

# 基于RflySim3D接口的固定翼静态网格体模型导入实验（仅限完整版及以上版本）

## 1. 实验目的

自定义的飞机模型在3ds

Max中调整，导入UE中验证拼接效果并烘焙，最后与配套XML文件一并导入RflySim3D并展示效果。

## 2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链<sup>[1]</sup>，3ds Max2021；Unreal Engine4.27。
- 硬件要求：笔记本/台式电脑1台<sup>[2]</sup>。

## 3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\3.RflySim3DUE\3.CustExps\e2\\_CusLoadFixWing\1.CusLoadActs](#)

MQ-9Reaper/max：3dsMax：处理的模型数据

MQ-9Reaper/UE4Cooked：烘焙好的数据

MQ-9Reaper/UE4Content：UE中的模型数据

MQ9ReaperTest.slx：Simulink模型

## 4. 实验内容或步骤

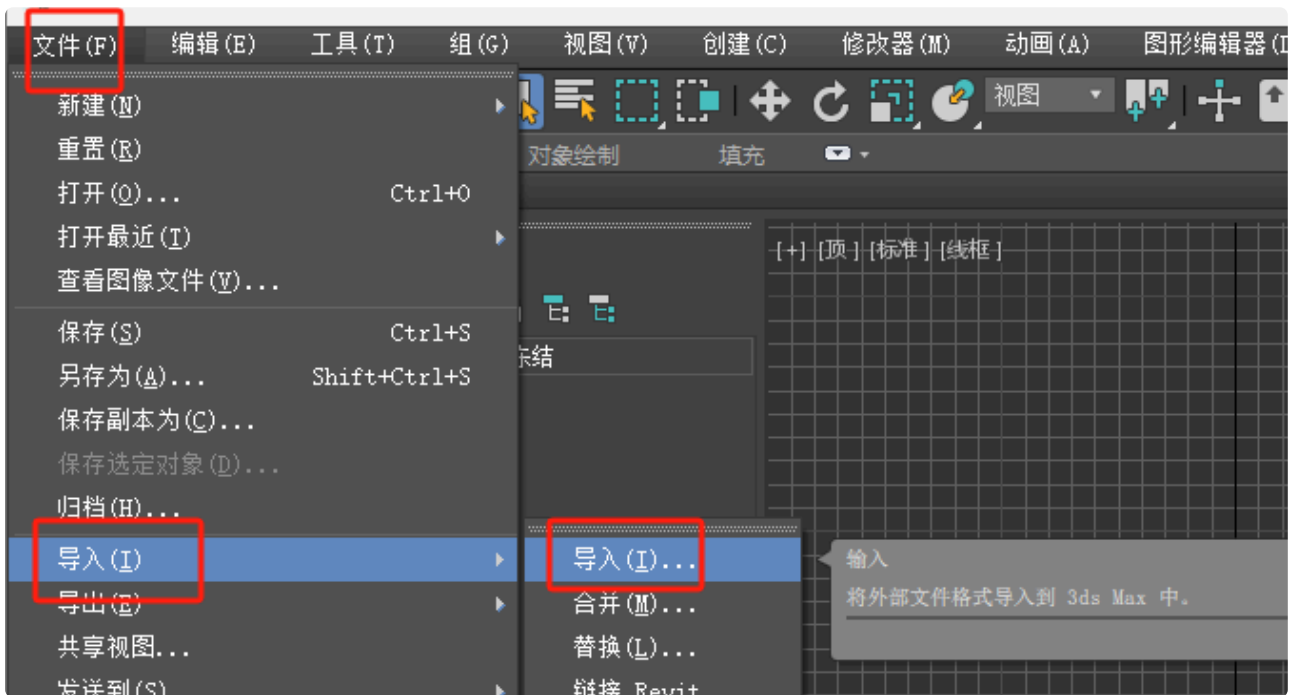
### 自定义固定翼模型加载实验（选做）

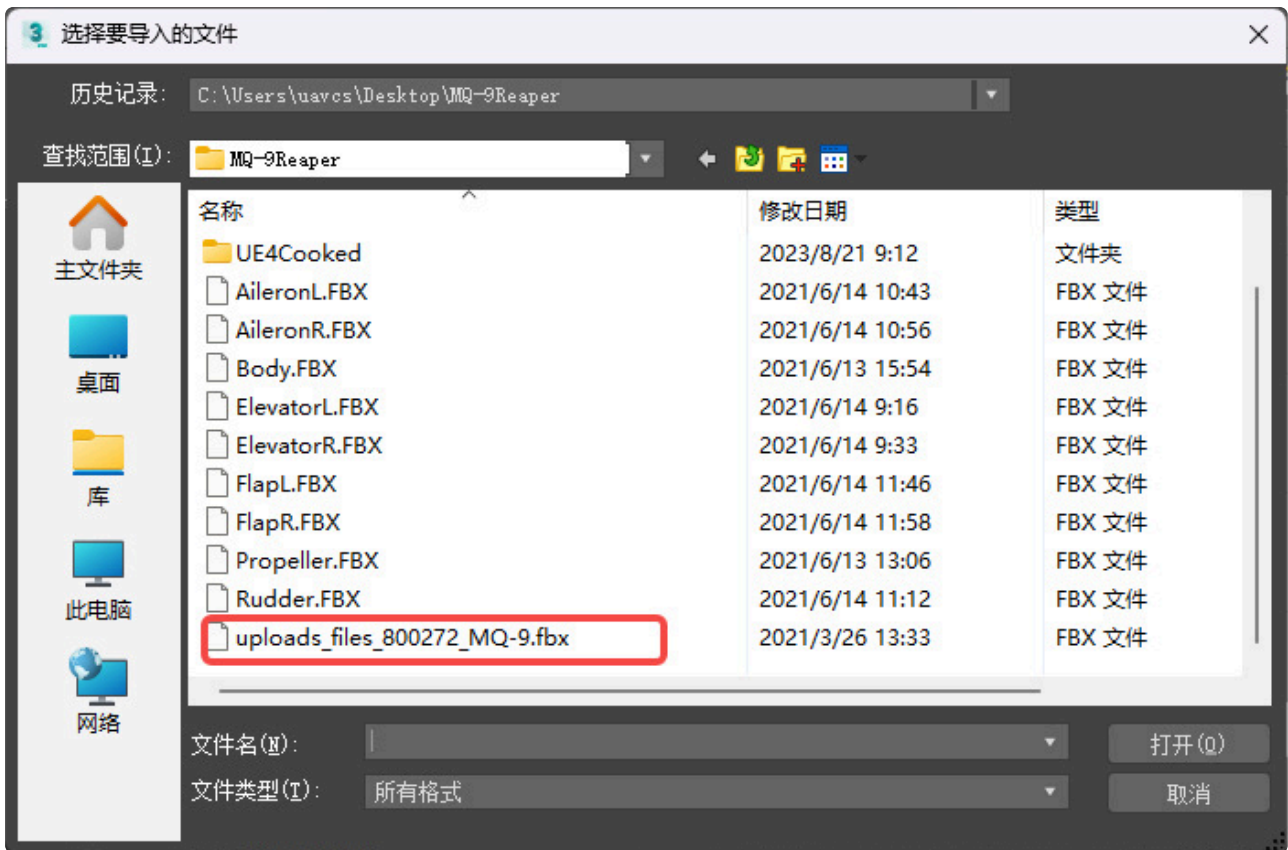
#### 步骤 0: 解压额外文件

解压MQ-9Reaper.zip压缩包到当前目录。

#### 步骤 1: 飞机坐标轴的归零

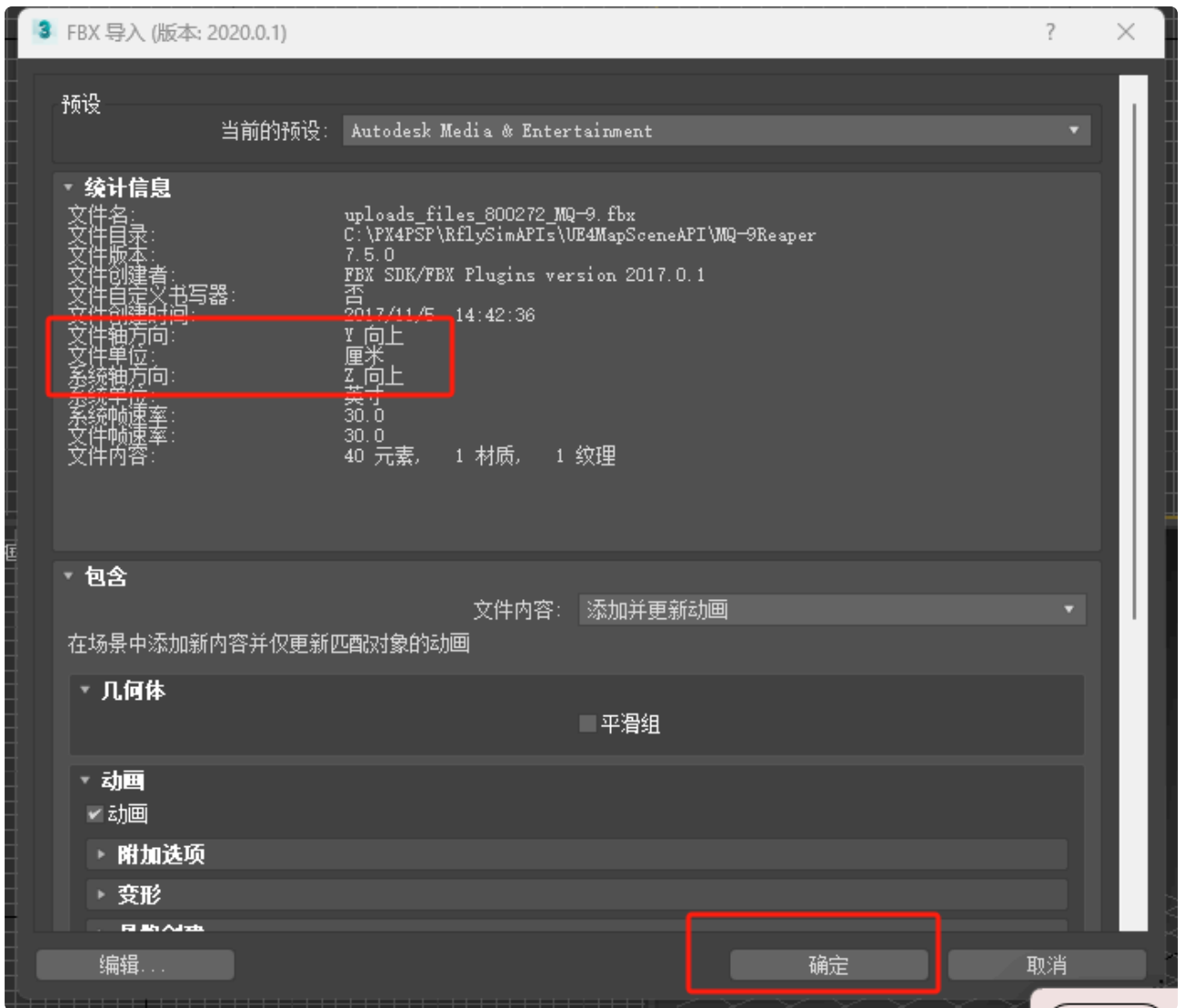
在MQ-9Reaper找到uploads\_files\_800272\_MQ-9.fbx格式的MQ-9无人机三维模型，在3Ds Max中导入。





在FBX导入器上将文件单位转化为厘米

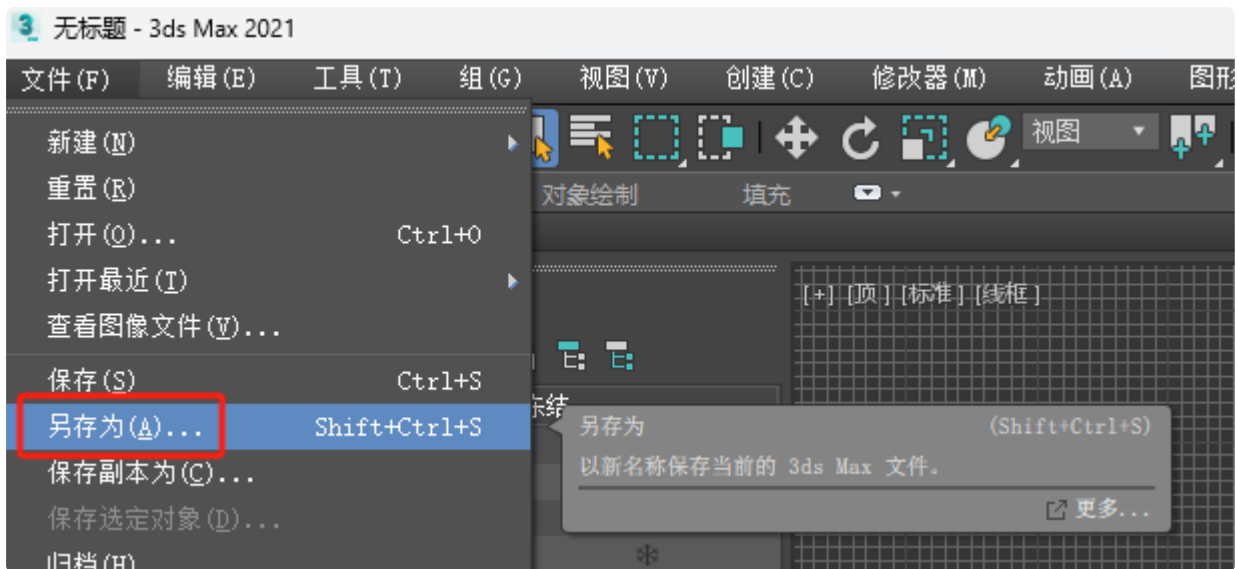




注意:

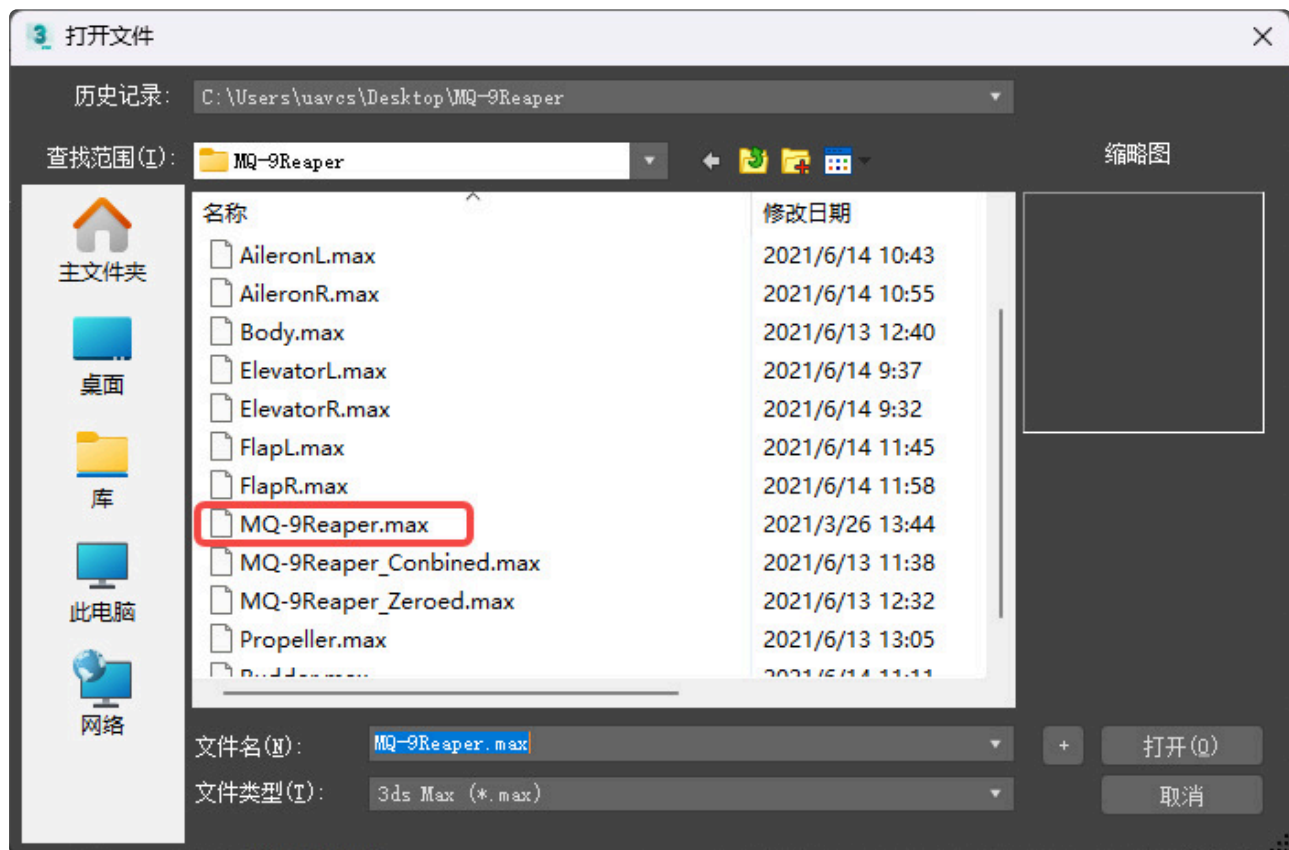
- 单位的选择以源文件为准，避免后序的坐标变换产生尺寸的差异
- 这里FBX 导入器在 UI 上未提供“轴转化”选项，这是因为在导入时 3ds Max FBX plug-in自动计算和应用“轴转化”。导入器仅对传入场景的根元素应用轴转化。

