# 虚幻引擎三位建模实现流程

## 导入模型数据（蓝图完成）

通过蓝图节点GetSectionfromStaticMesh从资源模型中获取模型数据，我主要获取了模型的顶点坐标，顶点拓扑关系（这决定了顶点是按照什么顺序来组成模型的表面），顶点法向量（这可以决定我们模型每个面的光照强度）

按照在虚幻引擎蓝图与C++交互一节中提到的方法，我在一个蓝图类中添加一个C++组件，（在我的项目中我命名为Halfedge），并且在C++类中设置共有变量，在变量声明之前添加虚幻引擎特有的声明：Uproperty（BlueprintReadWrite），这将使得我的变量可以在蓝图类中可见并且可编辑。通过这样设置，就可以完成蓝图类和C++类中的数据交换。

## 模型数据格式转换

根据之前提到的HalfEdge结构的优点，我需要把从蓝图中得到的模型数据（由数组来存储）转换为Halfedge数据结构的形式。

首先要实现的就是数据类型的转换，Halfedge中实现存储主要是通过标准空间下的容器以及OpenGL图形库的glm数学库，而虚幻引擎有自定义容器。我把每个顶点的坐标单独拿出来然后存入一个glm::vec3类型的变量。同样地，可以实现顶点拓扑结构，顶点法线结构的数据类型转换。

## 数据结构转换

由于虚幻中模型资源只能通过FBX格式导入。FBX模型格式的一个特点是每个物理位置有多个顶点，其中的每个顶点对应着不同的面，以及这个面的法向量。这意味着一个四面体有12个顶点，对应着12个法向量。但是在半边结构中我们在一个物理位置上只需要一个顶点，面的索引和法线将被作为HalfEdge结构的属性存储。

我的顶点转换流程是：依次检查从蓝图中获得的顶点，通过判断位置信息，只保留该位置上出现的第一个顶点，同时更改顶点拓扑中的顶点索引：在检查到该位置上的第一个顶点时，存入我创建的过渡顶点结构，在检查到第二个及以后的重复顶点时，对顶点索引数组不做处理，但将顶点拓扑数组中所有用到该重复顶点的位置替换为第一个顶点的下标，并且保存该面的索引和重复顶点的法线，作为过渡顶点的属性存储起来。

//是否要说明过渡顶点结构

通过上述操作，我就得到了一个过渡顶点数据结构，接下来，我们再将数据从过渡顶点转换为HalfEdge Structure。这个过程相对比上一步要简单，只需按照HalfEdge库提供的接口来添加顶点，然后每三个顶点一组，按照修改后的顶点拓扑数组来添加HalfEdge中的面，接着添加顶点属性中的法线以及法线所在的面索引。

## 4.存储顶点并移动

我们想要完成的三维模型交互是通过一个小球来实现的，当小球与模型表面顶点足够近时（这个距离是由我来定义的），我们把圆球范围内的顶点索引都存储起来方便以后操作。接下来是处理顶点法线的问题，在第二节我们提到了FBX模型格式的顶点对应每个不同的面有不同的法线，虽然经过处理，但这些法线依旧被当成属性存储在顶点结构中。但是我们在移动顶点时必须指定顶点的移动方向，我们选择移动方向，这意味着我们需要把顶点的多个法线合并成一个。我这里只是将顶点的每个法线按照向量相加，然后归一化处理。

这样就完成了三维模型交互的第一步：顶点移动，但只是沿着它的法线方向移动一小段距离，生成的新模型不具有可拓展性（顶点，面的数量都没有发生变化）。

## 5.根据移动顶点生成新的顶点和平面

顶点移动会造成顶点附近的边变长，如果没有新增顶点的话，就会导致模型形状不能发生变化。所以我们需要在被改变顶点周围的边上生成新的顶点，这在HalfEdge Structure中需要新的半边，边，顶点来生成新的平面。当一条边大于某个阈值时，我们在这条边的中心点生成一个新的顶点，并且由这个顶点和原来的顶点生成新的平面，生成平面的时候，由于半边结构的限制，我们必须按照一定的拓扑顺序来生成，默认情况下我们取逆时针，这样生成的平面所包含的半边才能够和原来的半边进行匹配。

显然，当我们拉伸顶点的时候，不会只有一条边发生变化，甚至不会只有一个顶点发生变化，这就涉及到半边结构中数据的有效性和时效性。我们必须保证每次处理的顶点和边都是有效的，因为在上一段中我们删除了一些边和平面，这会对我们模型的结构造成影响。

在上一节中我存储了在圆球半径内的顶点，在这时派上了用场。由于半边结构的便利性，我可以拿到顶点周围的边，并且在保持没有重复的边的情况下把每条边存储起来（因为两个邻接顶点直接会有重复的边，而对于每条边我们只需要处理一次）