
1、实验名称及目的

RflySim3D 自定义固定翼模型加载实验：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、实验原理

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

r) 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。

3、实验效果



图 1

4、文件目录

文件夹/文件名称	说明
max	3dsMax 处理的模型数据
UE4Cooked	烘焙好的数据
UE4Content	UE 中的模型数据

5、运行环境

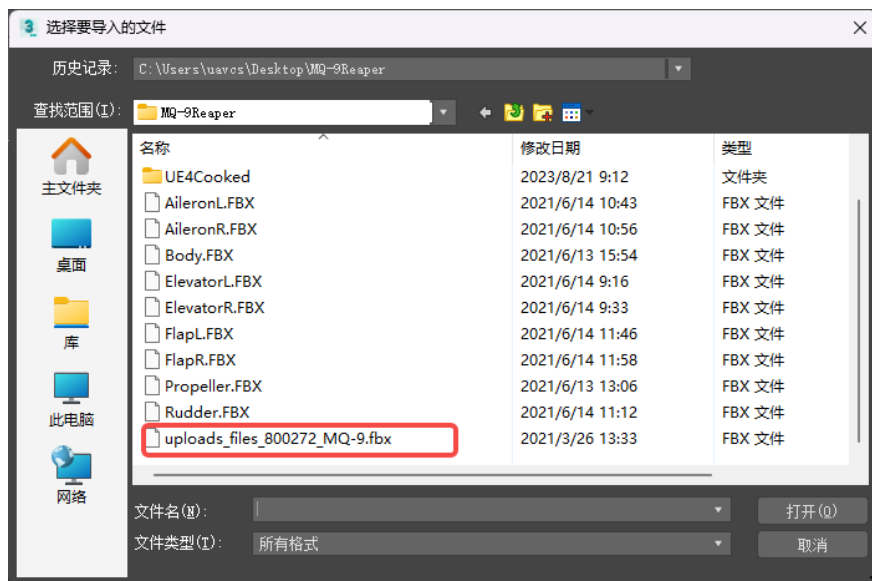
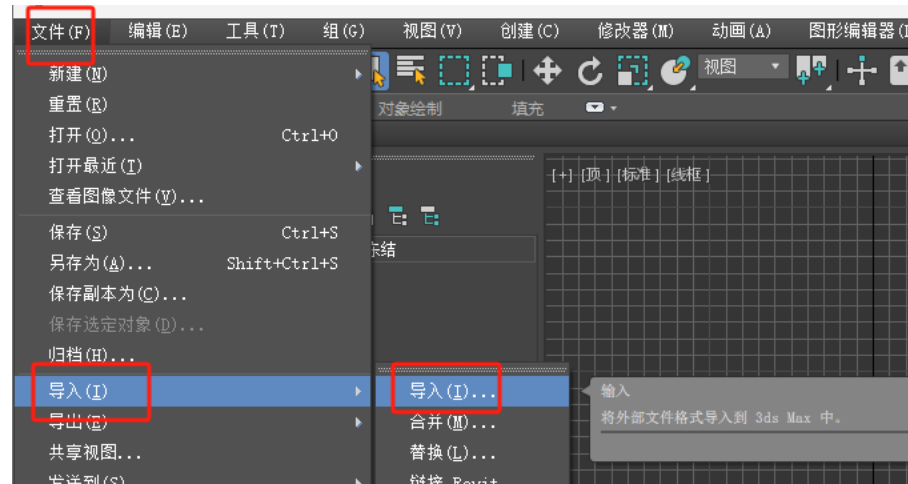
序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台免费版		
3	3ds Max2021		
4	Unreal Engine4.27		