导入完成后，可以另存得到MQ-9Reaper.max的整机三维模型文件。

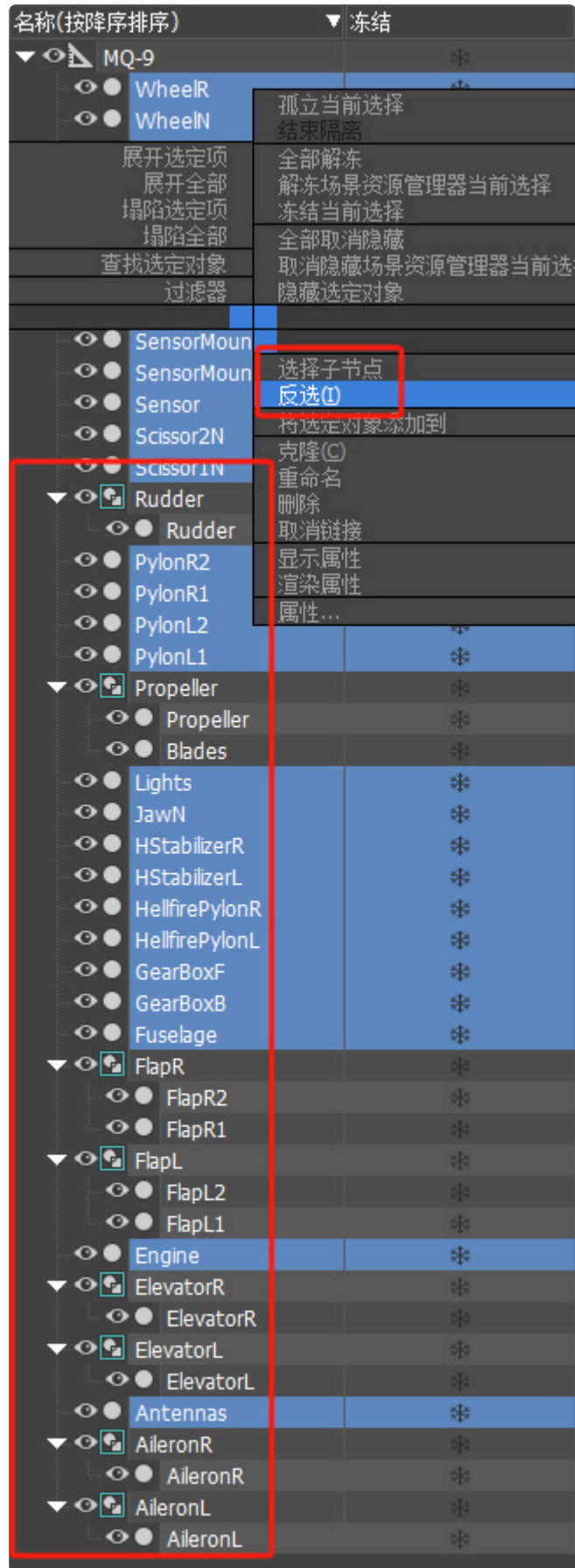


## 1.1 机身和执行器分组

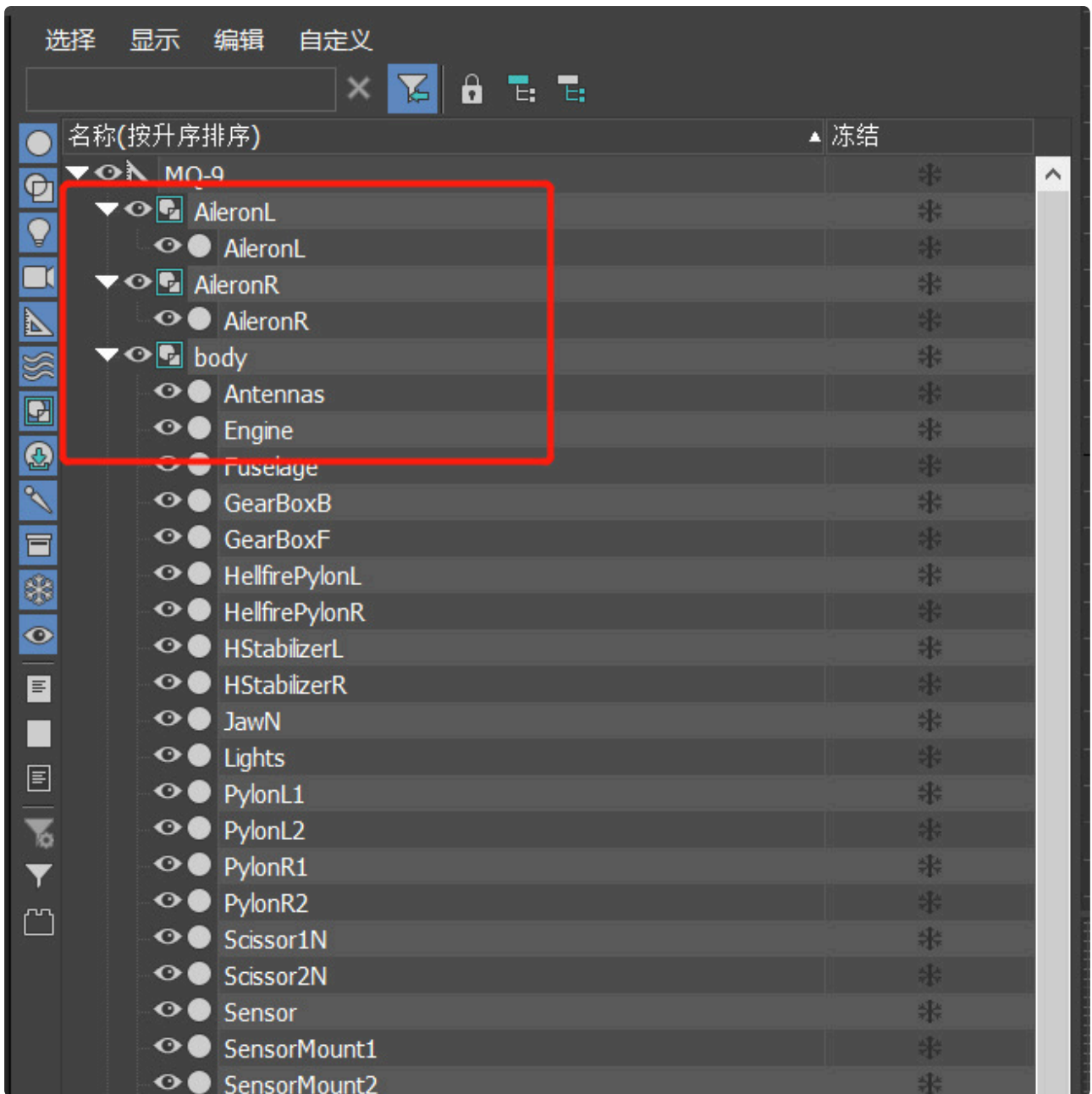
打开MQ-9Reaper.max文件

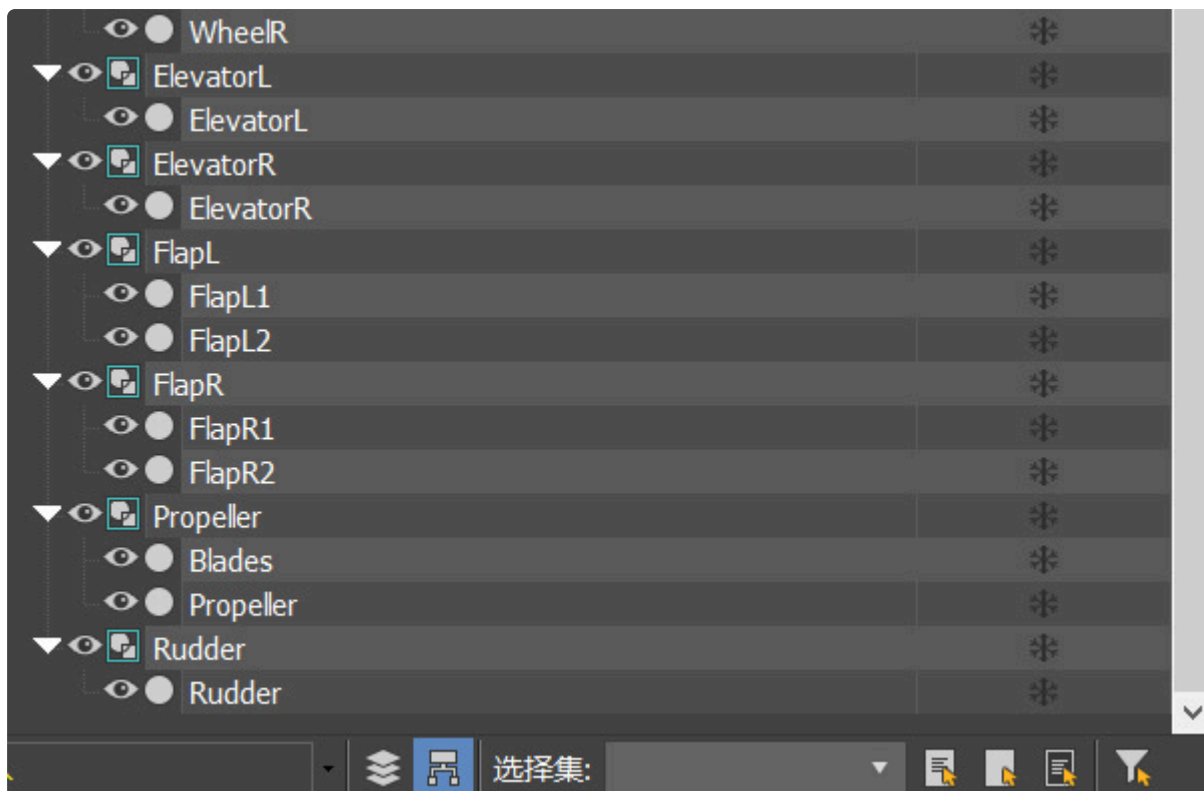


将副翼，螺旋桨等关键部件（8个执行器）选中，然后点击菜单栏上的“组” - “组”进行组合。选中所有执行器组件，然后右键“反选”，这样就选中了所有机身的组件。注意机身组件中不能包含场景根节点MQ\_9



除八个执行器：1. Propeller（转动）、2. ElevatorL（偏转）、3. ElevatorR（偏转）、4. AileronL（偏转）、5. AileronR（偏转）、6. Rudder（偏转）、7. FlapL（偏转）、8. FlapR（偏转），外，其余部件归入机身。





如上图所示，这里分为了9个子物体，包含body（机身），和8个执行器（注：即使只有一个物体，也要将其进行“组”操作，变为一个组，使得物体中心归零），另存得到文件名“MQ-9Reaper\_Conbined.max”

分组得到的名字为：

AileronL 左副翼

AileronR 右副翼

Body 机身

ElevatorL 左升降舵

ElevatorR 右升降舵

FlapL 左襟翼

FlapR 右襟翼

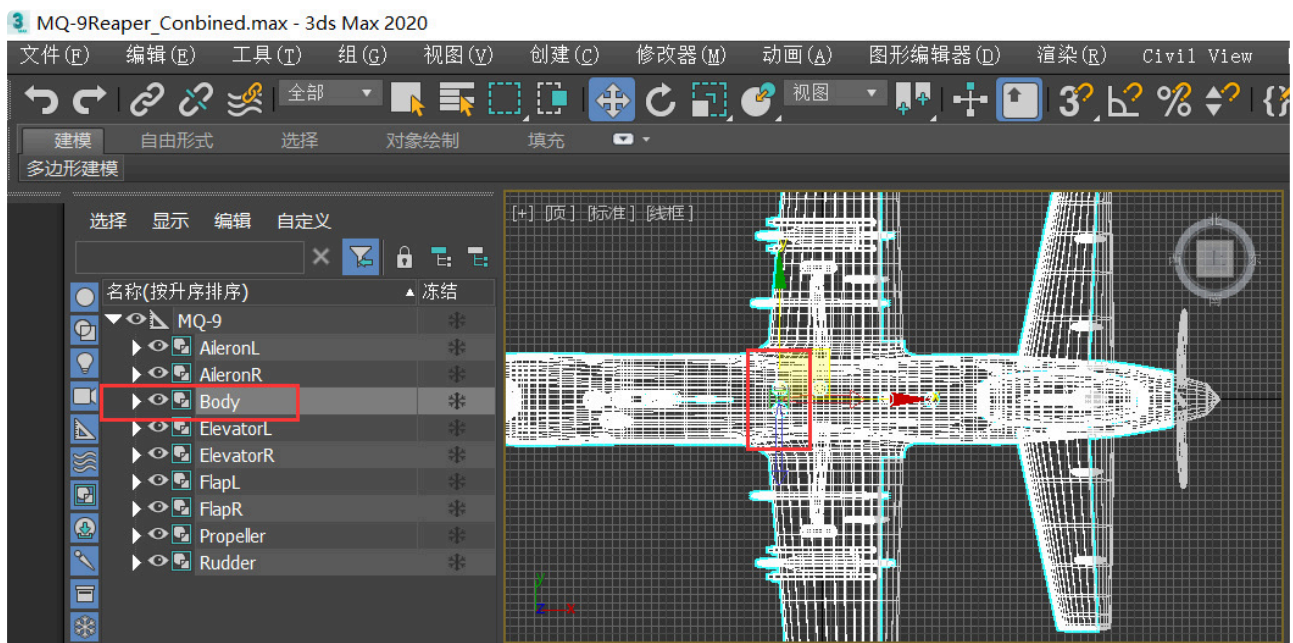
Propeller 螺旋桨

Rudder 方向舵

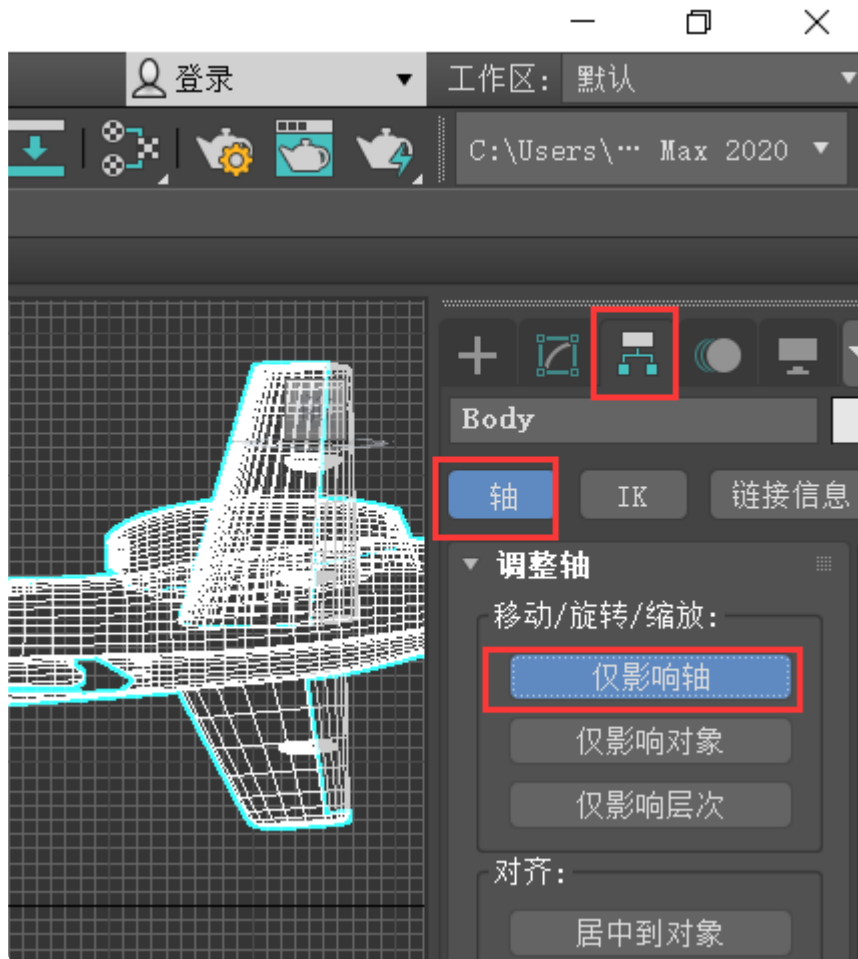
注意：XML中八个执行器（和运动方式）的定义顺序为：1. Propeller（转动）、2. ElevatorL（偏转）、3. ElevatorR（偏转）、4. AileronL（偏转）、5. AileronR（偏转）、6. Rudder（偏转）、7. FlapL（偏转）、8. FlapR（偏转）

## 1.2 确定飞机重心位置

打开“MQ-9Reaper\_Conbined.max”，然后，选中机体Body的物体，可以看到机体的中心轴并未在期望位置（一般在机翼中心附近），因此需要将机体坐标轴移动到期望位置。



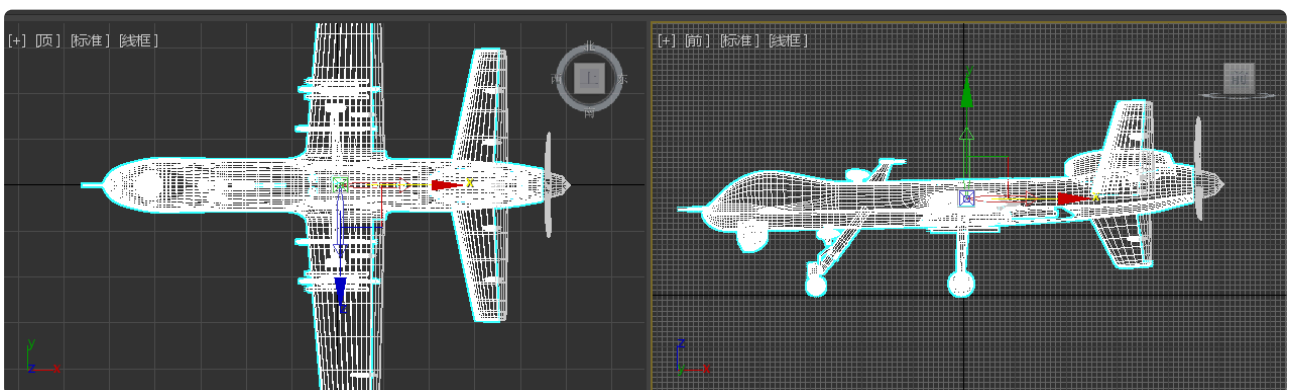
点击右侧工具栏的第三项“层次” - “轴” - “仅影响轴”，然后可以拖动机体的中心轴。



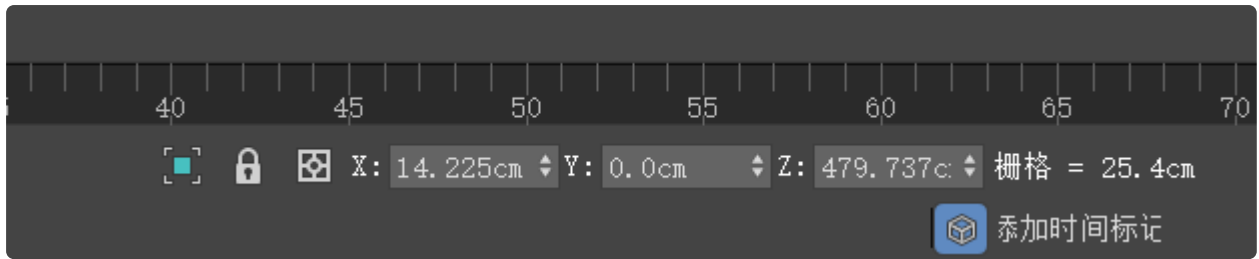
同时点击工具栏的移动图标



用鼠标，手动将坐标轴移动到期望的位置（固定翼一般在机翼中心附近，多旋翼一般将中心置于机架圆心），其他的人物和车辆之类可以将中心置于物体几何中心（或最底端的中心）。在RflySim3D中观察飞机的欧拉姿态角将会以本轴心为基准。



然后，从状态栏中读取当前坐标中心的位置

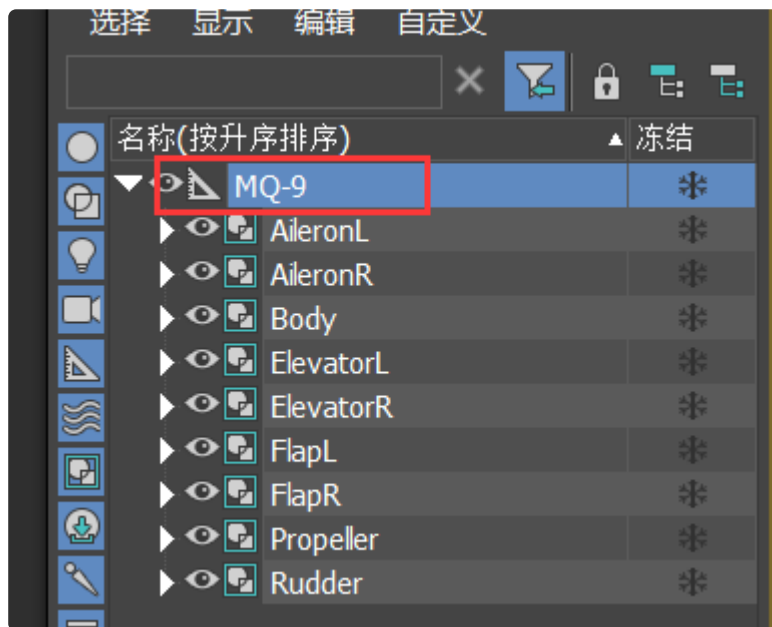


例如，这里为[14.225,0,479.737]

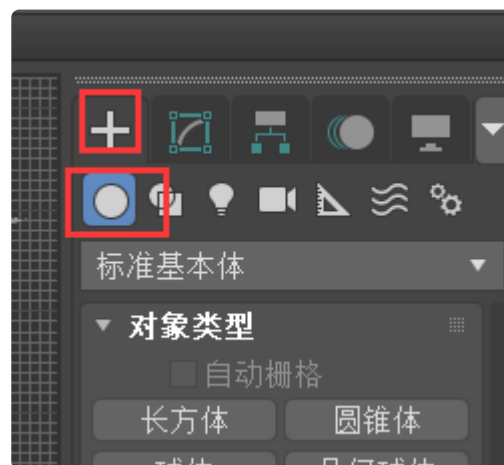
### 1.3 将飞机位置挪动到场景中心

打开MQ-9Reaper\_Combined.max文件并另存为MQ-9Reaper\_Zeroed.max。

选中顶层主物体“MQ-9”



