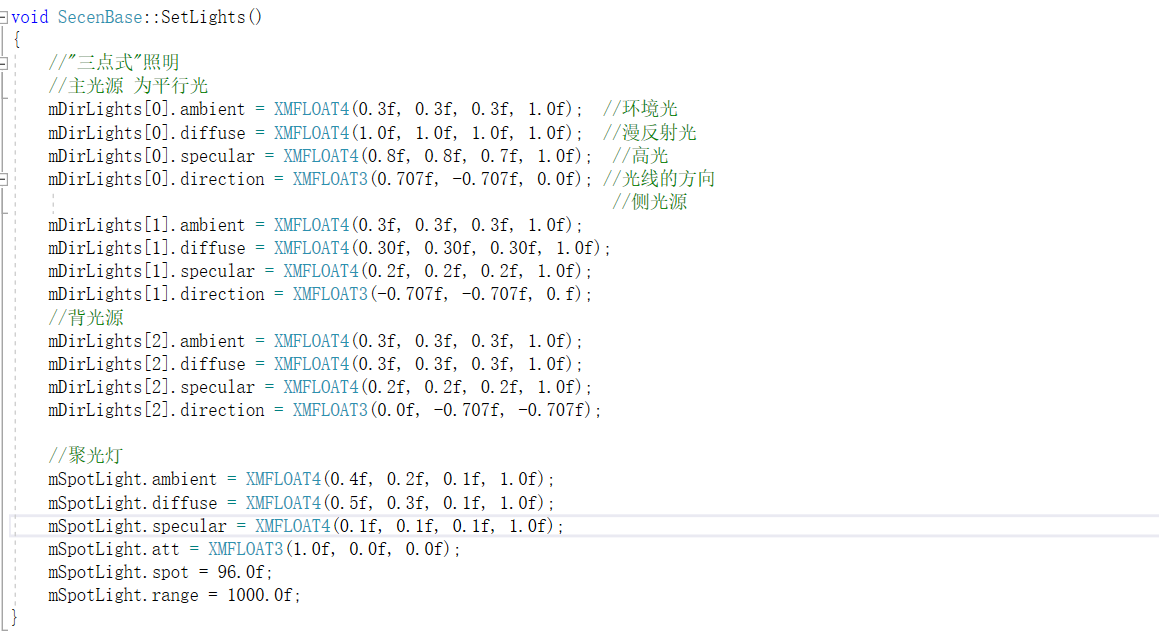
SecenBase类：

SecenBase类为所需绘制各种场景的基类，主要的虚函数为BuildBuffers(顶点和索引缓冲区相关设置) ，有个辅助函数BuildBuffers\_被BuildBuffers调用以提高代码复用；BuildSRVs(纹理文件读取)；BuildInputLayouts(输入布局设置)，有个辅助函数BuildInputLayouts \_被BuildInputLayouts调用以提高代码复用；这三类虚函数会被子类实现。SecenBase类会有一个非虚函数Init，调用这三个虚函数。见下图：



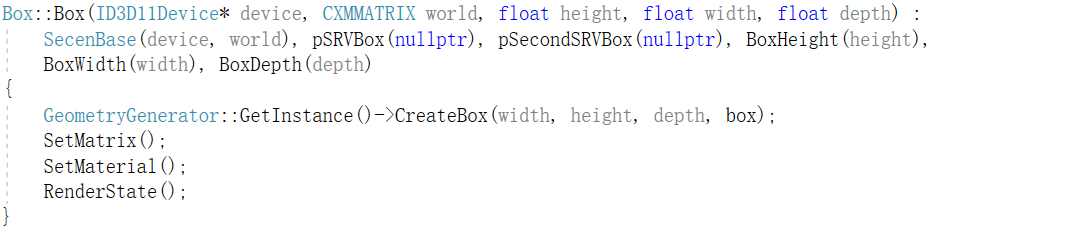
虚函数Draw被子类实现用于绘制自身，虚函数SetMaterial被子类用于设置自身场景的材质属性。

