HLOD

HLOD

静态处理流程

> 在Unity层级界面新建一个空物体，添加一个HLOD组件，然后将需要构建HLOD结构的物体都放到这个物体下面，点击HLOD组件的生成按钮，即可开始生成HLOD层级结构，生成的过程如下。

构建包围盒

获取HLOD物体下所有OBJ的Renderer对象，随后将其包围盒转换到HLOD这个物体所在的**OBJ空间**，将HLOD物体下的子物体的包围盒合并组成一个大的包围盒。这个大的包围盒包含所有的属于HLOD的子物体。

构建四叉树

> 树的划分是按照米划分的

1. 获取可以被渲染的GameObj

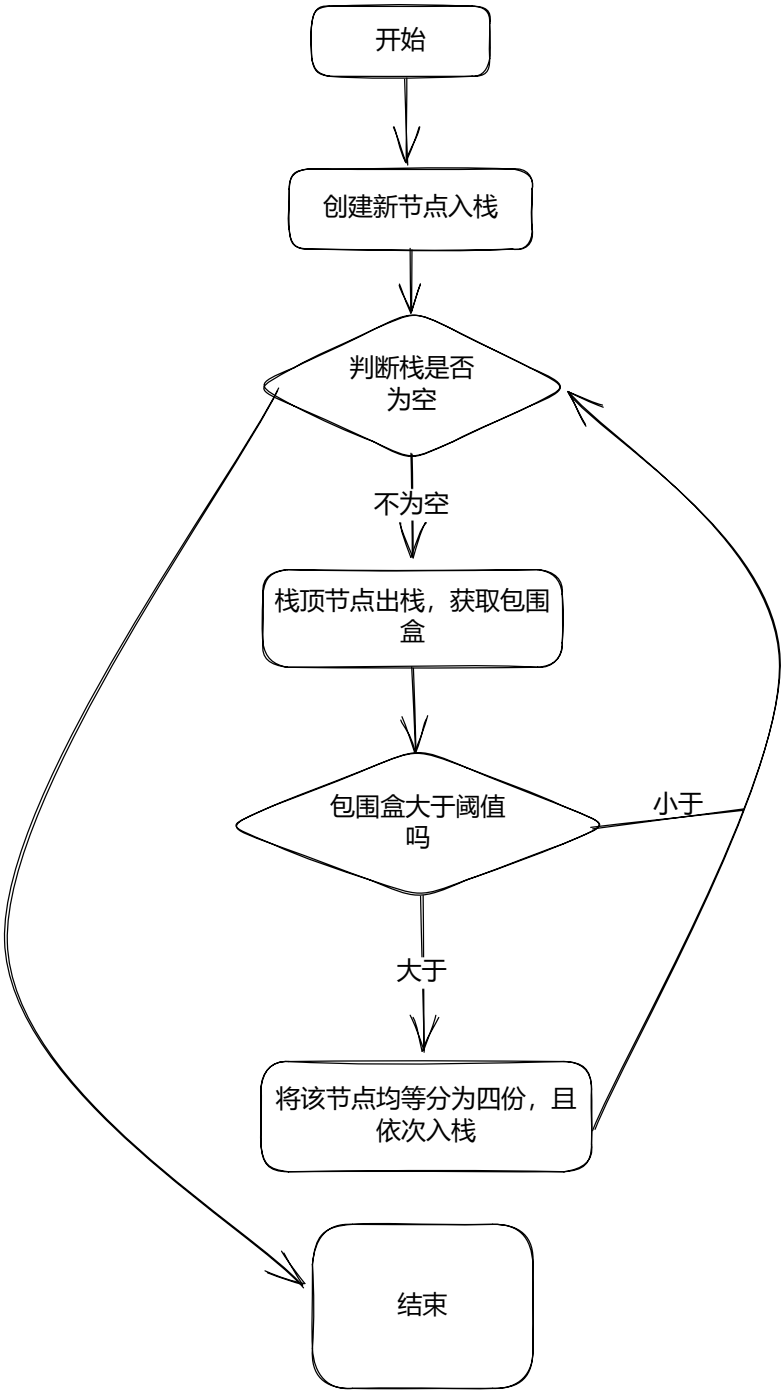
GameObj上如果挂有**HLODMeshSetter**，**LODGroup**，**MeshRenderer**中任何一个，就会被收集到，但是不会重复收集。同时，如果是预制体，只会收集最外层的预制体。

2. 拿掉在HLOD组件设置的切割类型，创建实体，开始划分空间

    1. 获取所有可以被渲染的物体的可以包含它与它子物体的包围盒，放到一个队列中。

    2. 利用栈来对节点进行处理

        1. 划分节点构建四叉树



2. 将每个物体放到四叉树中最适合的位置（**基于距离和包围盒的紧凑程度**）

3. 如果勾选了**UseSubHlodTree**

   将跟节点包围盒划分，然后再重复上面的过程

> 注：一个根节点的划分SubTree不能超过256个

准备简化模型合批的数据

1. 用队列按照层遍历树，将结果保存到**List<TravelQueueItem>**中转站，将子节点的**OBJ**逐层的拷贝到父节点，意味着，每个节点（**中转站，不是真正的树的结构**）一定包含它下面所有节点的物体。

2. 遍历按照层展开成**List**的四叉树

    1. 拿到该节点的所有物体的碰撞体，小于一个规定范围的不要
    2. 遍历这个节点的所有**OBJ**，拿到每个OBj激活的Renderer中，Lod等级最低的那个MeshRenderer
    3. 根据MeshRenderer构建**一个WorkingObj**添加到**HLODBuildInfo**List中。
    构建这个WorkingBoj主要是CopyMesh数据和Texture和Material
    4. 同时将物体的碰撞体也添加到**HLODBuildInfo**List中，其排列顺序也是按照层展开的四叉树

3. 处理初始化好的**HLODBuildInfo**

主要是判断其保存的WorkingObj的数量如果是0，就将其释放，不为0的保存下来

4. 返回一个处理好的**List< HLODBuildInfo>**

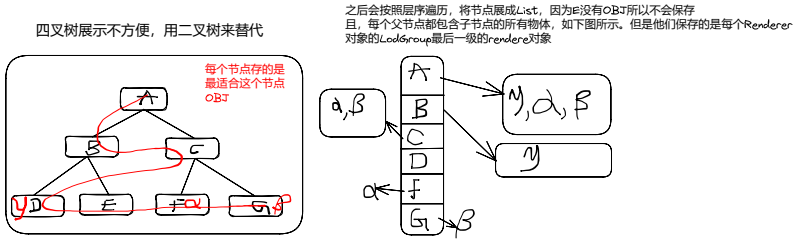
> 关于WorkingObj查看此处

简化模型

如果在HLOD设置为None，则不会触发。可以将Simplygon加进来

合批

提供了两种合批方式
首先看一下经过前几个步骤后现在的场景数据变成什么样了:



MaterialPerservingBatcher(材质保留的合批)

1. 按节点遍历，将一个节点下的OBJ按照材质分别存储
2. 将分好材质Mesh进行合并，合并就是将顶点数据合并到一起，不会合并带骨骼的物体

3. 输出，这时上图右边的列表里的OBJ将是按照材质合并好的WorkingObj

SimpleBatcher

1. 获取该节点下所有物体的贴图，然后合并贴图
2. 合并该节点下的所有Mesh
3. 创建一个新的材质

处理加载方式