右侧工具栏进入第一个“创建”模式



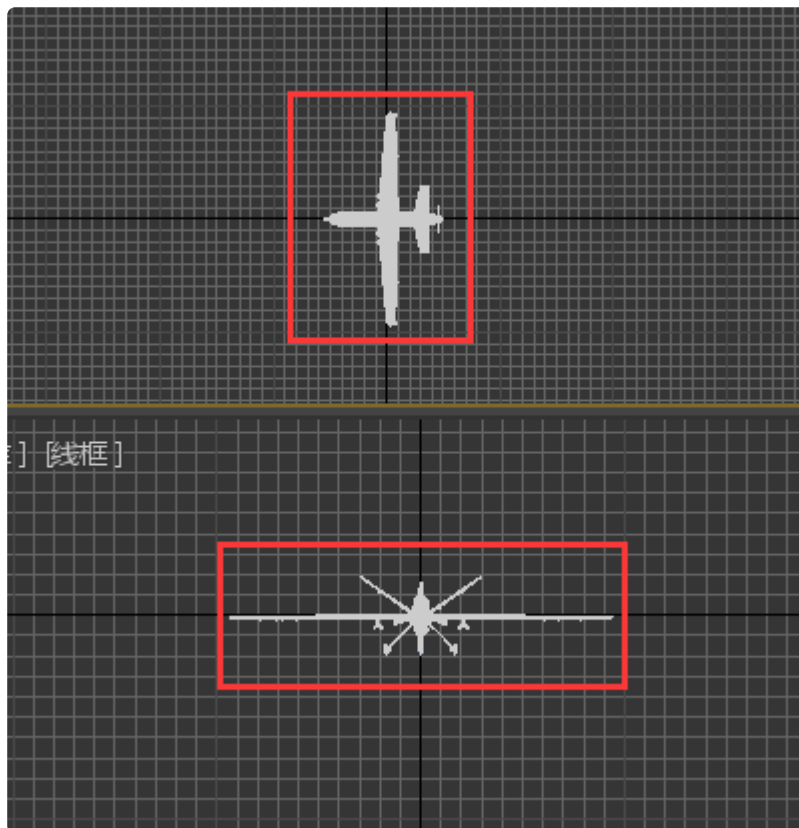
点击菜单栏移动按钮



在下侧状态栏，输入-14.225,0,-479.737即可将物体整体移动



可见，如下图所示，飞机中心已经置于场景中心（黑色粗线形成的坐标轴）

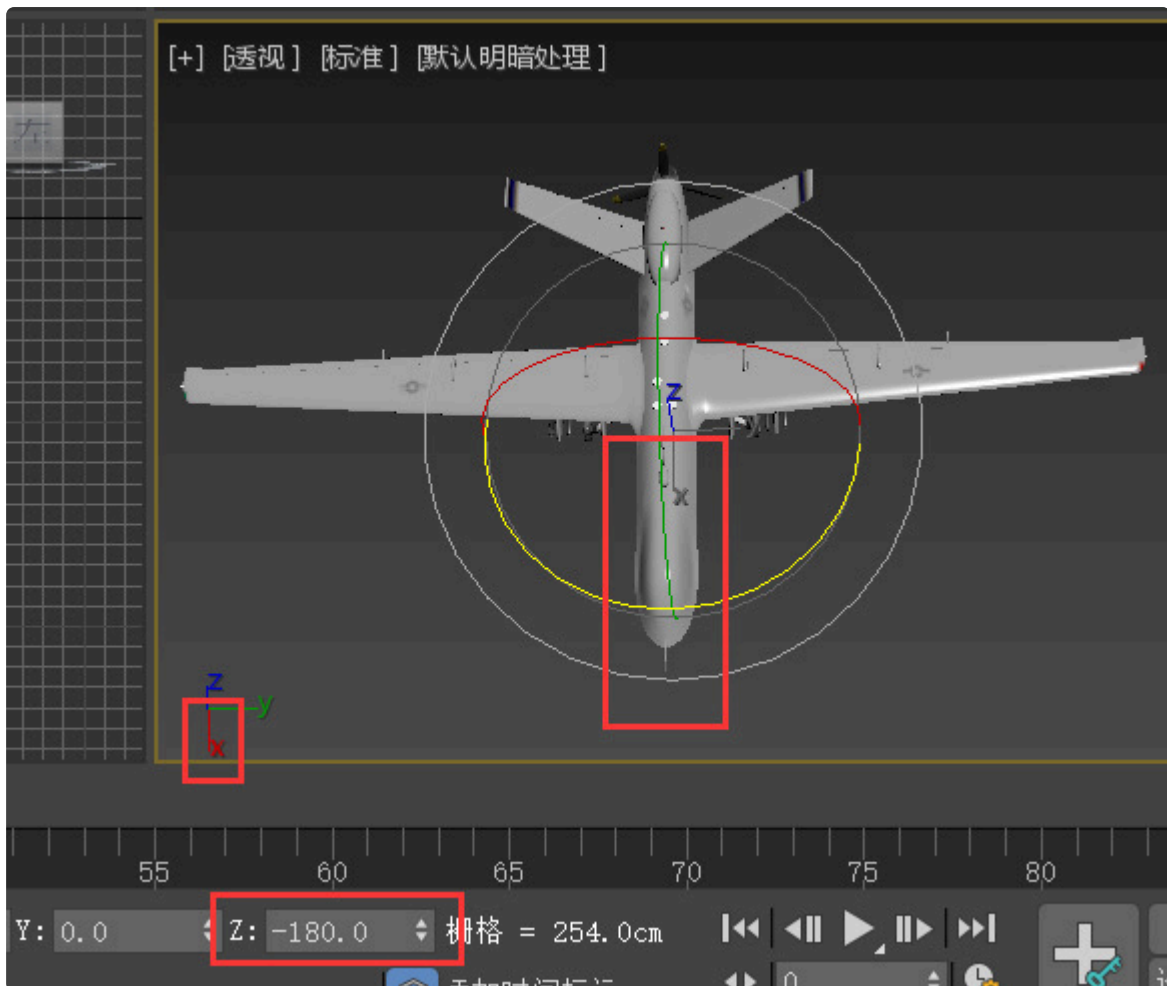


此时，还发现目前飞机的X轴与机头方向相反，因此需要将飞机沿Z轴旋转180度。

选中“MQ-9”飞机整体，点击工具栏的“旋转”工具



如下图所示，在Z轴处输入-180，就能实现机头方向沿Z轴翻转



至此，飞机的中心和方向已调整到位，点击保存。

根据上述信息，可以得到XML文件中飞机整体信息为

```
<ClassID>100</ClassID>
```

```
<DisplayOrder>900</DisplayOrder>
```

```
<Name>MQ-9Reaper</Name>
```

```
<Scale>
```

```
<x>1</x>
```

```
<y>1</y>
```

```
<z>1</z>
```

```
</Scale>
```

```
<AngEulerDeg>
```

```
<roll>0</roll>
```

<pitch>0</pitch>

<yaw>0</yaw>

</AngEulerDeg>

注：ClassID取100表示固定翼，DisplayOrder取900表示在固定翼样式中排名最靠前（平台中默认的两个固定翼分别为1000和1100），由于没有放缩飞机Scale都选1倍，由于飞机x轴已经配置好（朝向机头），不需要在UE4中旋转，AngEulerDeg均设为0.

注意：以上尺寸对应了真实的大型无人机，长度可达25m左右。我们一般采用缩小版的模型机，因此可以将上述Scale缩小10倍，取0.1。也就得到如下结果

<Scale>

<x>0.1</x>

<y>0.1</y>

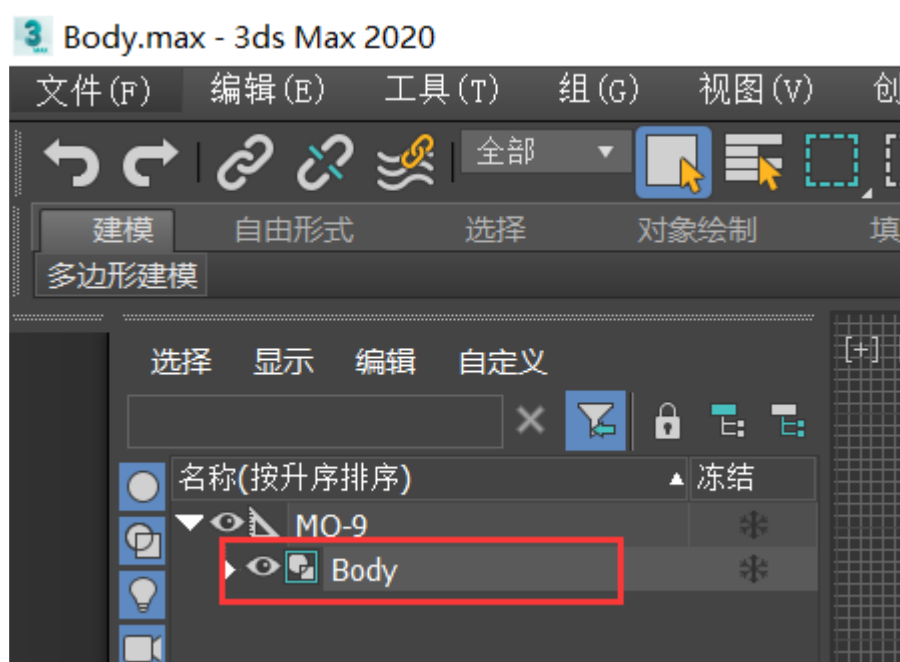
<z>0.1</z>

</Scale>

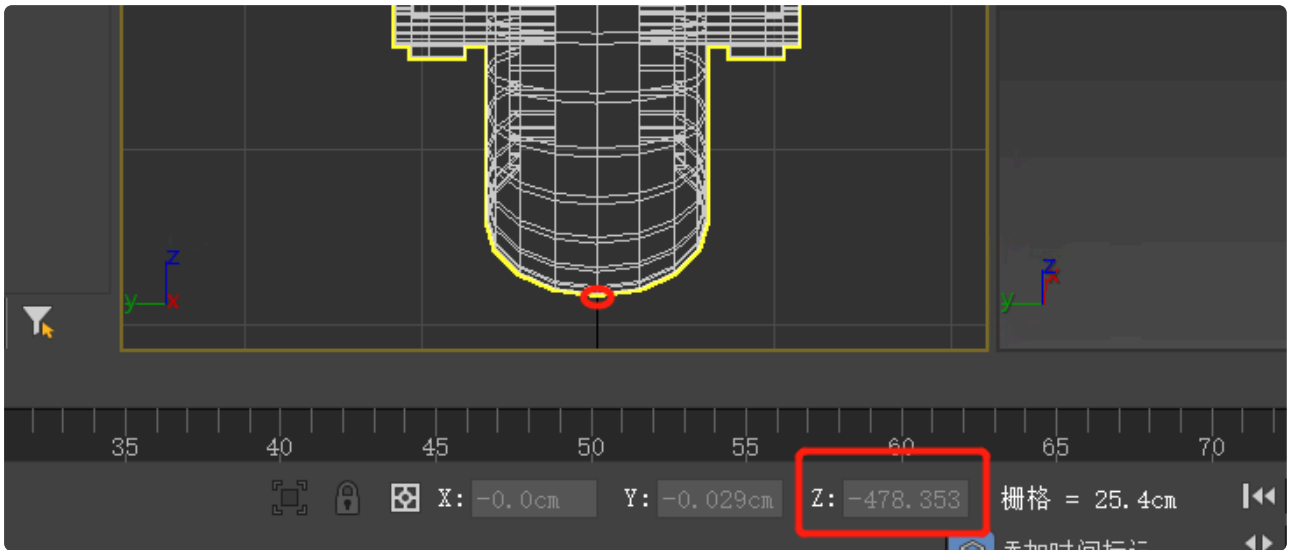
## 步骤2：机体模型导出

打开MQ-9Reaper-Zeroed.max，另存为名字Body.max

删除其他的模块，只保留机体“Body”组件



进入移动模式，将鼠标悬浮（不用点击）置于飞机最下沿（贴近地面的地方），可以在坐标栏读取此时鼠标位置，记录此时的z坐标值。例如，下图的z值为-478.355cm，表示飞机中心距离地面的高度为4.784m



直接导出成 “Body.FBX”





这里机体的XML项目可以表示为：

```
<body>
```

```
<ModelType>0</ModelType>
```

```
<MeshPath>/Game/MQ-9Reaper/Body </MeshPath>
```

```
<MaterialPath></MaterialPath>
```

```
<AnimationPath></AnimationPath>
```

```
<CenterHeightAboveGroundCm>478</CenterHeightAboveGroundCm>
```

```
<NumberHeightAboveCenterCm>700</NumberHeightAboveCenterCm>
```

```
<NumberSizeScale>50</NumberSizeScale>
```

```
</body>
```

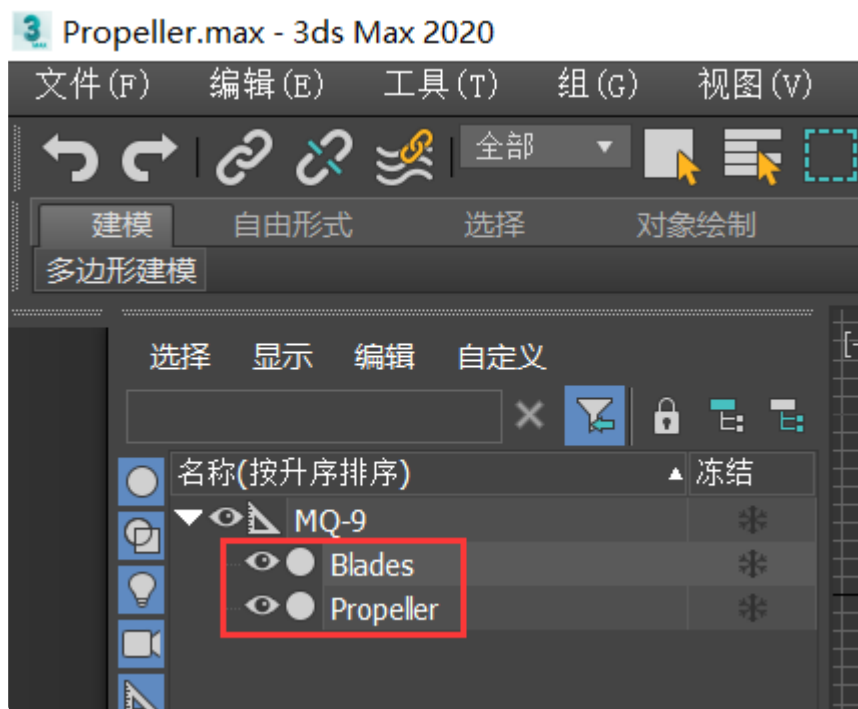
其中，ModelType取0，因为本模型没有动画，与之对应的AnimationPath给空即可。MeshPath需要根据实际导入的Content目录的位置来设定。MaterialPath是上述网格的材质路径，一般给空即可，会使用UE4中配置的材质，这里也可以指定其他样式从而改变飞机的颜色等属性。CenterHeightAboveGroundCm选刚才记录的中心距离地面的高度值478cm；NumberHeightAboveCenterCm是飞机数字标号的显示位置，一般在飞机顶上更高位置，这里给700；NumberSizeScale表示数字标号的（相对默认的F450飞机的比例）显示尺寸，一般根据飞机实际尺寸来，这里取50倍，可根据实际显示效果来调整倍数。

## 步骤3：螺旋桨模型的导出

### 第3.1步，提取螺旋桨模型

打开MQ-9Reaper-Zeroed.max，另存为名字Propeller.max

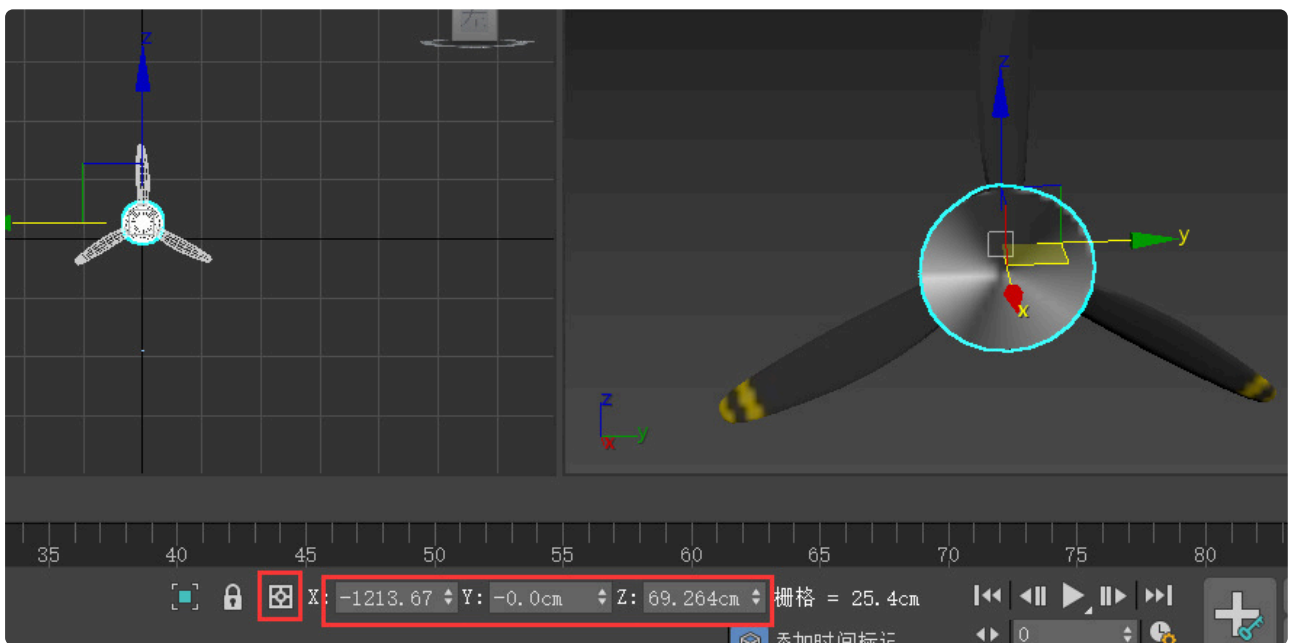
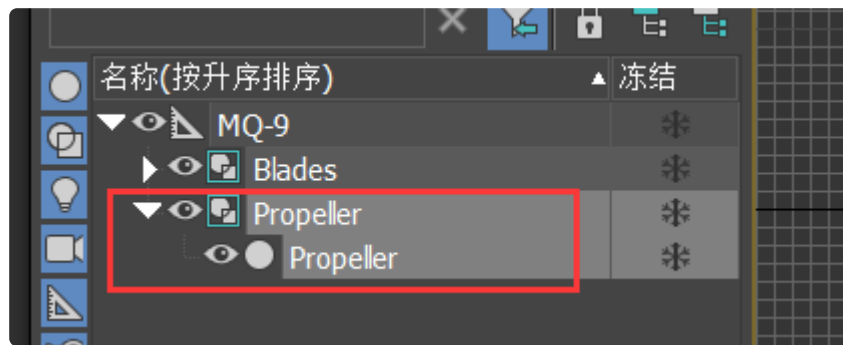
删除其他的模块，只保留机体“Propeller”组件，并将其解组为Blades和Propeller



然后，如下图将两个子组件单独组成组，确保物体轴心位于物体中心



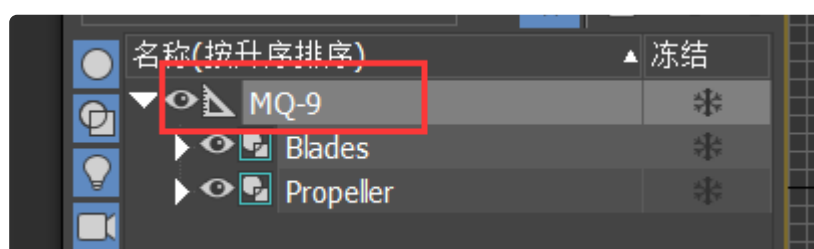
点击工具栏“移动”按钮，选中Propeller组合体，在下侧的坐标栏读取螺旋桨坐标



读数为-1213.674 0 69.264

### 步骤3.2：调整轴心对齐

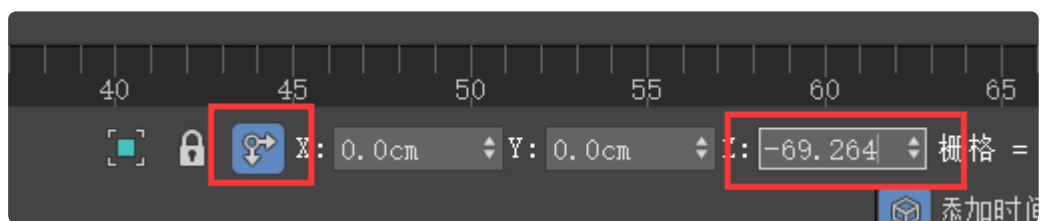
全选整体“MQ-9”，对上述坐标取反可知，需要将整体沿X移动1213.674，以及沿Z移动-69.264来保证螺旋桨移动到场景中心。



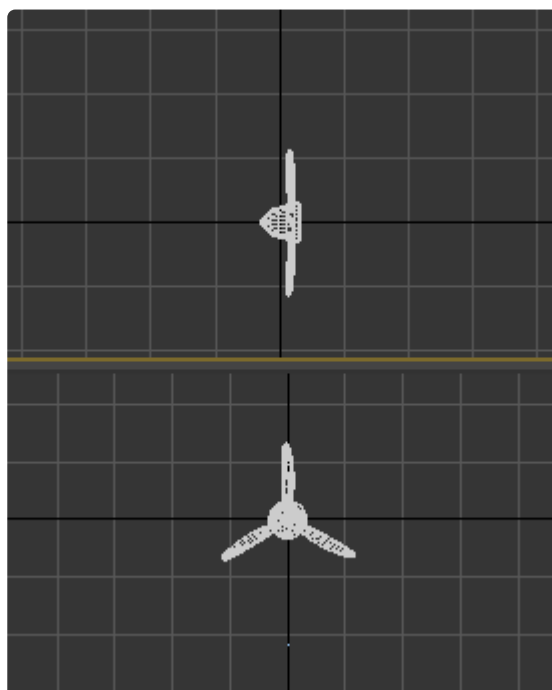
点击移动按钮，并在坐标输入栏左侧选择相对移动图标，在X轴输入1213.674，表示向X轴移动



同理在Z轴输入-69.264



可以看到，此时螺旋桨已经位于场景中心（黑色坐标轴）



同时，我们也要记录下，导入UE4后要将螺旋桨要挪回原位置，才能装配到飞机尾部，因此螺旋桨的安装坐标就是原来记录的位置，这里是【-1213.674 0 69.264】

注意，UE4的坐标y轴向右为正，与3Ds Max的y轴方向相反，因此y轴坐标需要取反。这里y为0，因此不需要操作。

**步骤3.3：保存本max文件，并导出为Propeller.fbx。同时记录螺旋桨的中心摆放坐标值为[-1213.674 0 69.264]，旋转轴为x轴[1,0,0]**

```
<Actuator>

<MeshPath>/Game/MQ-9Reaper/Propeller</MeshPath>

<MaterialPath></MaterialPath>

<RelativePosToBodyCm>

<x>-1213.674</x>

<y> 0</y>

<z>69.264</z>

</RelativePosToBodyCm>

<RelativeAngEulerToBodyDeg>

<roll>0</roll>

<pitch>0</pitch>

<yaw>0</yaw>

</RelativeAngEulerToBodyDeg>

<RotationAxisVectorToBody>

<x>1</x>

<y>0</y>

<z>0</z>

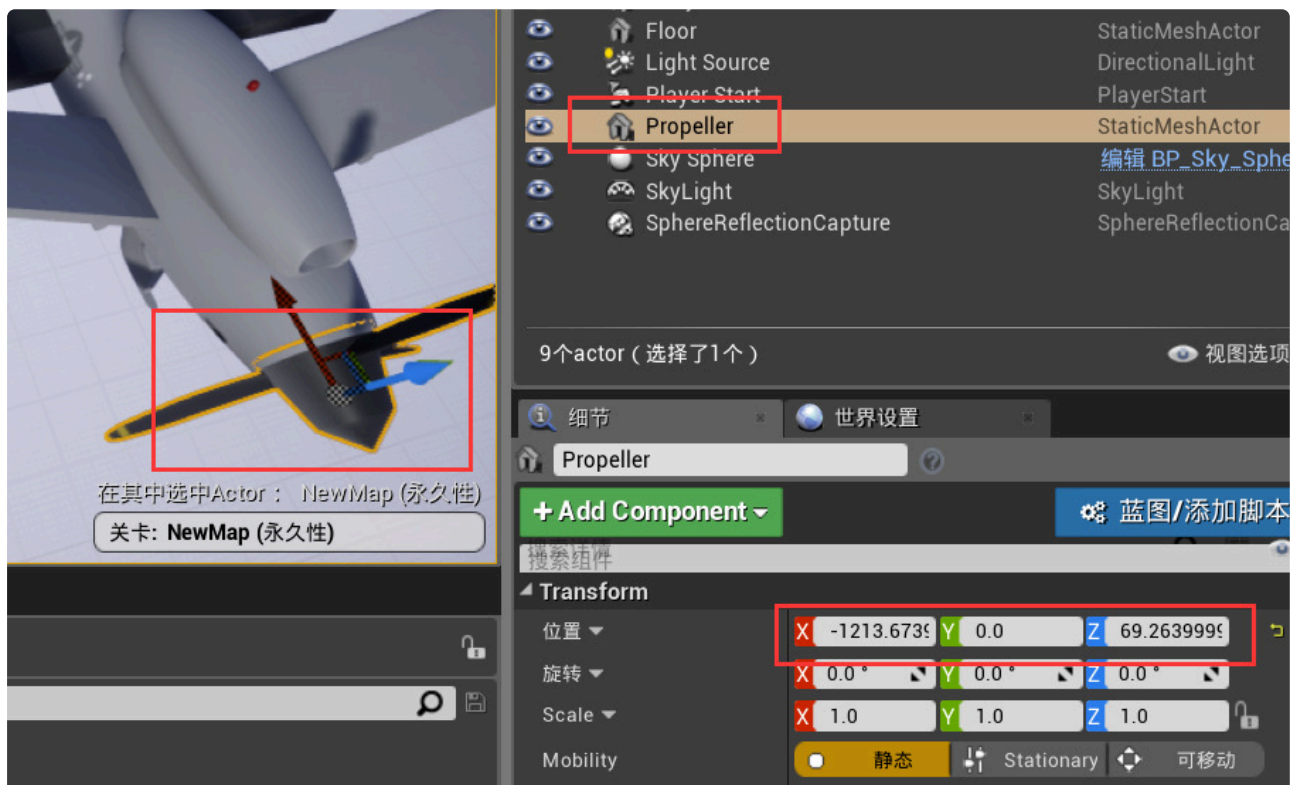
</RotationAxisVectorToBody>

<RotationModeSpinOrDefect>0</RotationModeSpinOrDefect>
```