光属性已经设置好，为固定数据，函数为SetLights，见下图：

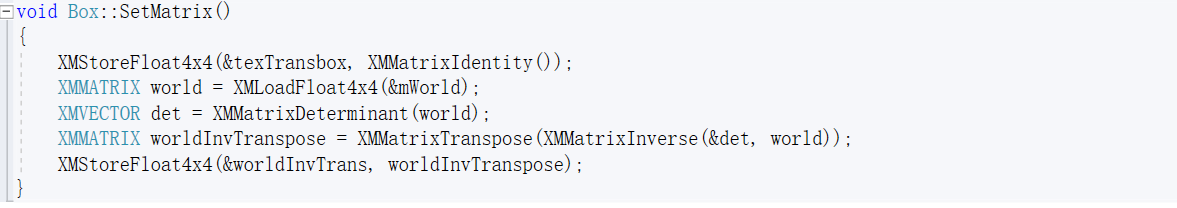
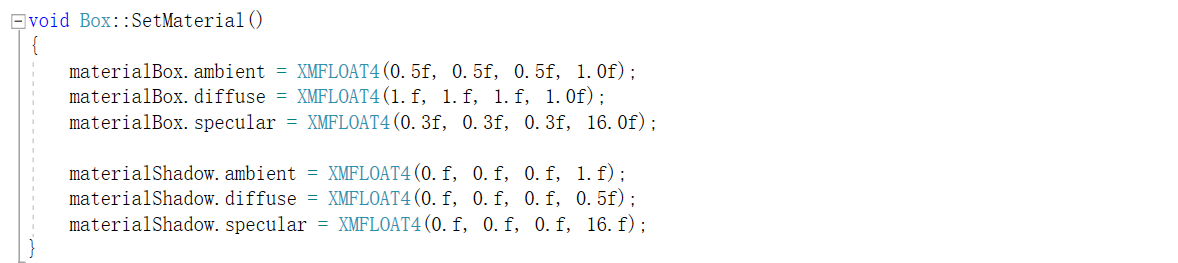


使用函数而不创建用光类的原因是场景简单，并且光属性不复杂。

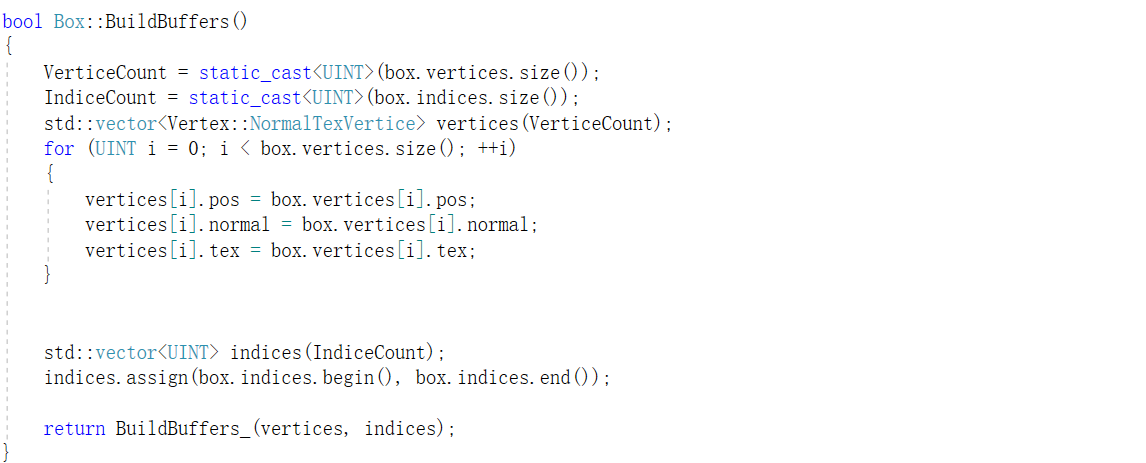
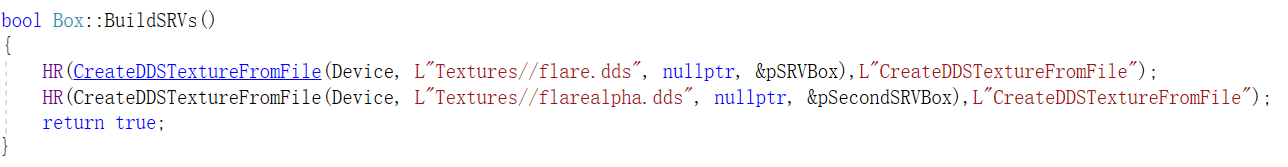
Box类：



初始化的时候根据长宽高先创造一个箱子、然后设置矩阵、箱子的材质、RenderState用于设置在绘制箱子平面阴影前的状态。

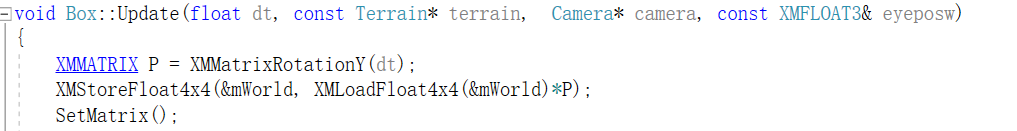
 

三个主要虚函数的实现：

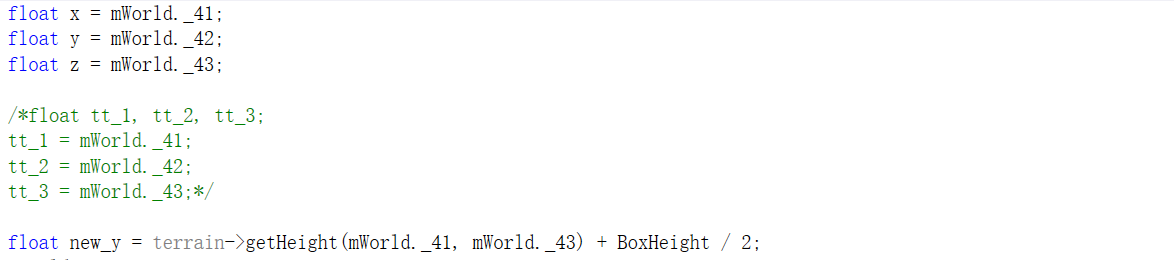
  

由于立方体箱子做圆周运动，且要求箱子以第一视角漫游的时候，可以漫游到立方体上，所以有自己的Update函数，主要实现如下功能：

1. 旋转做圆周运动

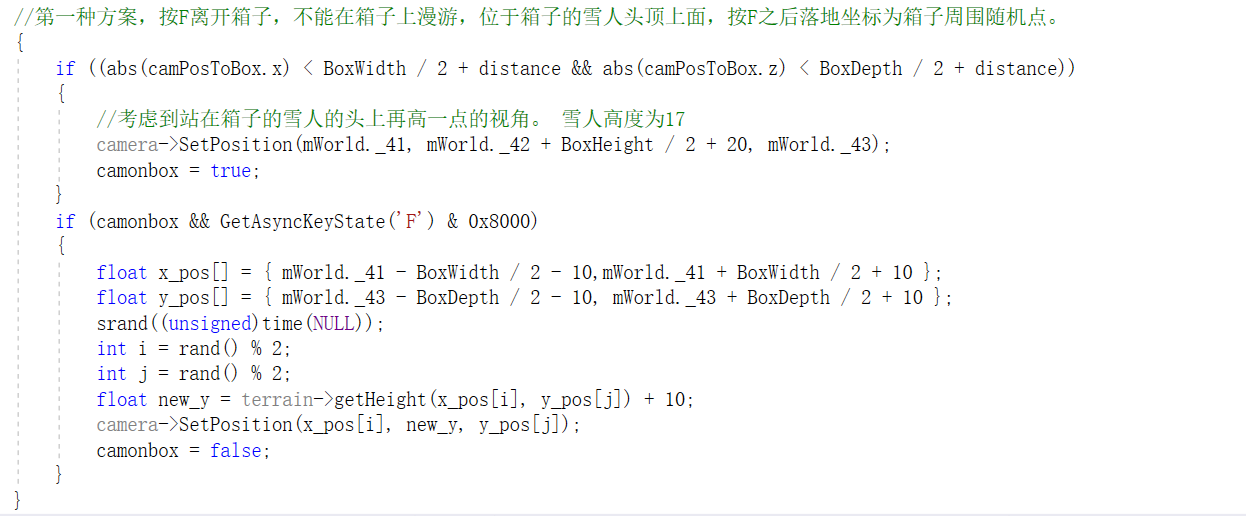


1. 箱子一直置于地形高度之上



1. 当相机距离箱子一定距离的时候的时候，自动漫游到箱子上面，在箱子上的相机高度比站在箱子上的雪人高一点。下车有两种方案：

（1）按键下车。



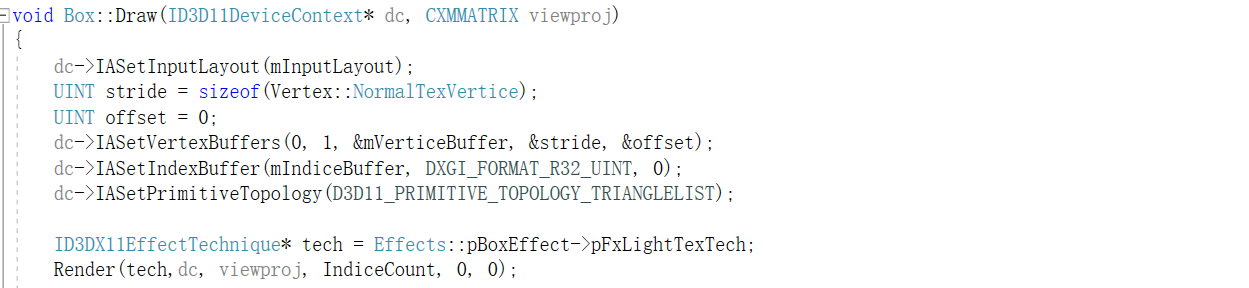
（2）在箱子上漫游距离，然后距离箱子中心一定距离后自动下车



1. 设置相关的平面阴影矩阵，由于其需要地形和相机，所以就放在Update函数里。