</Actor>

注意：MeshPath是螺旋桨导入UE4后的存放文件名；MaterialPath为材质文件名，一般给空即可。RelativePosToBodyCm是刚才记录的螺旋桨中心在UE4坐标系中的坐标位置；RelativeAngEulerToBodyDeg是执行器需要旋转的角度，这里我们在3Ds Max已经调整好角度，因此都给0即可。RotationAxisVectorToBody选x轴即[1 0 0]转动。RotationModeSpinOrDefect选0，即表示绕轴旋转的执行器种类，PWM对应单位为转每分。

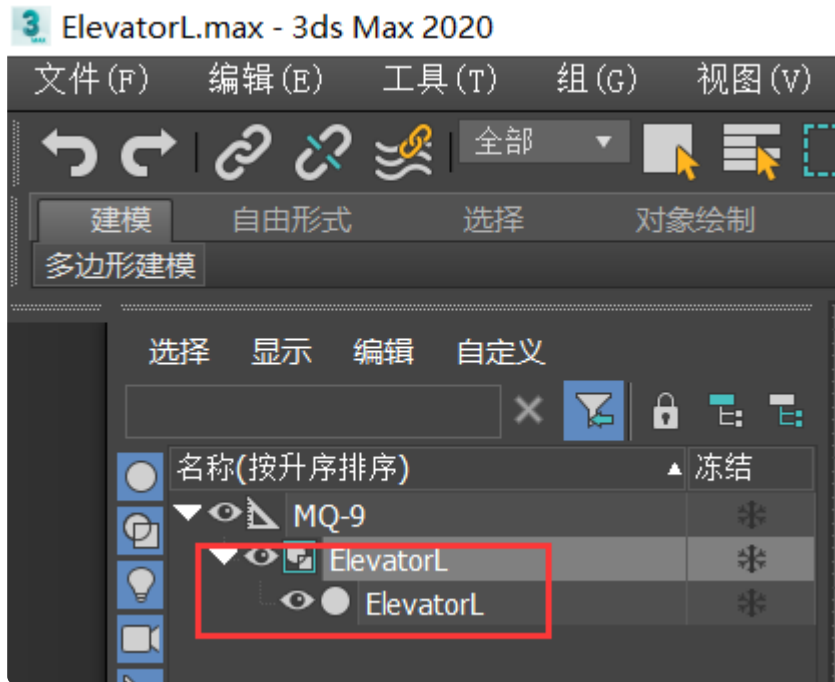
注：将机身和螺旋桨都导入UE4场景中，并拖入关卡，将上述位置的值输入Propeller的Transform位置框，可以预览相对位置是否正确。



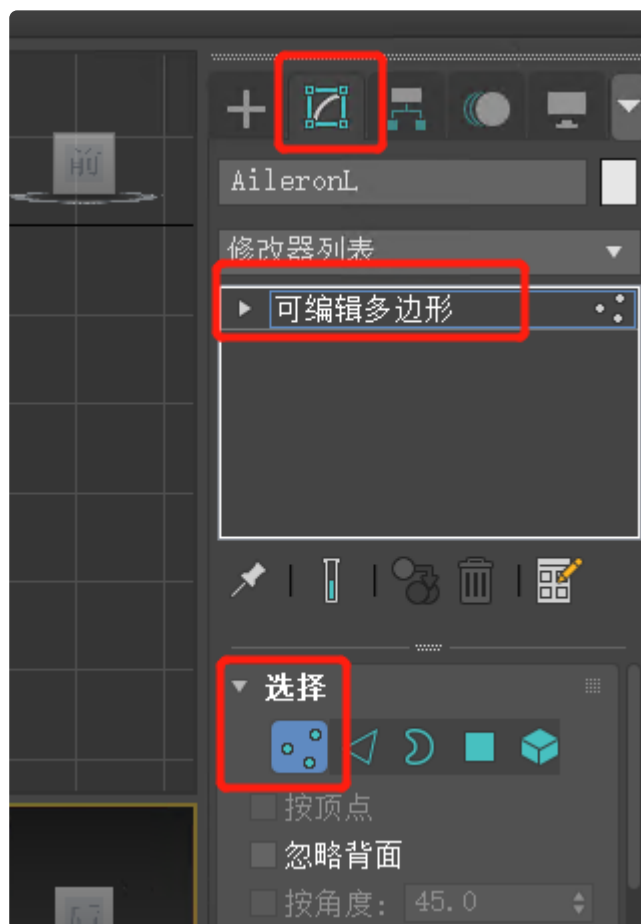
## 步骤4: ElevatorL左升降舵的导出

打开MQ-9Reaper-Zeroed.max，另存一个文件为ElevatorL.max

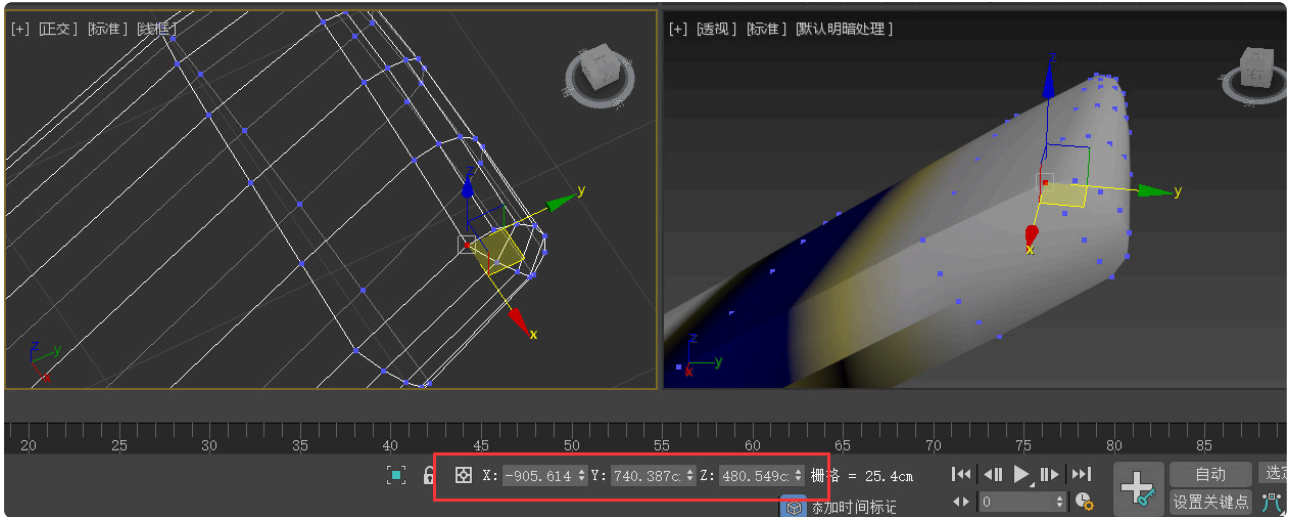
删掉其他物体，只保留“ElevatorL”，并选中它



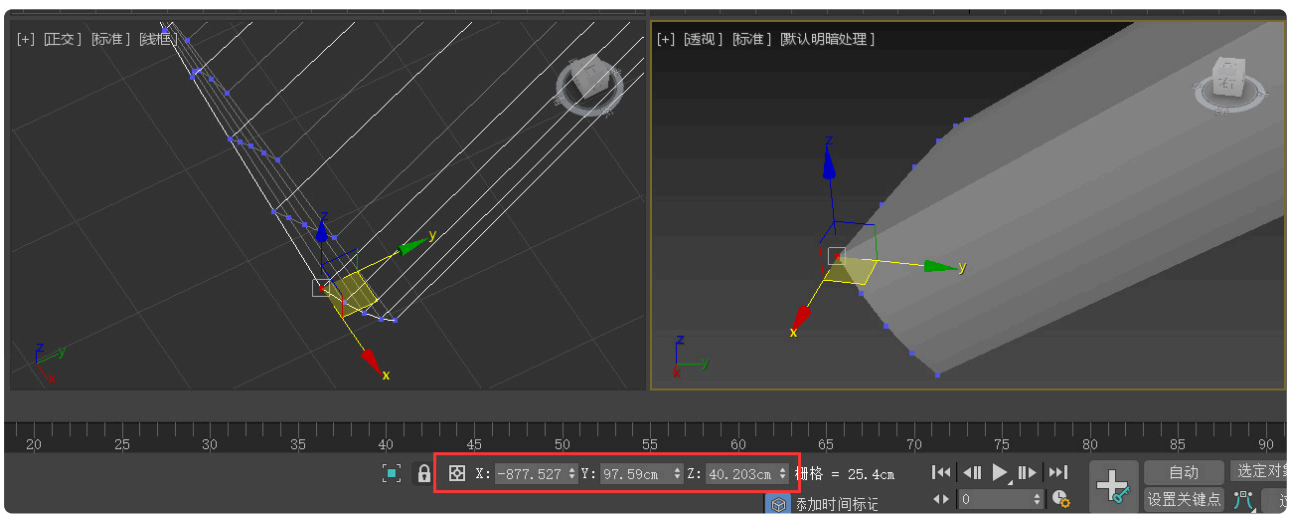
点右侧第二个工具条，可编辑多边形，点，进入点选择模式



依次，选择升降舵上前沿直线的左右两个顶点（作为转动的旋转轴），在移动模式下，可以在状态栏读取这个点做坐标



上图为左上前沿顶点，得到坐标为【-905.614cm，740.387cm，480.549cm】



上图为右上前沿顶点，得到坐标为【-877.527cm，97.59cm，40.203cm】

上述两个点求平均可得到期望中心转轴中点坐标[-891.5705 418.9885 260.3760]，两个向量做差可得转轴的方向向量[28.0870 -642.7970 -440.3460]，此方向向量归一化可得：[0.0360 -0.8245 -0.5648]（注：设D为方向向量，MATLAB归一化代码：D./norm(D)）

## 4.2 调整中心与轴向

我们需要将物体逆向移动，使得刚才的中心点移动到3Ds Max场景的原点，移动的距离为[-891.5705 418.9885 260.3760]取反，因此为X:891.5705, Y:-418.9885,Z:-260.3760]

选中ElevatorL组合，进入移动模式，点击进入相对移动模式



依次，向X轴移动891.5705cm



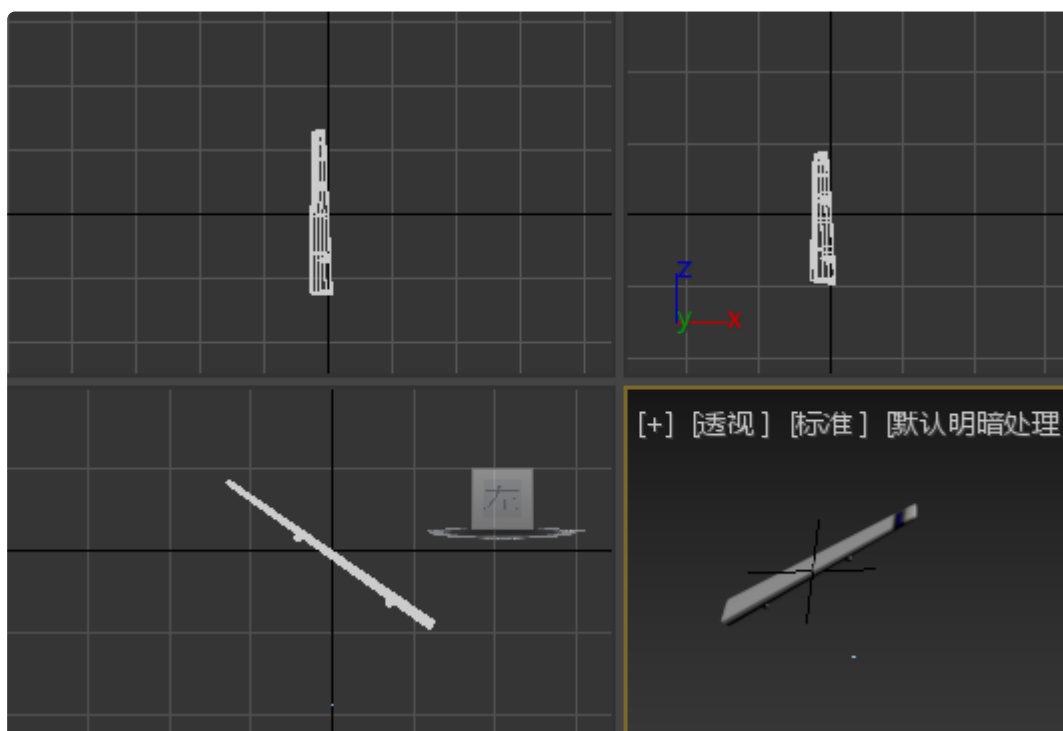
向Y移动-418.9885



向Z移动-260.3760



此时，可以看到升降舵转轴中心已经置于场景中心（黑色粗线形成的坐标轴）



此时，可以保存本ElevatorL.max文件，并按之前同样方法，导出为ElevatorL.FBX，再导入UE4编辑器中。

### 4.3 XML文件编写

由于UE4的y轴与3Ds

Max相反，需要对y坐标进行取反操作，得到最终的轴心坐标为：[-891.5705 -418.9885 260.3760]，旋转轴向为：[0.0360 0.8245 -0.5648]。

将上述中心与旋转轴填入，于是可以得到下面所示的XML代码

```
<Actuator>
<MeshPath>/Game/MQ-9Reaper/ElevatorL</MeshPath>
<MaterialPath></MaterialPath>
<RelativePosToBodyCm>
<x>-891.5705</x>
<y> -418.9885</y>
<z>260.3760</z>
</RelativePosToBodyCm>
<RelativeAngEulerToBodyDeg>
<roll>0</roll>
<pitch>0</pitch>
<yaw>0</yaw>
</RelativeAngEulerToBodyDeg>
<RotationAxisVectorToBody>
<x>0.0360</x>
<y>0.8245</y>
<z>-0.5648</z>
</RotationAxisVectorToBody>
```

<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>

</Actuator>

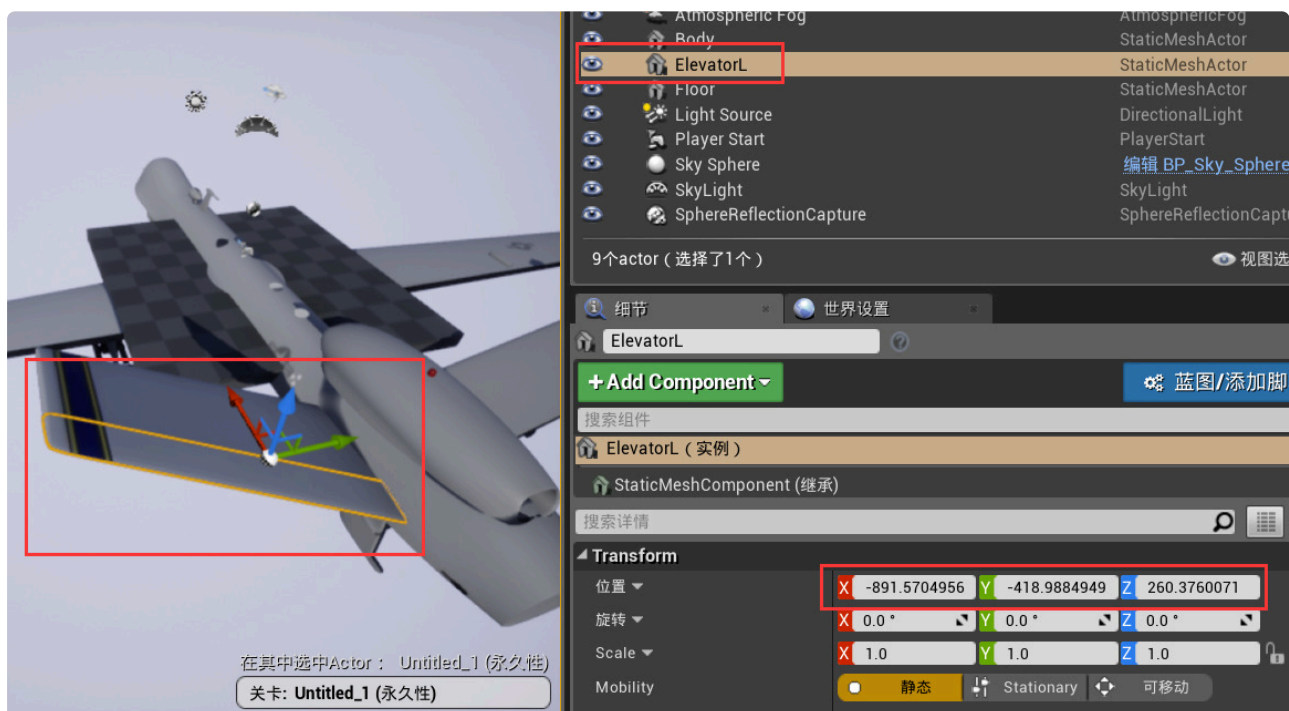
注意：MeshPath是左升降舵导入UE4后的存放文件名；MaterialPath为材质文件名，一般给空即可。RelativePosToBodyCm是刚才记录的左副翼中心在UE4坐标系中的坐标位置[-889.4375,-425.4935,251.7450]；RelativeAngEulerToBodyDeg是执行器需要旋转的角度，取[0,0,0]即可。RotationAxisVectorToBody选升降舵的旋转轴，[-0.0360 -0.8245

0.5648]（注意，可能需要根据实际情况进行反向，例如向前推升降舵杆，升降舵信号增加，升降舵向下偏转，飞机低头，因此升降舵轴向应该朝向左侧）。

RotationModeSpinOrDefect选1，即表示绕轴偏转的执行器种类，PWM对应单位为度。

#### 步骤4.4：导入UE4，拖入关卡中，输入位置坐标，查看拼合效果如下

将机身与左升降舵导入UE4并拖入场景中，将上述位置和欧拉角输入，可以查看两者是否匹配。



#### 步骤5：右升降舵的导入

打开MQ-9Reaper-Zeroed.max，另存一个文件为ElevatorR.max

删掉其他物体，只保留“ElevatorR”，并选中它，开始下面操作

同样的方法可以得到右升降舵的三维文件和XML文件。关键结果记录如下：

## 步骤5.1: 获取转轴信息

在3Ds Max中读取得到左上前沿端点: [-877.527, -97.591, 40.203]

右上前沿端点: [-905.614, -740.387, 480.549]

中点: [-891.5705 -418.9890 260.3760]

方向向量: [-28.0870 -642.7960 440.3460], 归一化后得到[-0.0360 -0.8244 0.5648]

## 步骤5.2: 移动到场景中心

将物体整体移动: [891.5705 418.9890 -260.3760], 使得物体中心移动到场景中心。

换算成UE4 (y轴反向) 中的中点坐标为: [-891.5705 418.9890 260.3760], 旋转轴向量为: [-0.0360 0.8244 0.5648]

按上面的方法, 将物体坐标轴置零, 导出成fbx文件, 并导入到UE4中, 填写的XML文件如下

## 步骤6.3: 编写XML文件

换算成UE4 (y轴反向) 中的中点坐标为: [-891.5705 418.9890 260.3760], 旋转轴向量为: [-0.0360 0.8244 0.5648]

得到XML文件数据如下

```
<Actuator>
<MeshPath>/Game/MQ-9Reaper/ElevatorR</MeshPath>
<MaterialPath></MaterialPath>
<RelativePosToBodyCm>
<x>-891.5705</x>
<y>418.9890</y>
<z>260.3760</z>
</RelativePosToBodyCm>
<RelativeAngEulerToBodyDeg>
<roll>0</roll>
```