然后是Draw函数的实现



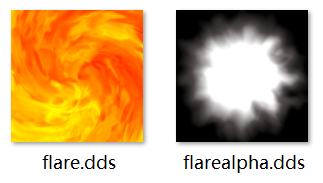
Render是基类SecenBase已经实现的函数，主要为了提高代码复用。



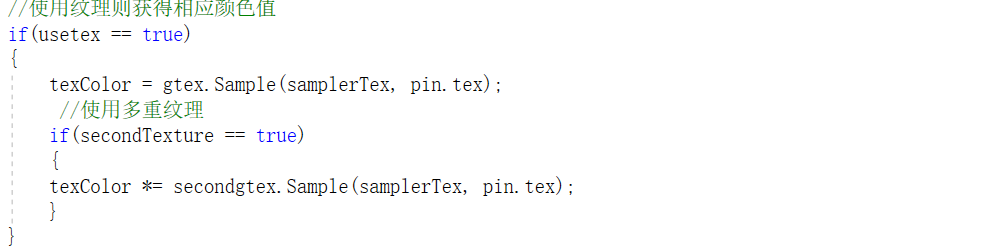
里面的SetEffect为虚函数，需要子类场景自己实现，以下是箱子的相关实现。



至此，箱子的初始化，update和Draw叙述完毕，这三个函数会被SnowDemo类实现调用，后续会有描述，先叙述箱子纹理相关操作。两个纹理图像。



在Box.fx当中为简单的相乘

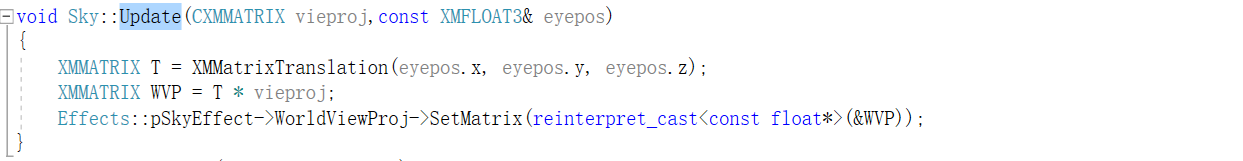


Effect.h和Effect.c参照龙书简单的进行了一些改变。很容易看明白。不介绍了。

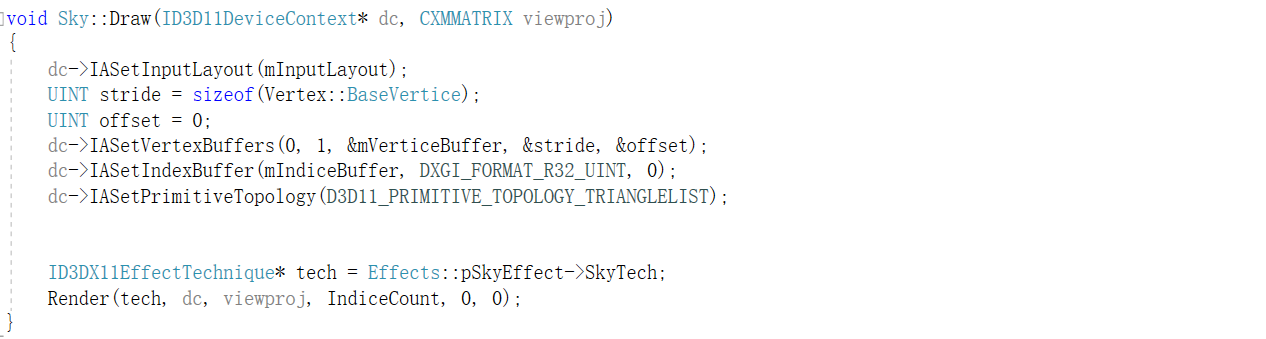
Ps:在绘制箱子的平面阴影时发现并不适用于凹凸不平的地形，所以其它场景就没进行阴影绘制了，shadowmap还在研究，绘制上出了点bug要解决。需要改进。

Sky类:

需要注意的是相机视角相对天空盒的设置。在Update当中实现。



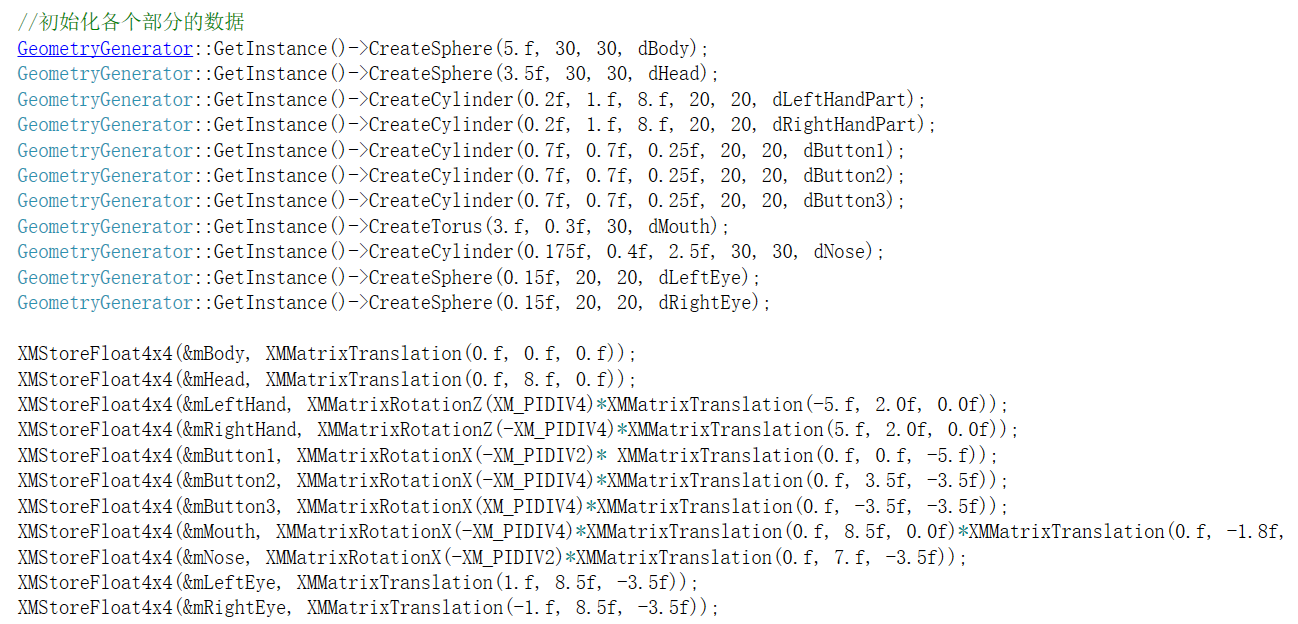
然后是绘制函数Draw



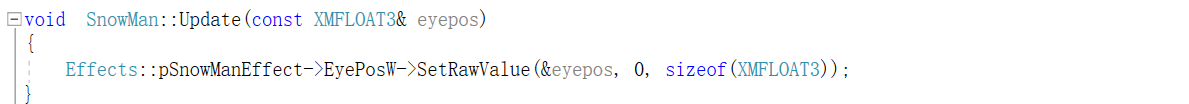
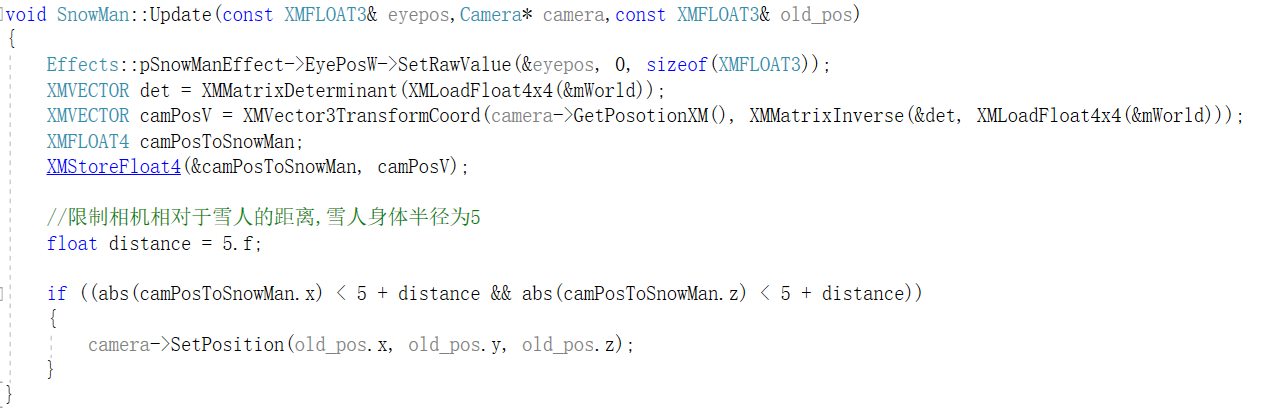
SnowMan类：