<pitch>0</pitch>

<yaw>0</yaw>

</RelativeAngEulerToBodyDeg>

<RotationAxisVectorToBody>

<x>-0.0360</x>

<y>0.8244</y>

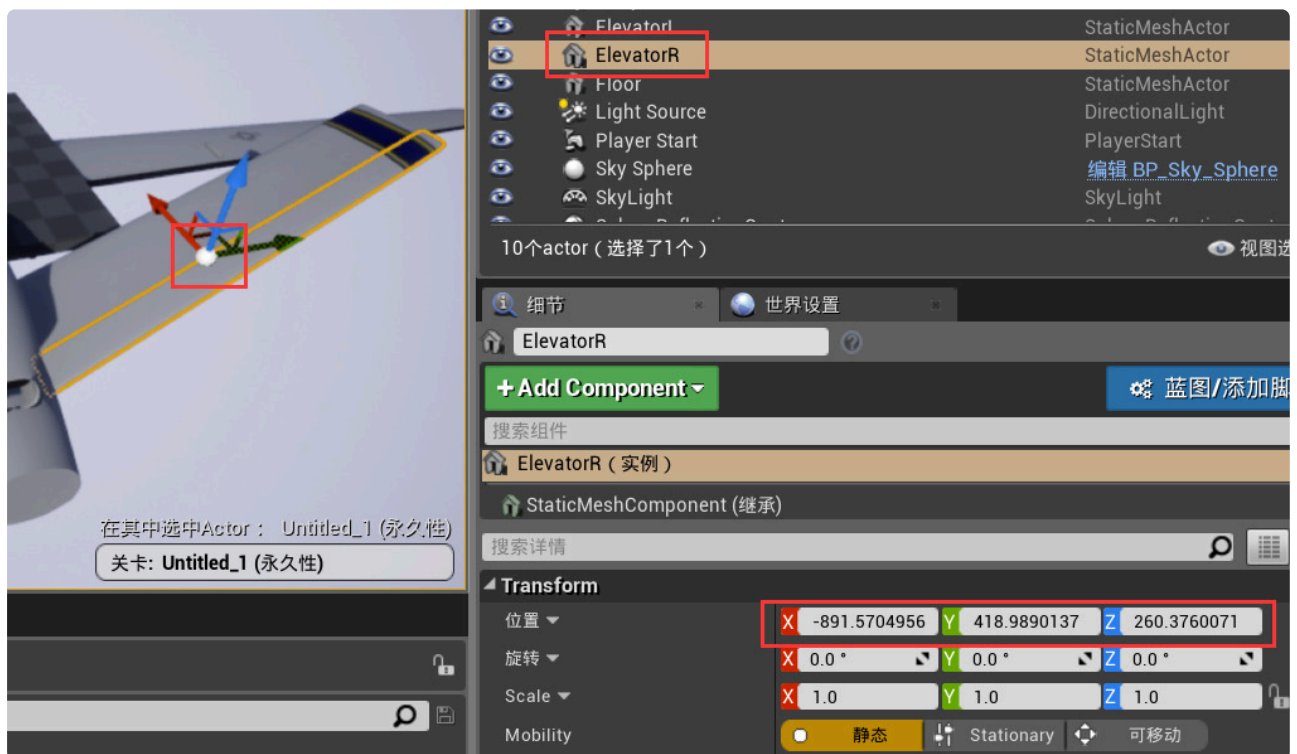
<z>0.5648</z>

</RotationAxisVectorToBody>

<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>

</Actuator>

## 步骤5.4: 导入UE4，拖入关卡中，输入位置坐标，查看拼合效果如下



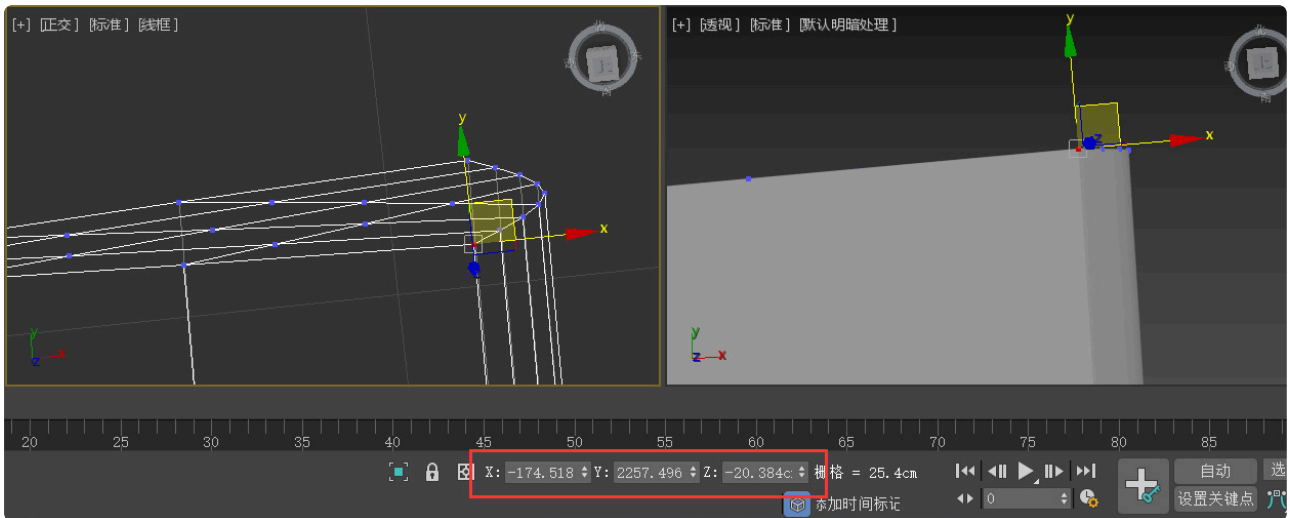
## 步骤6: 左侧副翼AileronL的执行器导出方法

打开MQ-9Reaper-Zeroed.max，另存一个文件为AileronL.max

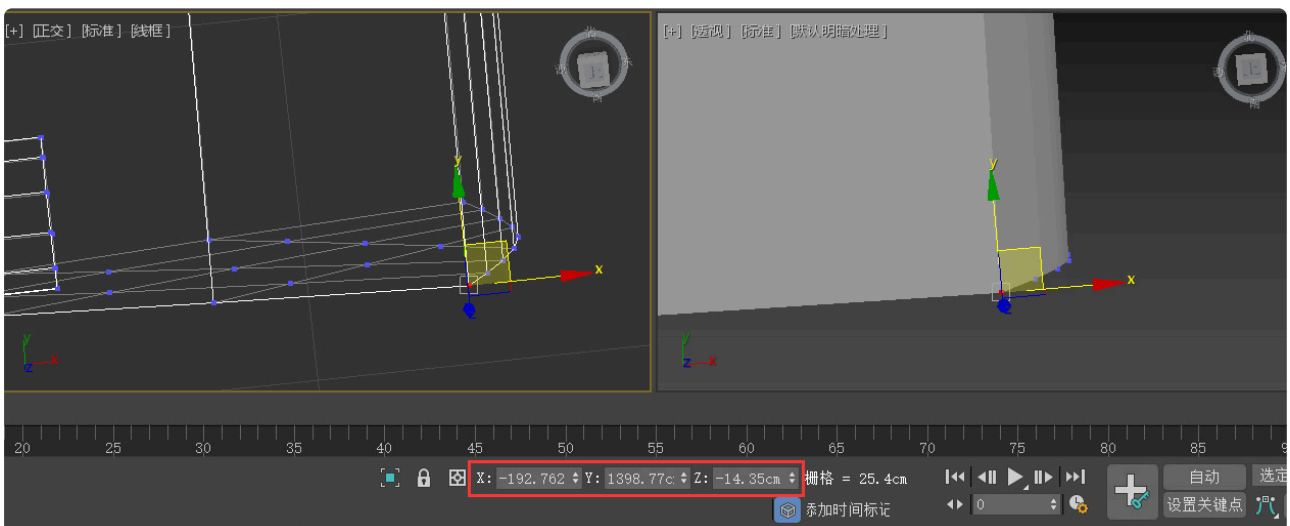
删掉其他物体，只保留“AileronL”，并选中它，开始下面操作

## 步骤6.1：获取转轴信息

在3Ds Max中读取得到左上前沿端点：[-174.518, 2257.496, -20.384]



右上前沿端点：[-192.762, 1398.77, -14.35]

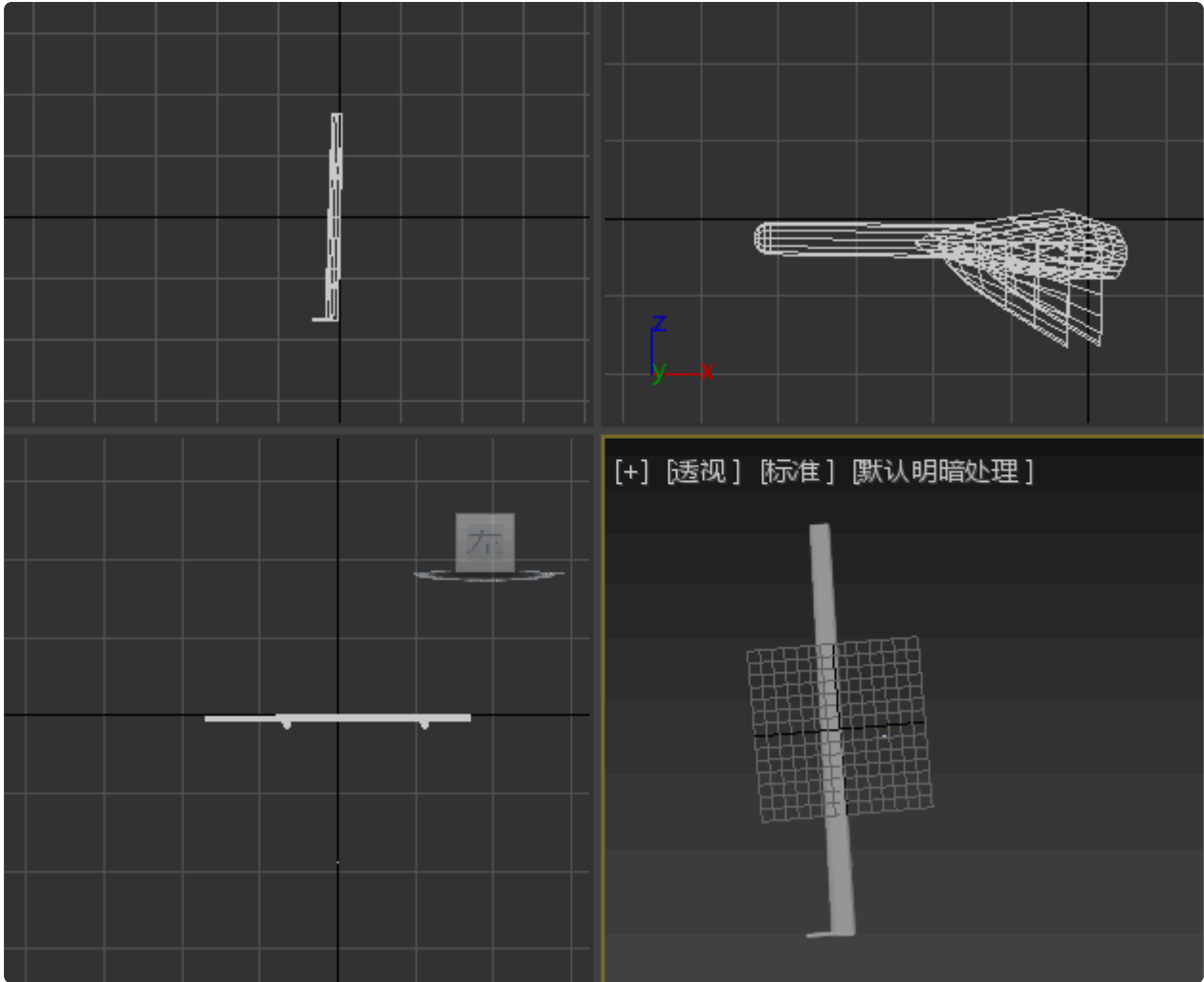


中点：[-183.64 1828.133 -17.367]

方向向量：[-18.2440 -858.7260 6.0340]，归一化后得到[-0.0212 -0.9997 0.0070]

## 步骤6.2：移动到场景中心

根据以上信息，将物体整体移动：[183.64 -1828.133 17.367]，使得物体中心移动到场景中心。



保存文件，并导出成fbx，再导入到UE4中。

### 步骤6.3：编写XML文件

换算成UE4（y轴反向）中的中点坐标为：[-183.64 -1828.133 -17.367]，旋转轴向量为：[-0.0212 0.9997 0.0070]

```
<Actuator>
```

```
<MeshPath>/Game/MQ-9Reaper/AileronL</MeshPath>
```

```
<MaterialPath></MaterialPath>
```

```
<RelativePosToBodyCm>
```

```
<x>-183.64</x>
```

```
<y>-1828.133</y>
```

```
<z>-17.367</z>
```

```
</RelativePosToBodyCm>
```

<RelativeAngEulerToBodyDeg>

<roll>0</roll>

<pitch>0</pitch>

<yaw>0</yaw>

</RelativeAngEulerToBodyDeg>

<RotationAxisVectorToBody>

<x>-0.0212</x>

<y>0.9997</y>

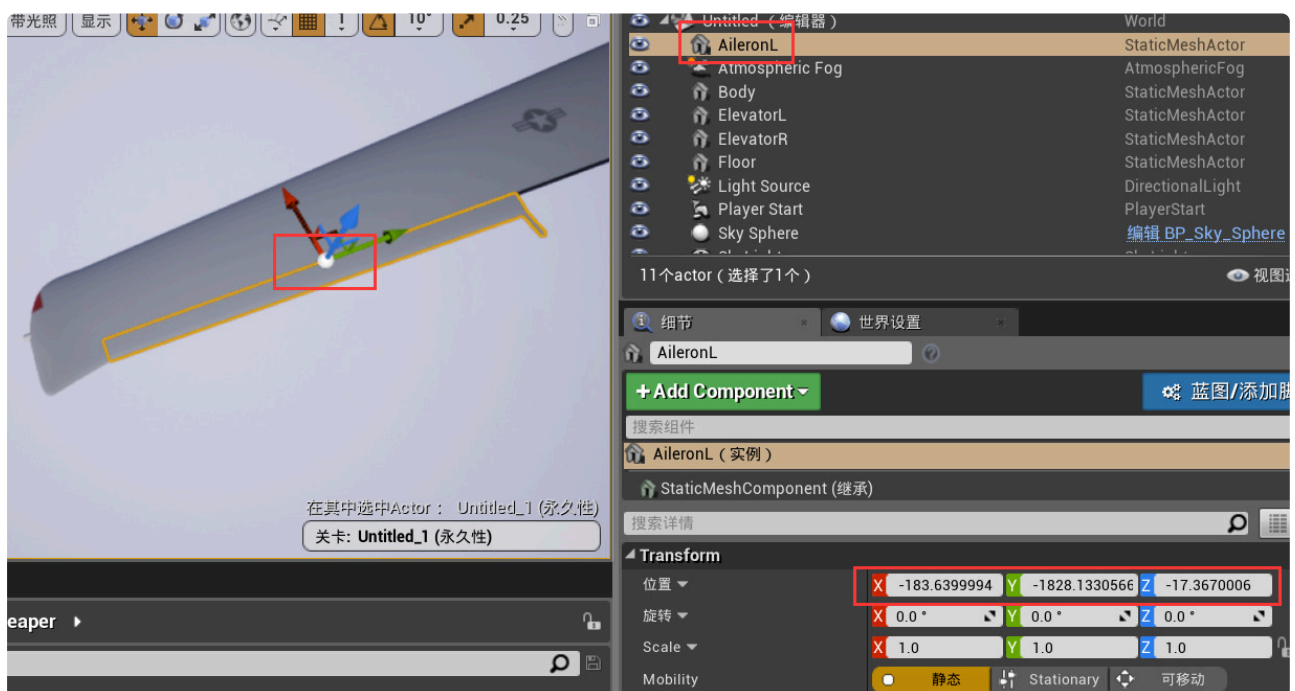
<z>0.0070</z>

</RotationAxisVectorToBody>

<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>

</Actuator>

步骤6.4：导入UE4，拖入关卡中，输入位置坐标，查看拼合效果如下



## 步骤7：右侧副翼AileronR的执行器导出

打开MQ-9Reaper-Zeroed.max，另存一个文件为AileronR.max

删掉其他物体，只保留“AileronR”，并选中它，开始下面操作

## 步骤7.1：获取转轴信息

在3Ds Max中读取得到左上前沿端点：[-192.762, -1398.77, -14.349]

右上前沿端点：[-174.517, -2257.496, -20.384]

中点：[-183.6395 -1828.133 -17.3665]

方向向量：[18.2450 -858.7260 -6.0350]，归一化后得到[0.0212 -0.9997 -0.0070]

## 步骤7.2：移动到场景中心

根据以上信息，将物体整体移动：[183.6395 1828.133 17.3665]，使得物体中心移动到场景中心。

保存文件，并导出成fbx，再导入到UE4中。

## 步骤7.3：编写XML文件

换算成UE4（y轴反向）中的中点坐标为：[-183.6395 1828.133 -17.3665]，旋转轴向量为：[0.0212 0.9997 -0.0070]

```
<Actuator>
```

```
<MeshPath>/Game/MQ-9Reaper/AileronR</MeshPath>
```

```
<MaterialPath></MaterialPath>
```

```
<RelativePosToBodyCm>
```

```
<x>-183.6395</x>
```

```
<y>1828.133</y>
```

```
<z>-17.3665</z>
```

```
</RelativePosToBodyCm>
```

```
<RelativeAngEulerToBodyDeg>
```

```
<roll>0</roll>
```

```
<pitch>0</pitch>
```

```
<yaw>0</yaw>
```

```
</RelativeAngEulerToBodyDeg>
```

```
<RotationAxisVectorToBody>
```

```
<x>0.0212</x>
```

```
<y>0.9997</y>
```

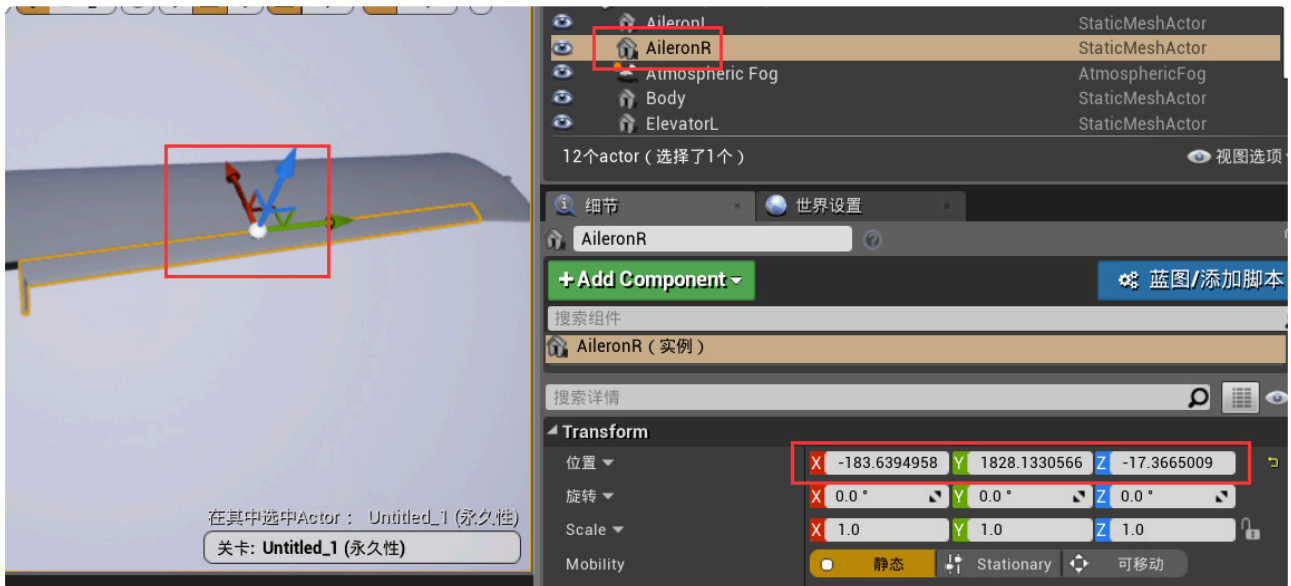
```
<z>-0.0070</z>
```

```
</RotationAxisVectorToBody>
```

```
<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>
```

```
</Actuator>
```