主要由球体，圆环，圆柱等制作的雪人，由头部、眼睛、嘴巴、鼻子、身体、手部、纽扣组成。

注意的就是雪人各部分部件之间的组合，必须组合起来是个雪人，得实验多次。下图为组成的具体实现，是以身体为主调整其它部件的坐标。

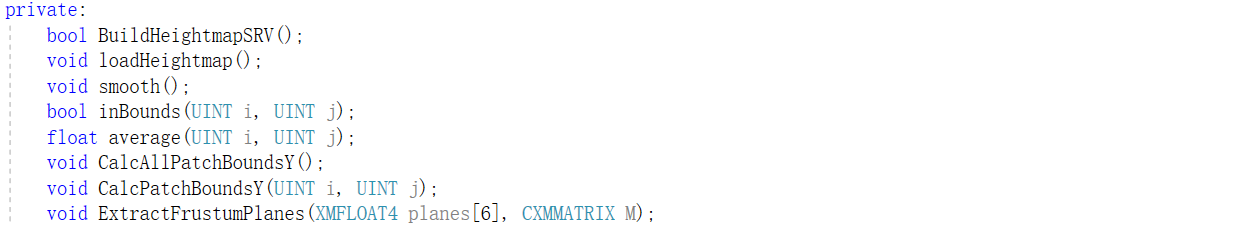


自身有两个Update实现，其中一个用于位于地面上的雪人设置简单碰撞检测，一个用于设置视角。



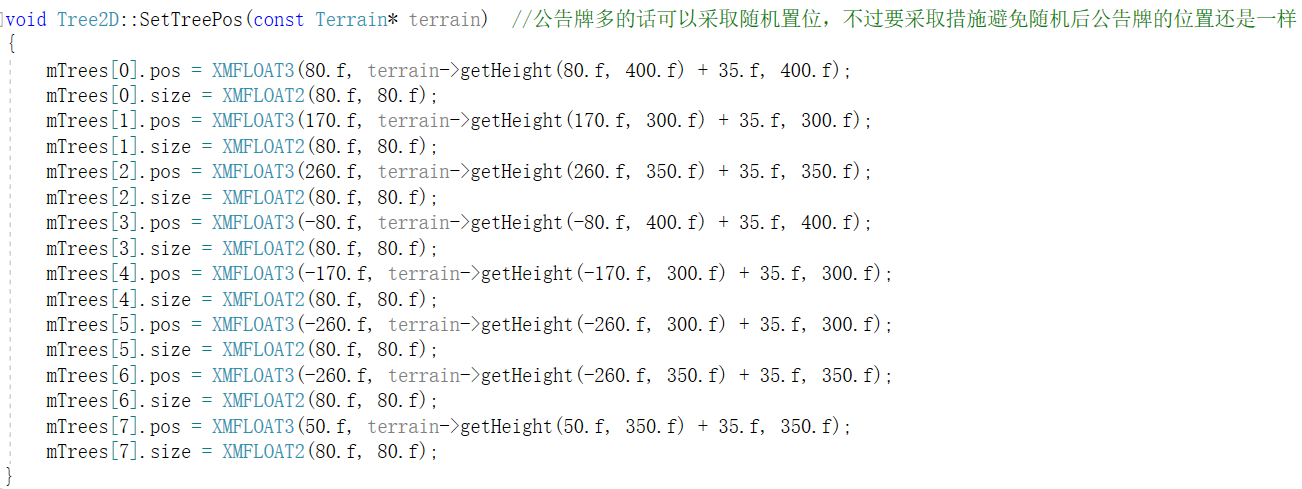
Terrain类：

高度图的读取以及为了更好的绘制地形进行了相关的平滑处理以及曲面细分处理，基于此，实现了一些相关的私有函数。地形纹理采用的纹理数组，用github上的texassemble制作的纹理数组。



Tree2D类：

原理是公告牌，具有欺骗视角的性质。读取的是纹理数组，用的texassemble制作的纹理数组。需要注意的是为了让其能够正确的位于地形之上，首先得基于地形高度设置好其具体的坐标。



SnowParticle类：

粒子的主要区别是fx文件的实现。

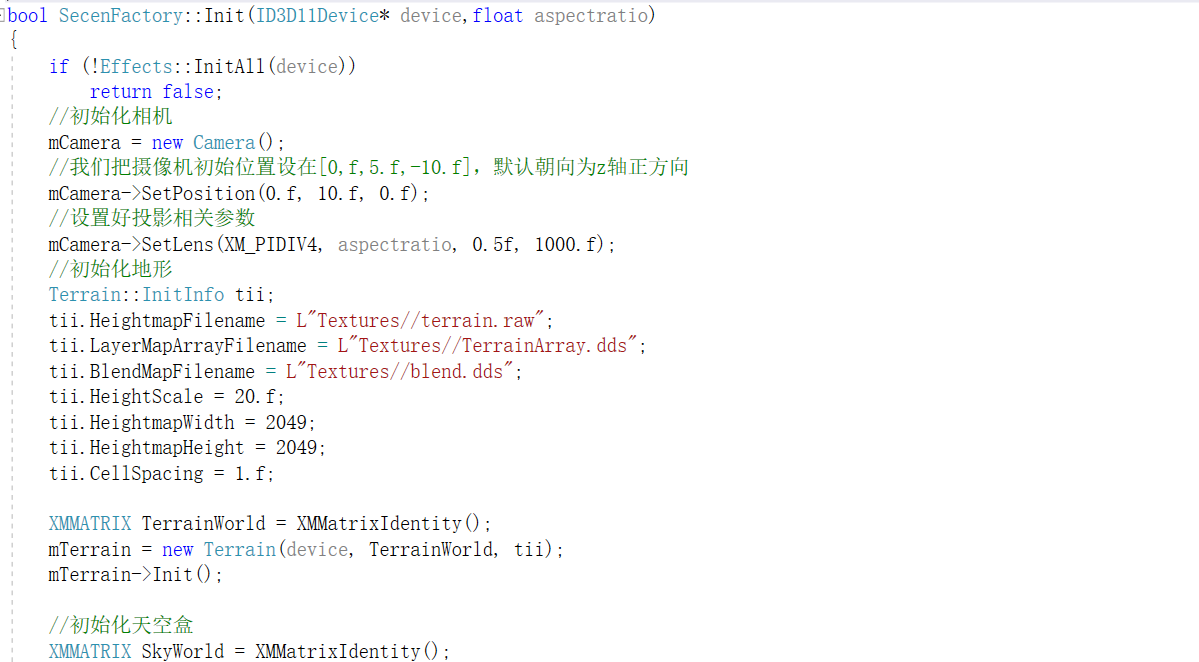
SecenFactory类：

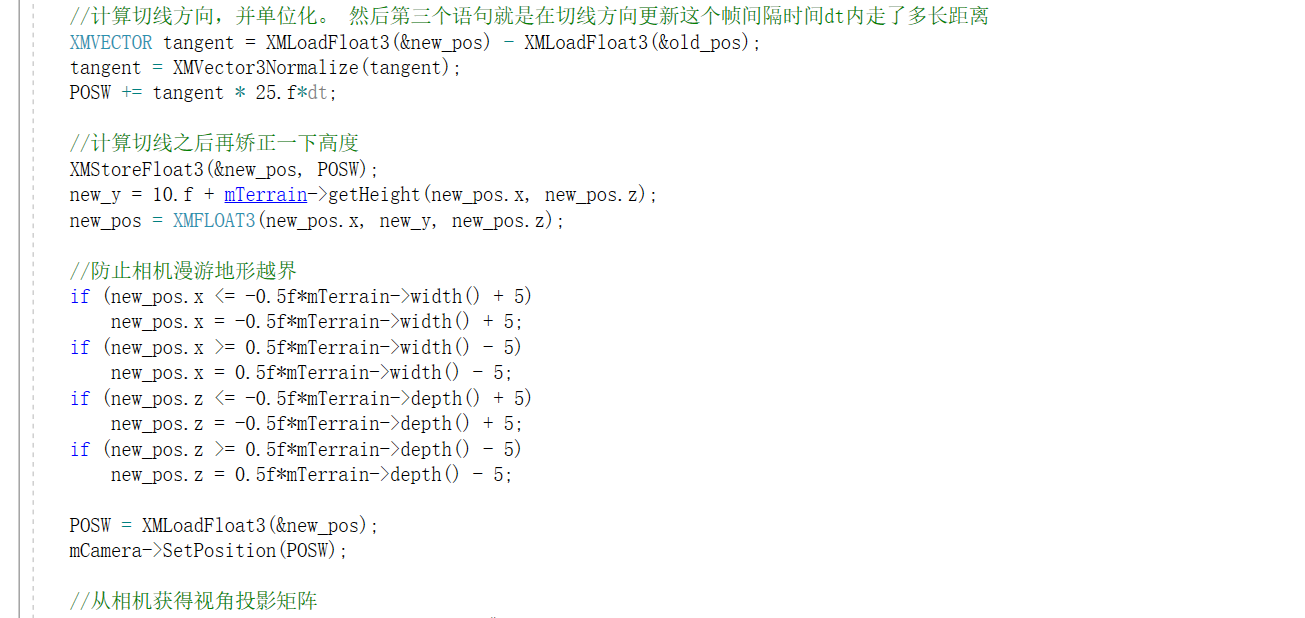
为了便于管理各种场景，实现了类似于工厂模式的类SecenFactory，用于管理这些场景。