## 步骤7.4: 导入UE4, 拖入关卡中, 输入位置坐标, 查看拼合效果如下



## 步骤8: 方向舵Rudder的执行器导出

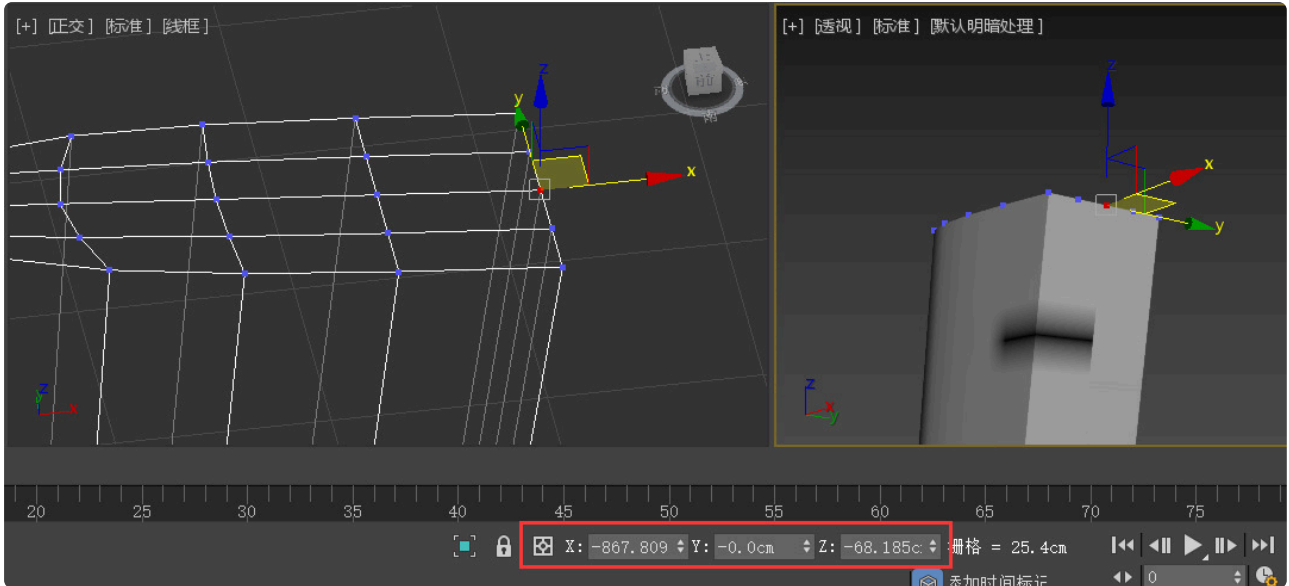
打开MQ-9Reaper-Zeroed.max, 另存一个文件为Rudder.max

删掉其他物体, 只保留“Rudder”, 并选中它, 开始下面操作

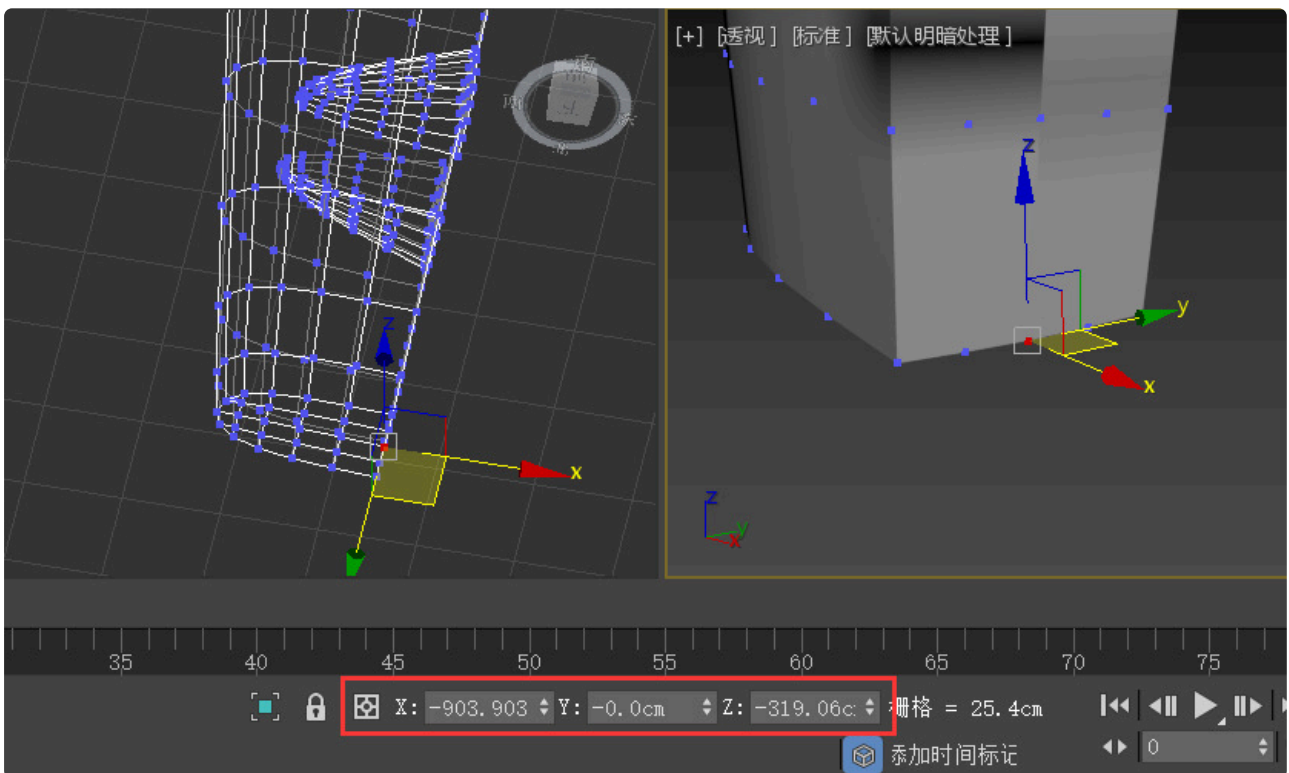
### 步骤8.1: 获取转轴信息

由于Rudder的转轴是竖直方向的, 因此需要沿上下沿取两个点, 得到中心和转轴方向。

在3Ds Max中读取得到上前沿端点: [-867.809, 0, -68.185]



下前沿端点:  $[-903.903, 0, -319.06]$

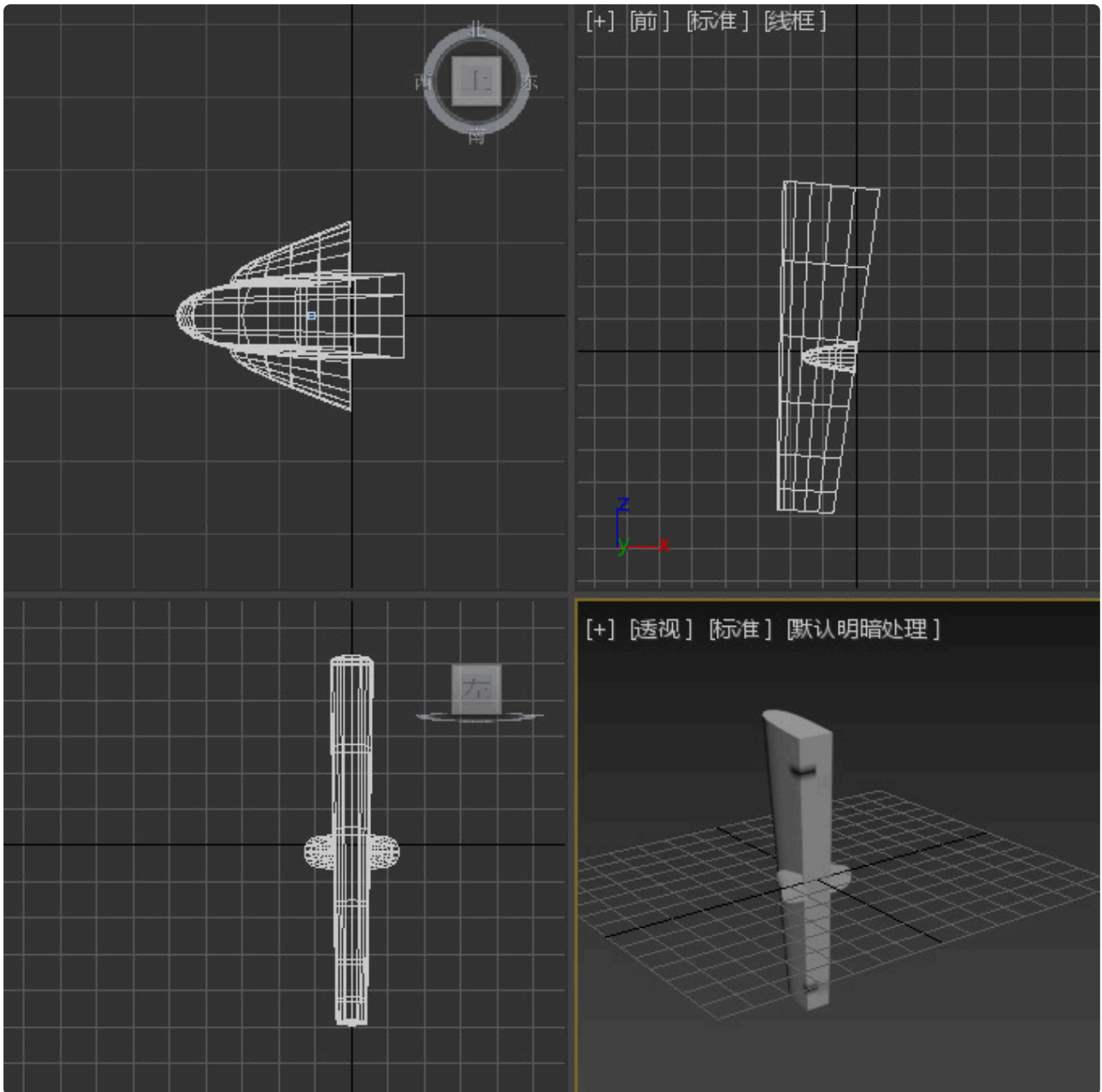


中点:  $[-885.8560 \ 0 \ -193.6225]$

方向向量:  $[-36.0940 \ 0 \ -250.875]$ , 归一化后得到 $[-0.1424 \ 0 \ -0.9898]$

## 步骤8.2: 移动到场景中心

根据以上信息, 将物体整体移动:  $[885.8560 \ 0 \ 193.6225]$ , 使得物体中心移动到场景中心。



保存文件，并导出成fbx，再导入到UE4中。

### 步骤8.3：编写XML文件

换算成UE4（y轴反向）中的中点坐标为：[-885.8560 0 -193.6225]，旋转轴向量为：[-0.1424 0 -0.9898]

```
<Actuator>
```

```
<MeshPath>/Game/MQ-9Reaper/Rudder</MeshPath>
```

```
<MaterialPath></MaterialPath>
```

```
<RelativePosToBodyCm>
```

```
<x>-885.8560</x>
```

<y>0</y>

<z>-193.6225</z>

</RelativePosToBodyCm>

<RelativeAngEulerToBodyDeg>

<roll>0</roll>

<pitch>0</pitch>

<yaw>0</yaw>

</RelativeAngEulerToBodyDeg>

<RotationAxisVectorToBody>

<x>-0.1424</x>

<y>0</y>

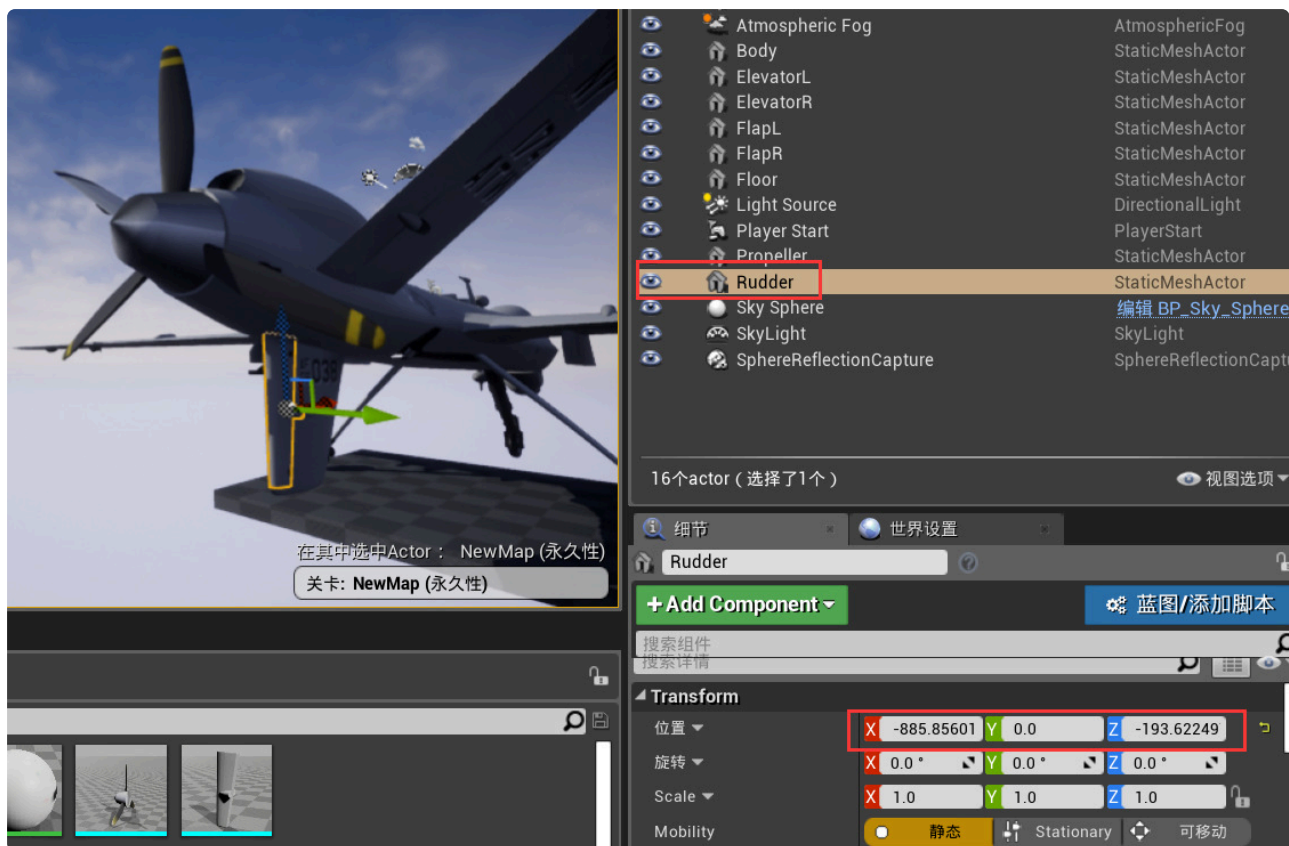
<z>-0.9898</z>

</RotationAxisVectorToBody>

<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>

</Actuator>

## 步骤8.4: 导入UE4, 拖入关卡中, 输入位置坐标, 查看拼合效果如下



## 步骤9: 左侧襟翼FlapL的执行器导出

打开MQ-9Reaper-Zeroed.max, 另存一个文件为FlapL.max

删掉其他物体, 只保留“FlapL”,

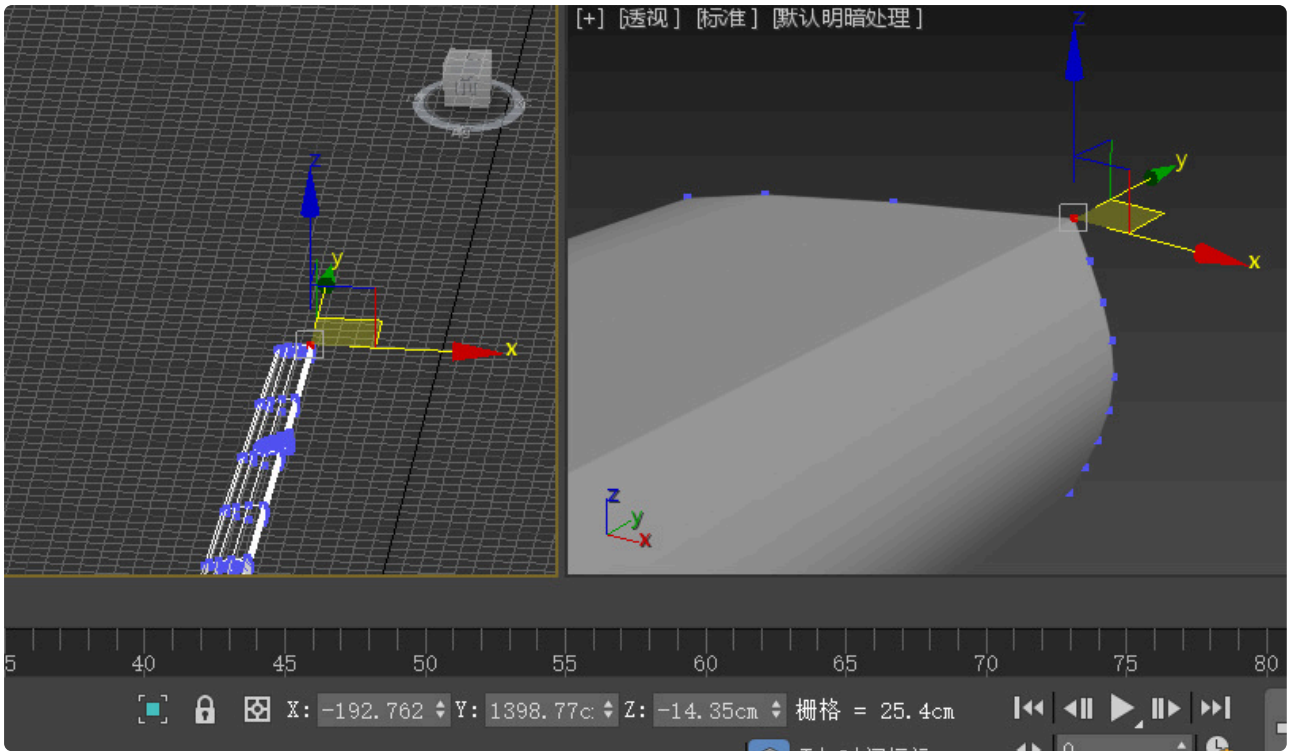
### 步骤9.1: 获取转轴信息

由于FlapL中包含了两个子物体, FlapL1和FlapL2, 需要将FlapL先解组, 才能进行点拾取操作。

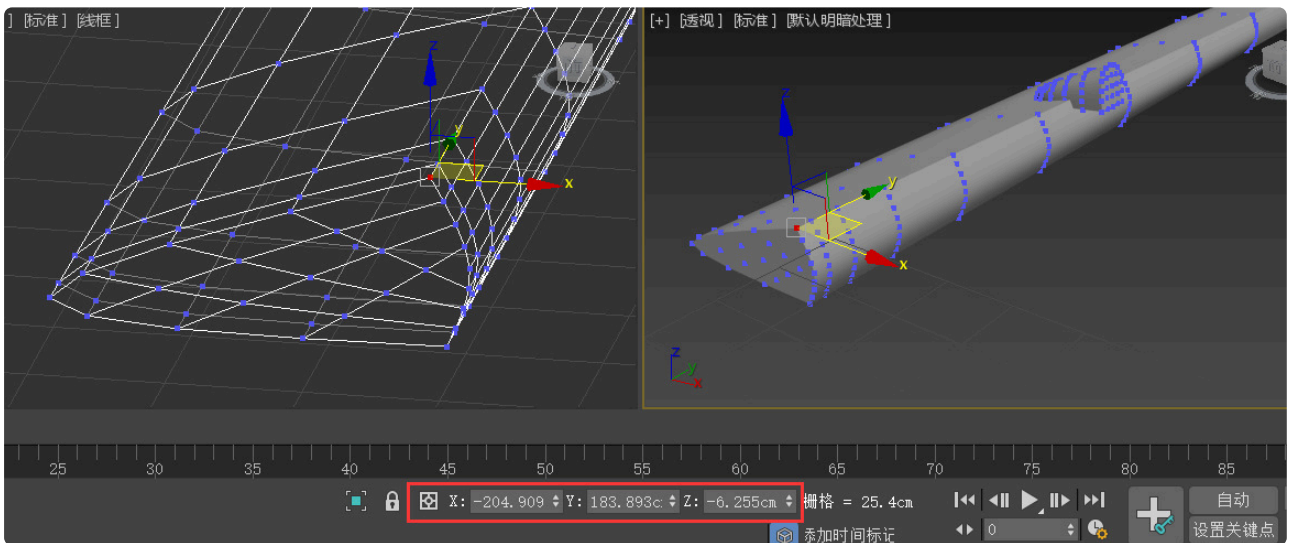
3Ds

Max中将FlapL解组, 选中FlapL2, 并进入点编辑模式, 读取得到最左上前沿端点:

[-192.762,  
1398.77, -14.35]



选中FlapL1, 右上前沿端点: [-204.909, 183.893, -6.255]

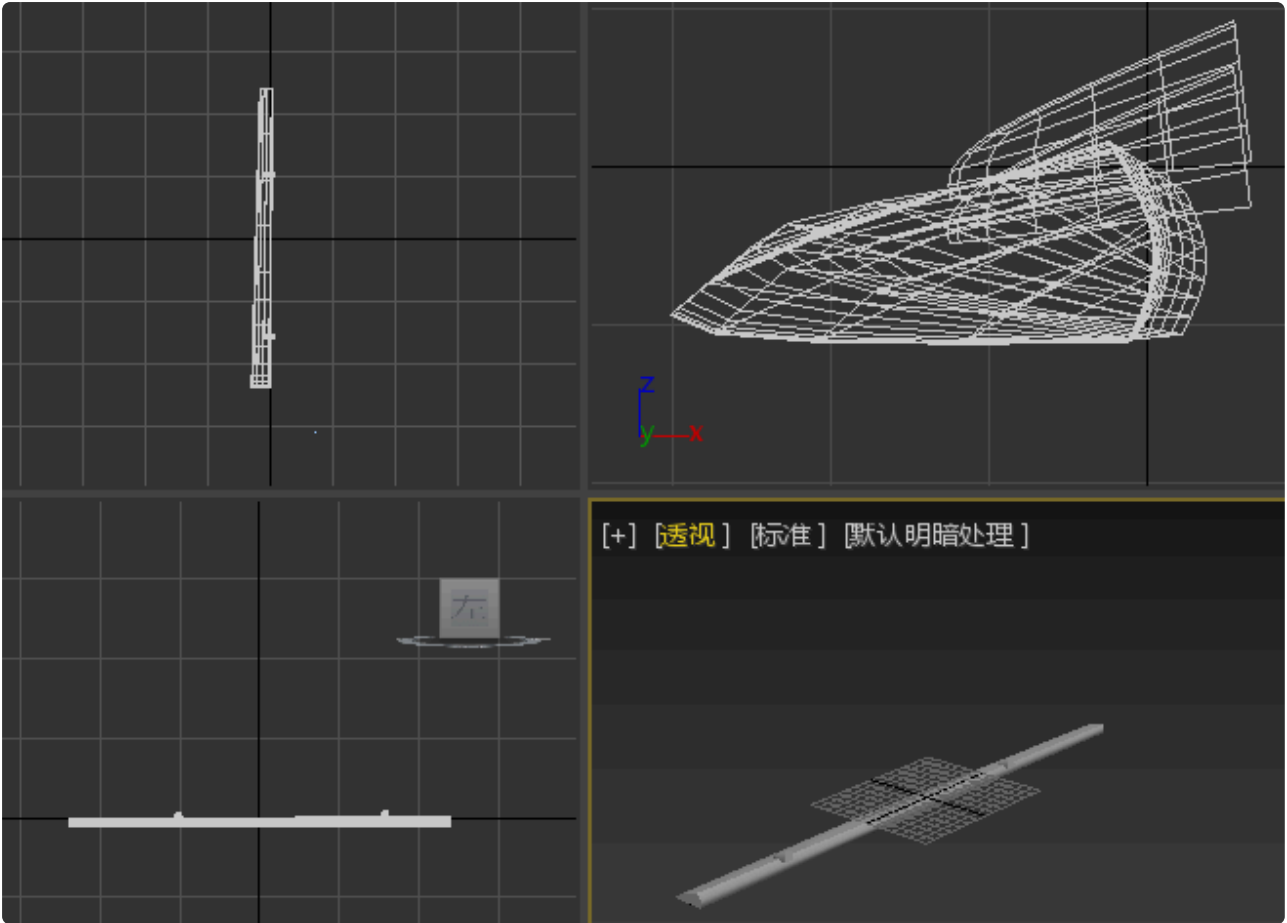


中点: [-198.8355 791.3315 -10.3025]