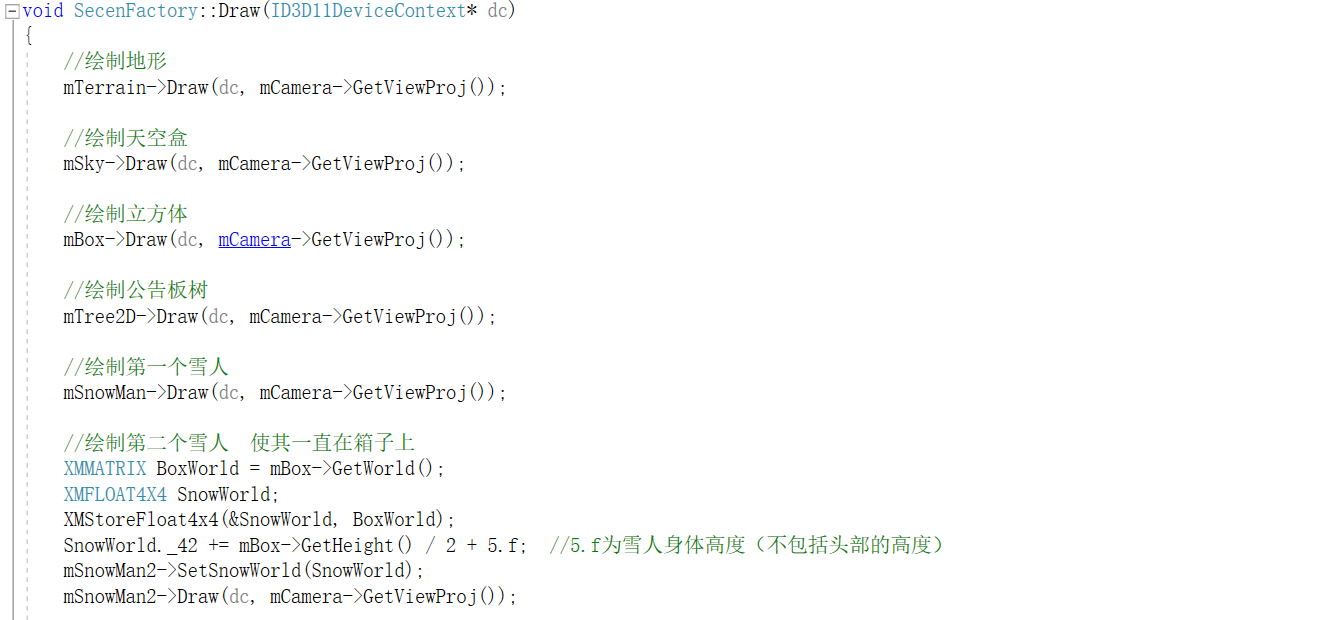
主要的函数就三个，分别为Init，Update，Draw函数，数据成员如下：



三个函数具体实现如下：

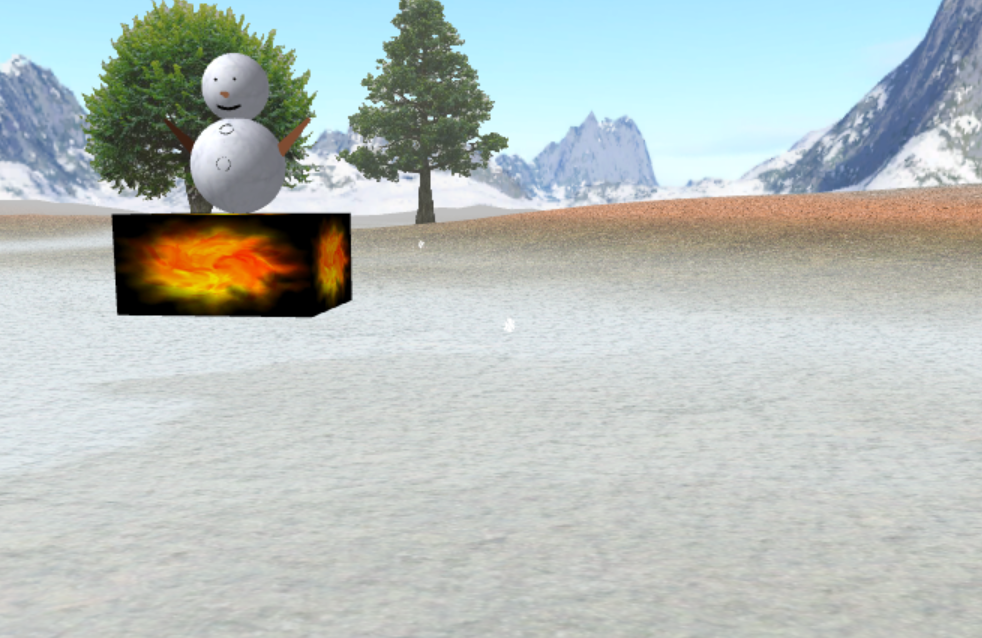




SecenFactory类对象会存在SnowDemo类里，SnowDemo继承自Dx11Base，主要是3D3设备驱动交换链等相关的设置，用于将场景呈现出来，SecenFactory类对象方法会被SnowDemo调用进而完成场景的绘制。

下面展现一些场景图：





相机位于圆周运动的箱子上面：