方向向量: [-12.147 -1214.877 8.095], 归一化后得到[-0.0100 -0.9999 0.0067]

## 步骤9.2: 移动到场景中心

根据以上信息, 选中所有物体, 将物体整体移动: [198.8355 -791.3315 10.3025], 使得物体中心移动到场景中心。



保存文件，并导出成fbx，再导入到UE4中。

### 步骤9.3：编写XML文件

换算成UE4（y轴反向）中的中点坐标为：[-198.8355 -791.3315 -10.3025]，旋转轴向量为：[-0.0100 0.9999 0.0067]

```
<Actuator>
```

```
<MeshPath>/Game/MQ-9Reaper/FlapL</MeshPath>
```

```
<MaterialPath></MaterialPath>
```

```
<RelativePosToBodyCm>
```

```
<x>-198.8355</x>
```

```
<y>-791.3315</y>
```

```
<z>-10.3025</z>
```

```
</RelativePosToBodyCm>
```

```
<RelativeAngEulerToBodyDeg>
```

<roll>0</roll>

<pitch>0</pitch>

<yaw>0</yaw>

</RelativeAngEulerToBodyDeg>

<RotationAxisVectorToBody>

<x>-0.0100</x>

<y>0.9999</y>

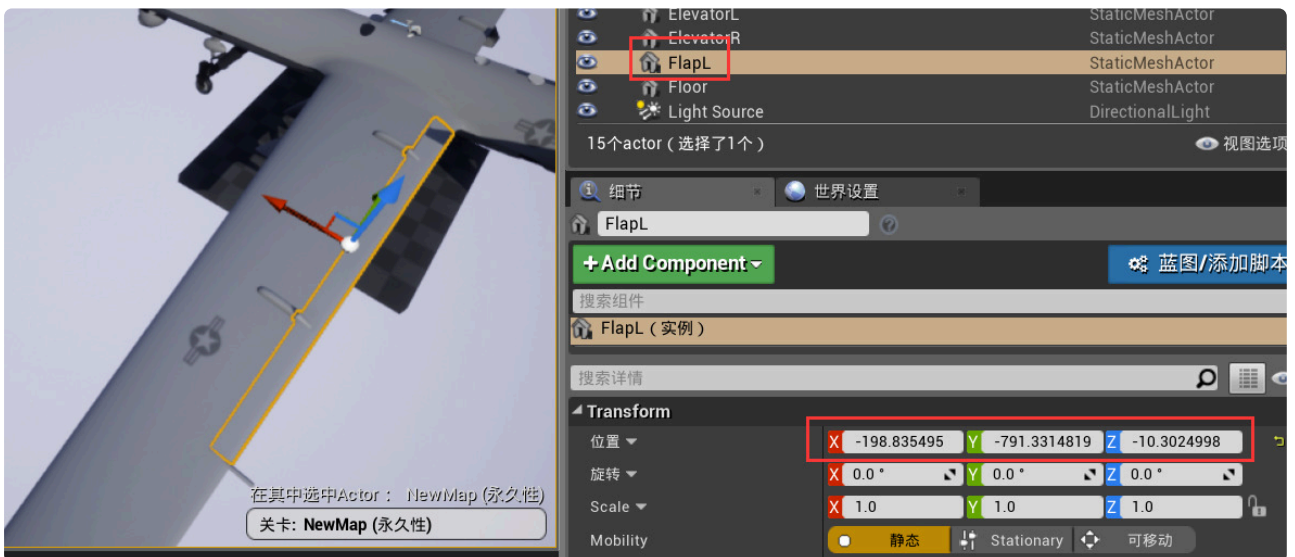
<z>0.0067</z>

</RotationAxisVectorToBody>

<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>

</Actuator>

## 步骤9.4：导入UE4，拖入关卡中，输入位置坐标，查看拼合效果如下



## 步骤10：右侧襟翼FlapR的执行器导出

打开MQ-9Reaper-Zeroed.max，另存一个文件为FlapR.max

删掉其他物体，只保留“FlapR”

## 步骤10.1：获取转轴信息

由于FlapR中包含了两个子物体，FlapR1和FlapR2，需要将FlapR先解组，才能进行点拾取操作。

3Ds

Max中将FlapR解组，选中FlapR1，并进入点编辑模式，读取得到最左上前沿端点：  
[-204.909,  
-183.893, -6.255]

选中FlapR2，右上前沿端点：[-192.762, -1398.77, -14.349]

中点：[-198.8355 -791.3315 -10.3020]

方向向量：[12.147 -1214.877 -8.094]，归一化后得到[0.0100 -0.9999 -0.0067]

## 步骤10.2：移动到场景中心

根据以上信息，将物体整体移动：[198.8355 791.3315  
10.3020]，使得物体中心移动到场景中心。

保存文件，并导出成fbx，再导入到UE4中。

## 步骤10.3：编写XML文件

换算成UE4（y轴反向）中的中点坐标为：[-198.8355 791.3315  
-10.3020]，旋转轴向量为：[0.0100 0.9999 -0.0067]

```
<Actuator>
```

```
<MeshPath>/Game/MQ-9Reaper/FlapR</MeshPath>
```

```
<MaterialPath></MaterialPath>
```

```
<RelativePosToBodyCm>
```

```
<x>-198.8355</x>
```

```
<y>791.3315</y>
```

```
<z>-10.3020</z>
```

```
</RelativePosToBodyCm>
```

```
<RelativeAngEulerToBodyDeg>
```

<roll>0</roll>

<pitch>0</pitch>

<yaw>0</yaw>

</RelativeAngEulerToBodyDeg>

<RotationAxisVectorToBody>

<x>0.01</x>

<y>0.9999</y>

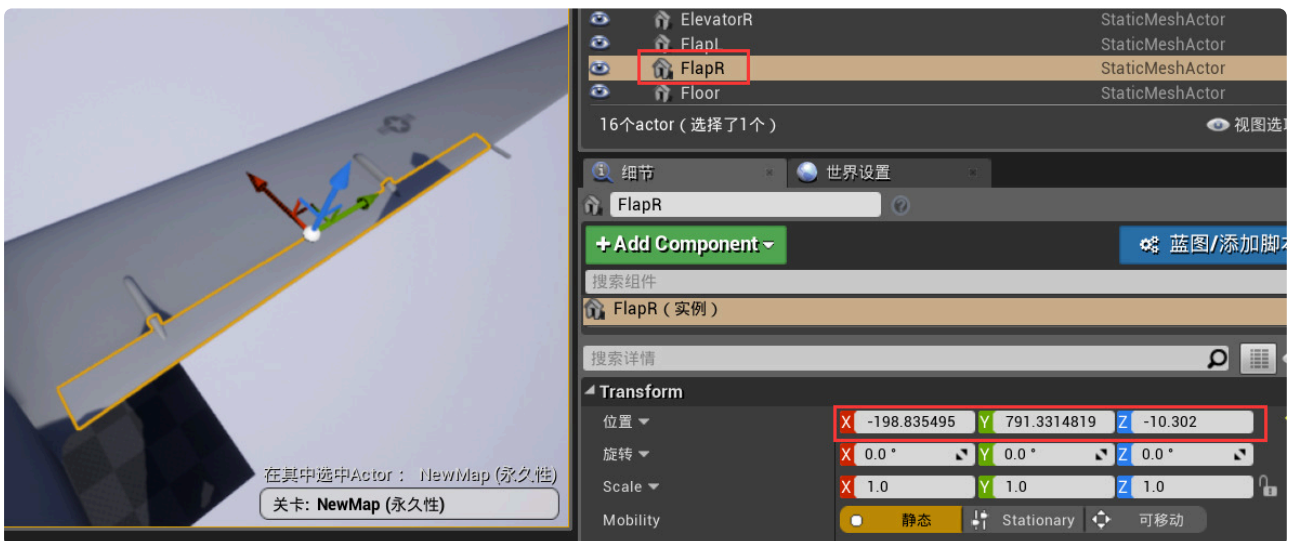
<z>-0.0067</z>

</RotationAxisVectorToBody>

<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>

</Actuator>

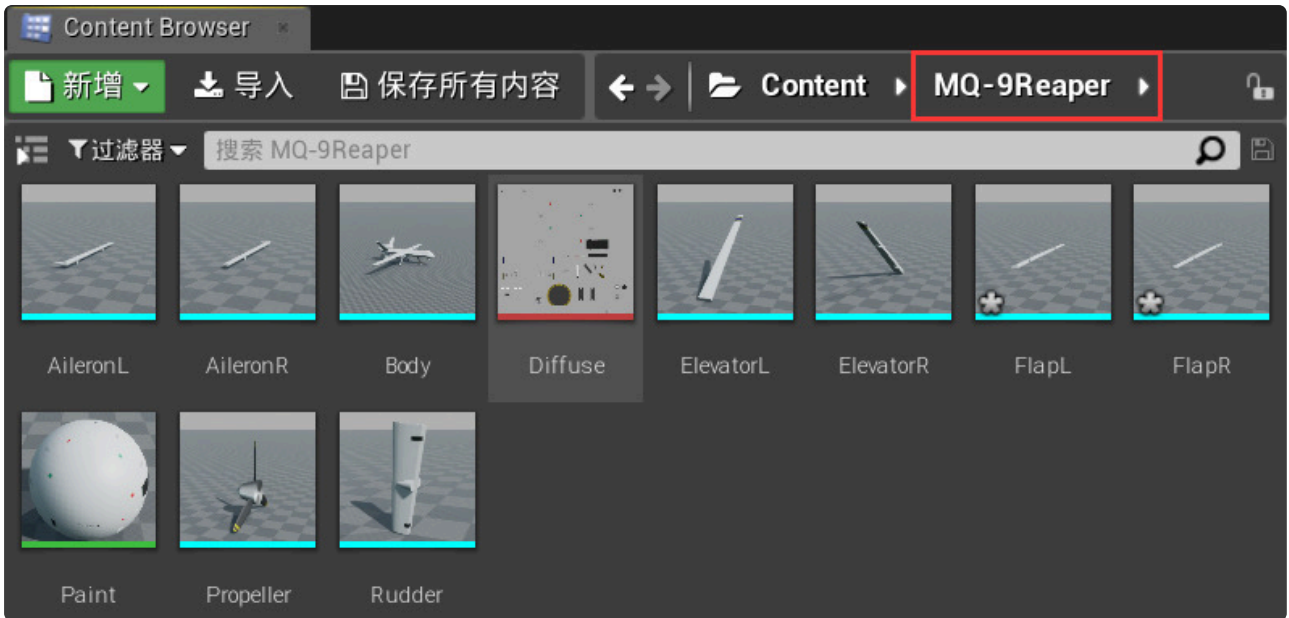
## 步骤10.4：导入UE4，拖入关卡中，输入位置坐标，查看拼合效果如下



## 步骤11：导入RflySim3D并进行微调

### 步骤11.1：导入UE4中查看整体效果

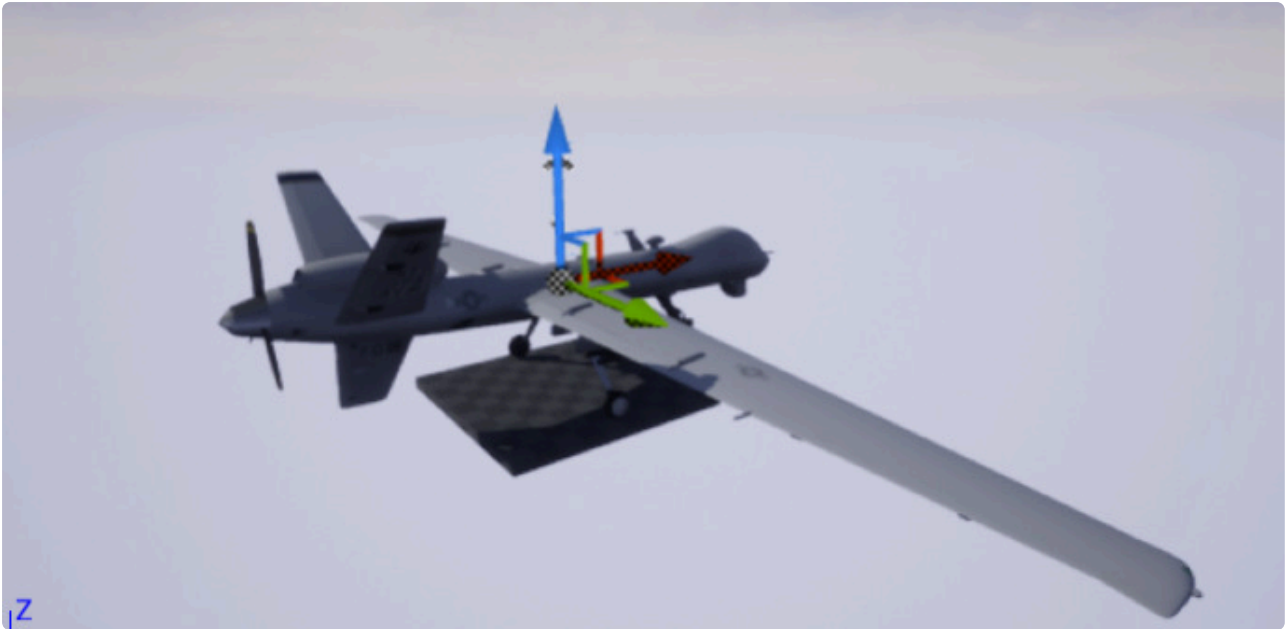
将上面生成得到的所有fbx组件都导入到UE4中。注：这里可以直接使用本文件夹的“UE4Content\MQ-9Reaper”目录拷贝到任意UE4项目的content目录下。注意：导入时需要将所有文件移动到一个文件夹，例如“MQ-9Reaper”，方便后续的导入导出。



在蓝图类中添加组件，并基于上述步骤中各个部件的导入以及配置位置坐标



将蓝图拖入场景，本例子的整体效果如下，可见各个部件没有错位，显示正常。



## 步骤11.2 烘焙出UE4文件

点击菜单栏“文件” - “为Windows烘焙内容”，烘焙结束后，去项目路径下的 Saved\Cooked\WindowsNoEditor\【你的项目名】\Content\MQ-9Reaper目录，直接将文件夹拷贝到RflySim3D的目录下C:\PX4PSP\RflySim3D\RflySim3D\Content\MQ-9Reaper

注：这里可以直接使用本文件夹下“UE4Cooked\MQ-9Reaper”目录下的例程内容，是已经烘焙好的。

## 步骤11.3：撰写XML文件

根据前文的机身和执行器的字段，加上摄像头视角的配置字段，得到最终XML文件MQ-9Reaper.xml代码（大小缩小到了0.1倍，要改回原尺寸请设置Scale的xyz分别为1）如下（例程见UE4Cooked\MQ-9Reaper\MQ-9Reaper.xml文件），将其拷贝到上面的MQ-9Reaper文件夹中

```

1  \<?xml version="1.0"?\>
2  \<vehicle\>
3  \<ClassID\>100\</ClassID\>
4  \<DisplayOrder\>900\</DisplayOrder\>
5  \<Name\>MQ-9Reaper\</Name\>
6  \<Scale\>
7  \<x\>0.1\</x\>
8  \<y\>0.1\</y\>
9  \<z\>0.1\</z\>
10 \</Scale\>
11 \<AngEulerDeg\>
12 \<roll\>0\</roll\>
13 \<pitch\>0\</pitch\>
14 \<yaw\>0\</yaw\>
15 \</AngEulerDeg\>
16 \<body\>
17 \<ModelType\>0\</ModelType\>
18 \<MeshPath\>/Game/MQ-9Reaper/Body \</MeshPath\>
19 \<MaterialPath\>\</MaterialPath\>
20 \<AnimationPath\>\</AnimationPath\>
21 \<CenterHeightAboveGroundCm\>478\</CenterHeightAboveGroundCm\>
22 \<NumberHeigthAboveCenterCm\>700\</NumberHeigthAboveCenterCm\>
23 \<NumberSizeScale\>50\</NumberSizeScale\>
24 \</body\>
25 \<ActuatorList\>
26 \<Actuator\>
27 \<MeshPath\>/Game/MQ-9Reaper/Propeller\</MeshPath\>
28 \<MaterialPath\>\</MaterialPath\>
29 \<RelativePosToBodyCm\>
30 \<x\>-1213.674\</x\>
31 \<y\> 0\</y\>
32 \<z\>69.264\</z\>
33 \</RelativePosToBodyCm\>
34 \<RelativeAngEulerToBodyDeg\>
35 \<roll\>0\</roll\>
36 \<pitch\>0\</pitch\>
37 \<yaw\>0\</yaw\>
38 \</RelativeAngEulerToBodyDeg\>
39 \<RotationAxisVectorToBody\>
40 \<x\>1\</x\>
41 \<y\>0\</y\>
42 \<z\>0\</z\>
43 \</RotationAxisVectorToBody\>
44 \<RotationModeSpinOrDefect\>0\</RotationModeSpinOrDefect\>
45 \</Actuator\>
46 \<Actuator\>
47 \<MeshPath\>/Game/MQ-9Reaper/ElevatorL\</MeshPath\>
48 \<MaterialPath\>\</MaterialPath\>
49 \<RelativePosToBodyCm\>
50 \<x\>-891.5705\</x\>
51 \<y\> -418.9885\</y\>
52

```

```

53  \<z\>260.3760\</z\>
54  \</RelativePosToBodyCm\>
55  \<RelativeAngEulerToBodyDeg\>
56  \<roll\>0\</roll\>
57  \<pitch\>0\</pitch\>
58  \<yaw\>0\</yaw\>
59  \</RelativeAngEulerToBodyDeg\>
60  \<RotationAxisVectorToBody\>
61  \<x\>0.0360\</x\>
62  \<y\>0.8245\</y\>
63  \<z\>-0.5648\</z\>
64  \</RotationAxisVectorToBody\>
65  \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
66  \</Actuator\>
67  \<Actuator\>
68  \<MeshPath\>/Game/MQ-9Reaper/ElevatorR\</MeshPath\>
69  \<MaterialPath\>\</MaterialPath\>
70  \<RelativePosToBodyCm\>
71  \<x\>-891.5705\</x\>
72  \<y\>418.9890\</y\>
73  \<z\>260.3760\</z\>
74  \</RelativePosToBodyCm\>
75  \<RelativeAngEulerToBodyDeg\>
76  \<roll\>0\</roll\>
77  \<pitch\>0\</pitch\>
78  \<yaw\>0\</yaw\>
79  \</RelativeAngEulerToBodyDeg\>
80  \<RotationAxisVectorToBody\>
81  \<x\>-0.0360\</x\>
82  \<y\>0.8244\</y\>
83  \<z\>0.5648\</z\>
84  \</RotationAxisVectorToBody\>
85  \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
86  \</Actuator\>
87  \<Actuator\>
88  \<MeshPath\>/Game/MQ-9Reaper/AileronL\</MeshPath\>
89  \<MaterialPath\>\</MaterialPath\>
90  \<RelativePosToBodyCm\>
91  \<x\>-183.64\</x\>
92  \<y\>-1828.133\</y\>
93  \<z\>-17.367\</z\>
94  \</RelativePosToBodyCm\>
95  \<RelativeAngEulerToBodyDeg\>
96  \<roll\>0\</roll\>
97  \<pitch\>0\</pitch\>
98  \<yaw\>0\</yaw\>
99  \</RelativeAngEulerToBodyDeg\>
100 \<RotationAxisVectorToBody\>
101 \<x\>-0.0212\</x\>
102 \<y\>0.9997\</y\>
103 \<z\>0.0070\</z\>
104
105

```

```

106 \</RotationAxisVectorToBody\>
107 \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
108 \</Actuator\>
109 \<Actuator\>
110 \<MeshPath\>/Game/MQ-9Reaper/AileronR\</MeshPath\>
111 \<MaterialPath\>\</MaterialPath\>
112 \<RelativePosToBodyCm\>
113 \<x\>-183.6395\</x\>
114 \<y\>1828.133\</y\>
115 \<z\>-17.3665\</z\>
116 \</RelativePosToBodyCm\>
117 \<RelativeAngEulerToBodyDeg\>
118 \<roll\>0\</roll\>
119 \<pitch\>0\</pitch\>
120 \<yaw\>0\</yaw\>
121 \</RelativeAngEulerToBodyDeg\>
122 \<RotationAxisVectorToBody\>
123 \<x\>0.0212\</x\>
124 \<y\>0.9997\</y\>
125 \<z\>-0.0070\</z\>
126 \</RotationAxisVectorToBody\>
127 \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
128 \</Actuator\>
129 \<Actuator\>
130 \<MeshPath\>/Game/MQ-9Reaper/Rudder\</MeshPath\>
131 \<MaterialPath\>\</MaterialPath\>
132 \<RelativePosToBodyCm\>
133 \<x\>-885.8560\</x\>
134 \<y\>0\</y\>
135 \<z\>-193.6225\</z\>
136 \</RelativePosToBodyCm\>
137 \<RelativeAngEulerToBodyDeg\>
138 \<roll\>0\</roll\>
139 \<pitch\>0\</pitch\>
140 \<yaw\>0\</yaw\>
141 \</RelativeAngEulerToBodyDeg\>
142 \<RotationAxisVectorToBody\>
143 \<x\>-0.1424\</x\>
144 \<y\>0\</y\>
145 \<z\>-0.9898\</z\>
146 \</RotationAxisVectorToBody\>
147 \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
148 \</Actuator\>
149 \<Actuator\>
150 \<MeshPath\>/Game/MQ-9Reaper/FlapL\</MeshPath\>
151 \<MaterialPath\>\</MaterialPath\>
152 \<RelativePosToBodyCm\>
153 \<x\>-198.8355\</x\>
154 \<y\>-791.3315\</y\>
155 \<z\>-10.3025\</z\>
156 \</RelativePosToBodyCm\>
157
158

```

```

159 \<RelativeAngEulerToBodyDeg\>
160 \<roll\>0\</roll\>
161 \<pitch\>0\</pitch\>
162 \<yaw\>0\</yaw\>
163 \</RelativeAngEulerToBodyDeg\>
164 \<RotationAxisVectorToBody\>
165 \<x\>-0.0100\</x\>
166 \<y\>0.9999\</y\>
167 \<z\>0.0067\</z\>
168 \</RotationAxisVectorToBody\>
169 \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
170 \</Actuator\>
171 \<Actuator\>
172 \<MeshPath\>/Game/MQ-9Reaper/FlapR\</MeshPath\>
173 \<MaterialPath\>\</MaterialPath\>
174 \<RelativePosToBodyCm\>
175 \<x\>-198.8355\</x\>
176 \<y\>791.3315\</y\>
177 \<z\>-10.3020\</z\>
178 \</RelativePosToBodyCm\>
179 \<RelativeAngEulerToBodyDeg\>
180 \<roll\>0\</roll\>
181 \<pitch\>0\</pitch\>
182 \<yaw\>0\</yaw\>
183 \</RelativeAngEulerToBodyDeg\>
184 \<RotationAxisVectorToBody\>
185 \<x\>0.01\</x\>
186 \<y\>0.9999\</y\>
187 \<z\>-0.0067\</z\>
188 \</RotationAxisVectorToBody\>
189 \<RotationModeSpinOrDefect\>1\</RotationModeSpinOrDefect\>
190 \</Actuator\>
191 \</ActuatorList\>
192 \<OnboardCameras\>
193 \<camera\>
194 \<name\>Chase_Camera\</name\>
195 \<RelativePosToBodyCm\>
196 \<x\>-2500\</x\>
197 \<y\>0\</y\>
198 \<z\>500\</z\>
199 \</RelativePosToBodyCm\>
200 \<RelativeAngEulerToBodyDeg\>
201 \<roll\>0\</roll\>
202 \<pitch\>0\</pitch\>
203 \<yaw\>0\</yaw\>
204 \</RelativeAngEulerToBodyDeg\>
205 \</camera\>
206 \<camera\>
207 \<name\>Front_Camera\</name\>
208 \<RelativePosToBodyCm\>
209 \<x\>1000\</x\>
210
211

```

```

212 \<y\>0\</y\>
213 \<z\>0\</z\>
214 \</RelativePosToBodyCm\>
215 \<RelativeAngEulerToBodyDeg\>
216 \<roll\>0\</roll\>
217 \<pitch\>0\</pitch\>
218 \<yaw\>0\</yaw\>
219 \</RelativeAngEulerToBodyDeg\>
220 \</camera\>
221 \<camera\>
222 \<name\>Back_Camera\</name\>
223 \<RelativePosToBodyCm\>
224 \<x\>-1000\</x\>
225 \<y\>0\</y\>
226 \<z\>0\</z\>
227 \</RelativePosToBodyCm\>
228 \<RelativeAngEulerToBodyDeg\>
229 \<roll\>0\</roll\>
230 \<pitch\>0\</pitch\>
231 \<yaw\>180\</yaw\>
232 \</RelativeAngEulerToBodyDeg\>
233 \</camera\>
234 \<camera\>
235 \<name\>Right_Camera\</name\>
236 \<RelativePosToBodyCm\>
237 \<x\>0\</x\>
238 \<y\>2000\</y\>
239 \<z\>0\</z\>
240 \</RelativePosToBodyCm\>
241 \<RelativeAngEulerToBodyDeg\>
242 \<roll\>0\</roll\>
243 \<pitch\>0\</pitch\>
244 \<yaw\>90\</yaw\>
245 \</RelativeAngEulerToBodyDeg\>
246 \</camera\>
247 \<camera\>
248 \<name\>Left_Camera\</name\>
249 \<RelativePosToBodyCm\>
250 \<x\>0\</x\>
251 \<y\>-2000\</y\>
252 \<z\>0\</z\>
253 \</RelativePosToBodyCm\>
254 \<RelativeAngEulerToBodyDeg\>
255 \<roll\>0\</roll\>
256 \<pitch\>0\</pitch\>
257 \<yaw\>-90\</yaw\>
258 \</RelativeAngEulerToBodyDeg\>
259 \</camera\>
260 \<camera\>
261 \<name\>Down_Camera\</name\>
262 \<RelativePosToBodyCm\>
263
264

```

```

265 \<x\>0\</x\>
266 \<y\>0\</y\>
267 \<z\>-500\</z\>
268 \</RelativePosToBodyCm\>
269 \<RelativeAngEulerToBodyDeg\>
270 \<roll\>0\</roll\>
271 \<pitch\>-90\</pitch\>
272 \<yaw\>0\</yaw\>
273 \</RelativeAngEulerToBodyDeg\>
274 \</camera\>
275 \<camera\>
276 \<name\>Up_Camera\</name\>
277 \<RelativePosToBodyCm\>
278 \<x\>0\</x\>
279 \<y\>0\</y\>
280 \<z\>500\</z\>
    \</RelativePosToBodyCm\>
    \<RelativeAngEulerToBodyDeg\>
    \<roll\>0\</roll\>
    \<pitch\>90\</pitch\>
    \<yaw\>0\</yaw\>
    \</RelativeAngEulerToBodyDeg\>
    \</camera\>
    \</OnboardCameras\>
    \</vehicle\>

```

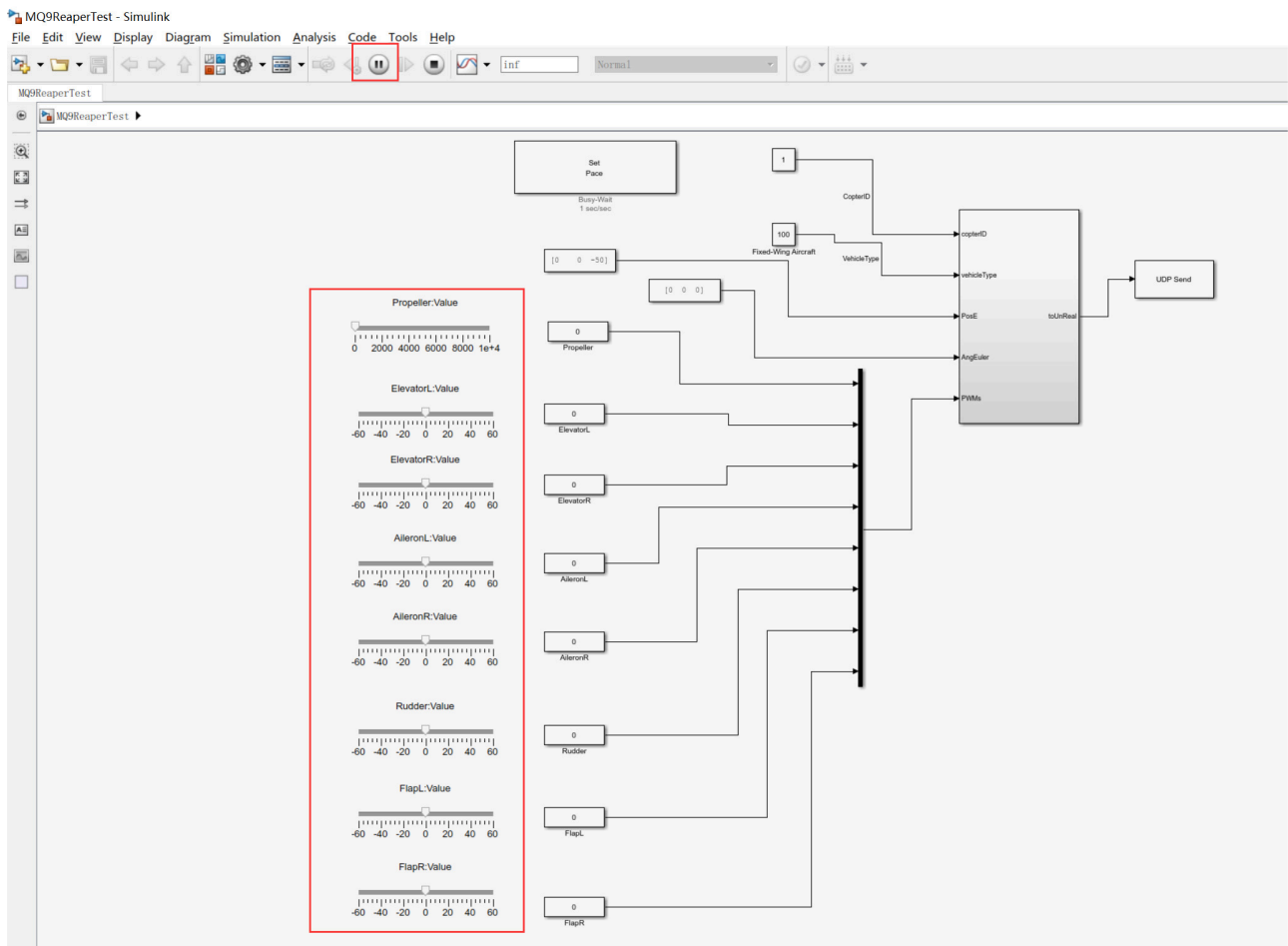
## 步骤11.4：测试最终效果

打开一个RflySim3D程序，再用MATLAB打开“MQ9ReaperTest.slx”

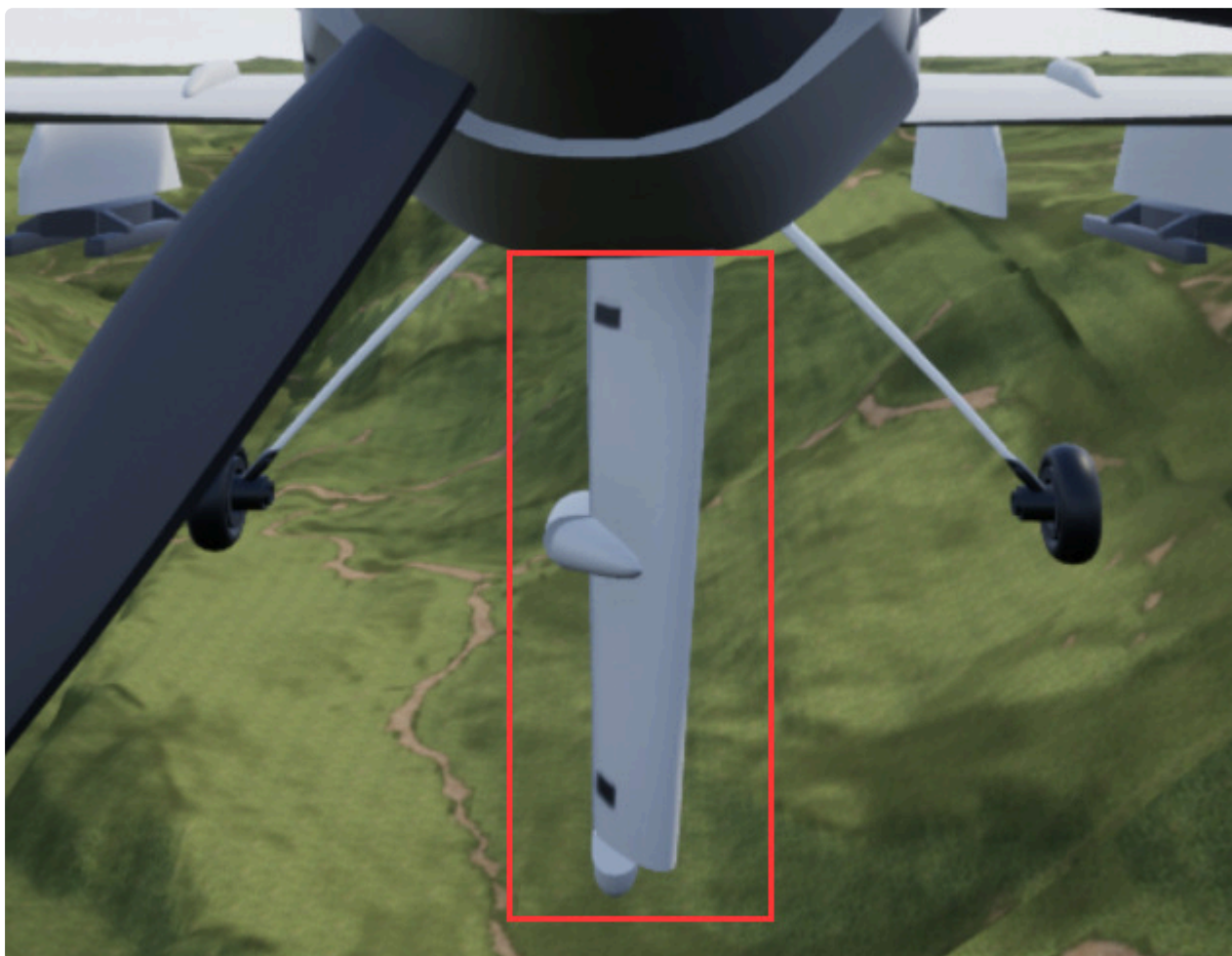
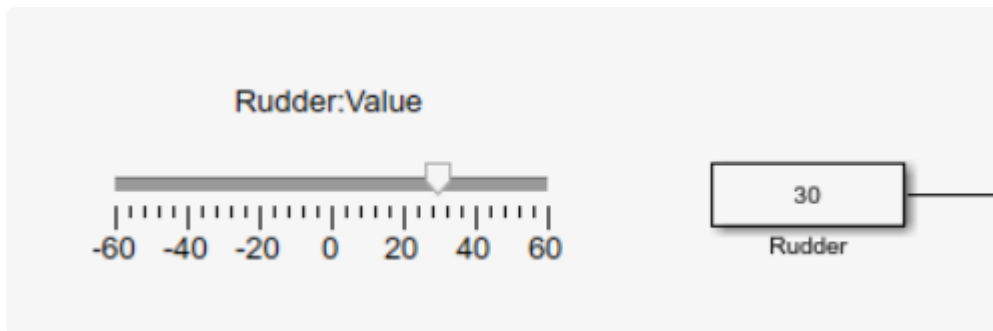
（例程见UE4Cooked\MQ-9Reaper\MQ9ReaperTest.slx文件），运行之后可以看到一个固定翼无人机在场景中。



依次拖动各个舵面的滑块，观察舵面是否能够按预设轴转动



例如：将方向舵设置到30度，看场景中飞机方向舵是否为30度偏转，有无错位现象



## 步骤12：最后效果

此文件目录下有一个已经完成的例程，在“\MQ-9Reaper\UE4Cooked”下找到MQ-9Reaper目录，将其拷入PX4PSP\RflySim3D\RflySim3D\Content目录下。最后打开RflySim3D，鼠标双击地面+O+100，就能创建出这个ClassID为100的飞机，然后按C键切换到刚刚导入的MQ-9Reaper飞机即可。



## 5. 关键知识点

### 关键知识点1

首先，在3ds Max中调整固定翼模型，区分出执行器（这里是8个舵面：AileronL 左副翼、AileronR 右副翼、ElevatorL 左升降舵、ElevatorR 右升降舵、FlapL 左襟翼、FlapR 右襟翼、Propeller 螺旋桨、Rudder 方向舵）和机身（除仿真所需执行器以外的所有组件），并获得其相对位置。然后将调整好的舵面和机身以静态网格体分别导入UE中。在UE中为舵面和机身添加材质，即定义它们的外观和纹理。最后，进行烘焙操作以将模型打包发布到windows平台，同时这会将材质渲染到模型表面以提高渲染效果并减少计算负担。最后将烘焙好的固定翼模型与配套的XML文件一同导入RflySim3D中。

这里选用3ds Max软件处理模型具有以下优势：

1. 精细调整模型：3ds Max提供了丰富的编辑工具，可以对模型进行精细调整。3ds Max还支持非破坏性编辑，可以在不影响原始模型的情况下进行修改。
2. 实时渲染：3ds Max内置了实时渲染器，可以在编辑过程中实时预览渲染效果。这使得用户可以即时看到模型和场景的最终外观，从而更好地进行调整和设计。
3. 多种格式导入导出：3ds Max支持导入和导出多种文件格式，如OBJ、FBX、STL、DWG等等。用户可以将3ds

Max中的模型和场景导出到其他软件进行后期处理，或者导入其他软件中的模型和场景进行进一步编辑。

在3dsMax中，通过[分组](#)来区分出执行器和机身。

1. 3dsmax创建组时，其所有成员对象都被严格链接至一个不可见的虚拟对象。该组对象使用这个虚拟对象的轴点和本地变换坐标系。
2. 虚拟对象：当你在3ds Max中创建一个组，所有成员对象都被链接至一个集中的虚拟对象，以通过合并多个简单运动来创建复杂运动。
3. 轴点：在3ds Max中，轴点（Pivot）是所有几何变换（包括旋转、移动和缩放）的参考点。在组中，所有成员对象的变换（旋转、缩放和移动）都以这个虚拟对象的轴点为中心进行。
4. 本地变换坐标系：本地坐标系是指被选中的组或物体自身的坐标轴。

将3ds Max软件处理完成的模型导入UE进行后期处理时，需要注意以下要点：

1. 在将3ds Max中处理完成的模型导入到UE中进行后期处理之前，需要注意两者之间坐标系的差异。3Ds Max是采用前左上的右手坐标系，UE4采用前右上的左手坐标系。
2. 导入格式：  
FBX是一种灵活的文件格式，归Autodesk所有，可以提供数字内容创建（DCC）应用程序之间的互操作性，某些应用程序（例如Autodesk Motionbuilder）本身支持该格式，而Autodesk Maya、Autodesk 3ds Max和Blender等其他软件使用[FBX插件](#)支持该格式。确保3dsMax导出时选择了正确的FBX版本，并按照UE的要求进行设置，例如坐标系、轴向、缩放等参数。
3. 材质和纹理：虽然FBX格式导入UE时支持导入材质和纹理资产，并自动将它们应用到静态网格体。但因为3ds Max和UE4使用不同的材质和纹理系统，导入UE后，需要重新调整和配置模型的材质和纹理。可以重新创建材质，或者导入3ds Max中使用的材质和纹理，并进行适当的调整和优化

## 6. 参考资料

1. [RflySim官方文档](#)
2. XML文件规则 `../.. /PX4PSP\RflySimAPIs\3.RflySim3DUE\API.pdf`
3. RflySim3D快捷键接口总览 `../.. /PX4PSP\RflySimAPIs\3.RflySim3DUE\API.pdf`

## 7. 常见问题

### Q1: 在3ds Max中导入FBX文件时，如何选择正确的单位？

A1: 在导入FBX文件时，应在FBX导入器上将文件单位转化为厘米，单位的选择应以源文件为准，避免后续的坐标变换产生尺寸的差异。此外，FBX导入器在UI上未提供"轴转化"选项，这是因为3ds Max FBX plug-in会自动计算和应用轴转化。

### Q2: 在将执行器导出到UE4时，为什么需要对Y轴坐标进行取反操作？

A2: 这是因为3ds Max和UE4使用的坐标系不同。3ds Max使用右手坐标系（前左上），而UE4使用左手坐标系（前右上），两者的Y轴方向相反。所以需要对Y坐标进行取反操作，以确保模型在UE4中的位置与在3ds Max中的一致。

### Q3: 如何正确设置执行器的旋转轴？

A3: 设置执行器旋转轴的步骤包括：1) 在3ds Max中选择执行器上旋转轴线上的两点；2) 计算这两点的中点作为旋转中心；3) 计算两点连线的方向向量作为旋转轴；4) 将方向向量归一化；5) 在XML文件中正确设置RotationAxisVectorToBody参数。注意可能需要根据实际情况对旋转轴方向进行调整，以符合实际控制逻辑。

---

1. <https://rflysim.com/> ↩

2. 推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf> ↩