推荐配置请见：<https://doc.rflysim.com/>

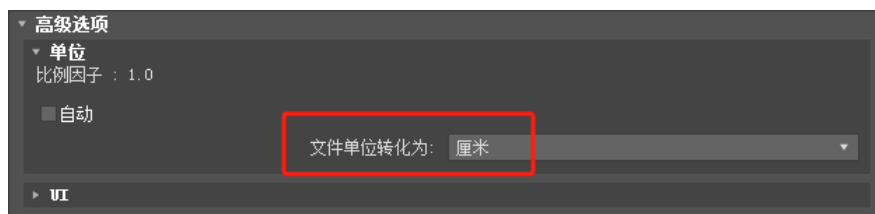
6、实验步骤

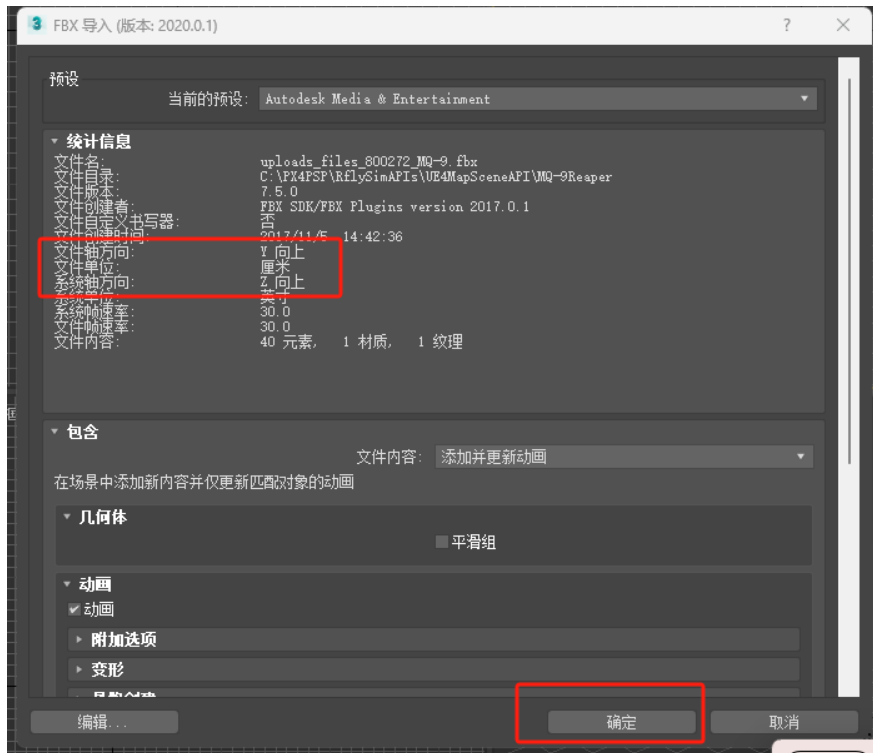
Step 1: 飞机坐标轴的归零

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。



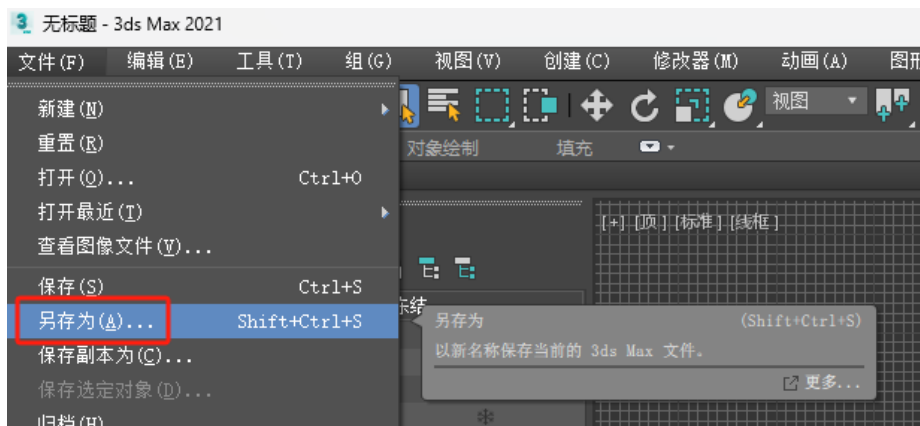
在 FBX 导入器上将文件单位转化为厘米





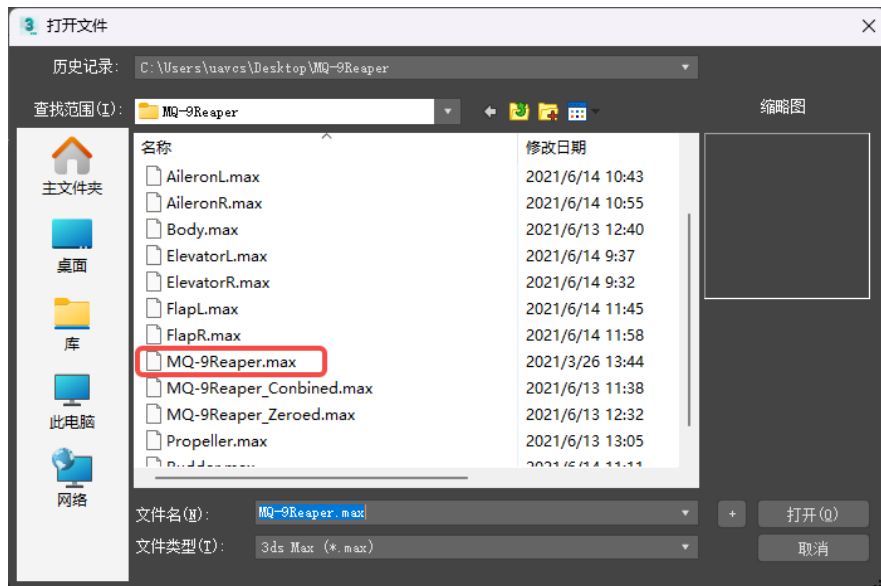
注意：

- 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

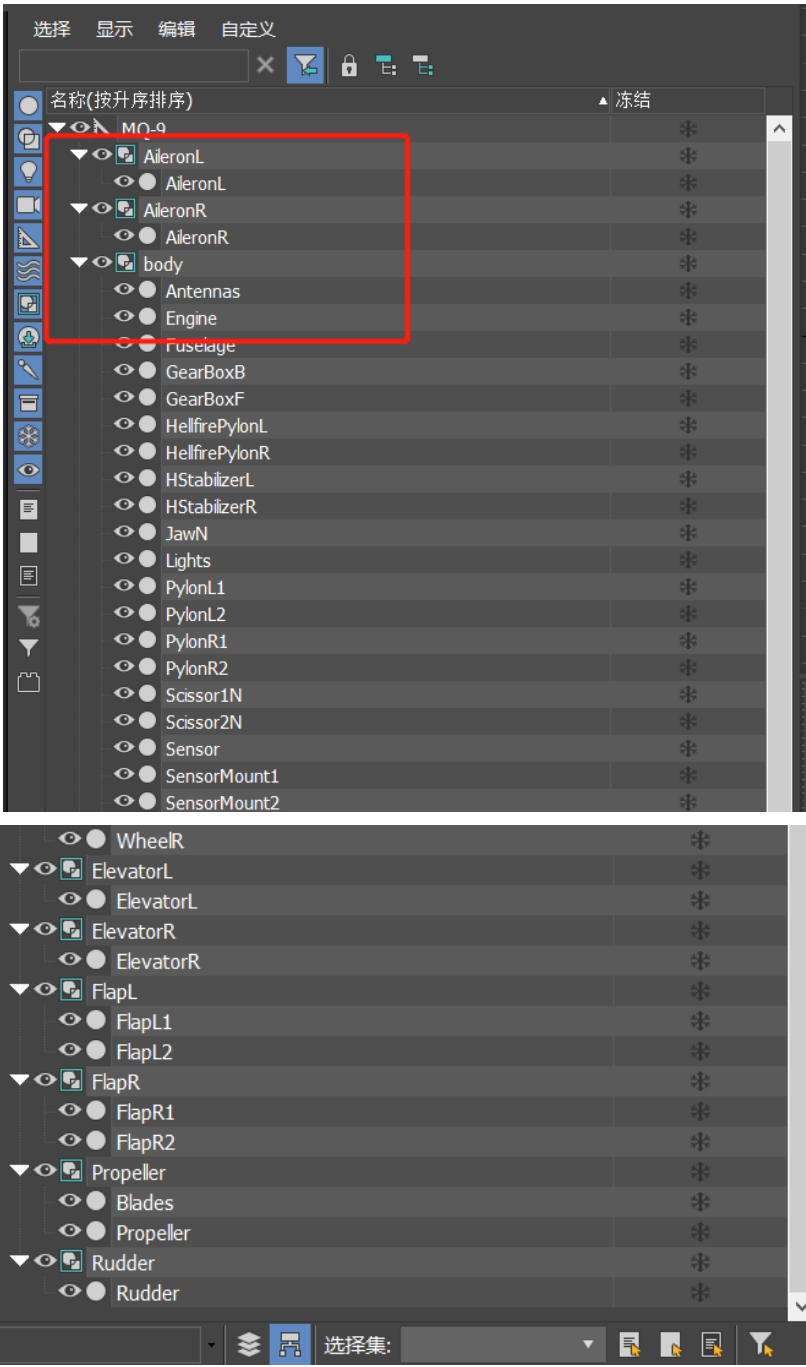
打开 MQ-9Reaper.max 文件



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

名称(按降序排序)		冻结
▼ MQ-9		*
WheelR		*
WheelN		*
展开选定项	孤立当前选择	
展开全部	结束隔离	
塌陷选定项	全部解冻	
塌陷全部	解冻场景资源管理器当前选择	
查找选定对象	冻结当前选择	
过滤器	全部取消隐藏	
	取消隐藏场景资源管理器当前选择	
	隐藏选定对象	
SensorMount		
SensorMount	选择子节点	
Sensor	反选(O)	
Scissor2N	将选定对象添加到	
Scissor1N	克隆(C)	
▼ Rudder	重命名	
Rudder	删除	
PylonR2	取消链接	
PylonR1	显示属性	
PylonL2	渲染属性	
PylonL1	属性...	
▼ Propeller		*
Propeller		*
Blades		*
Lights		*
JawN		*
HStabilizerR		*
HStabilizerL		*
HellfirePylonR		*
HellfirePylonL		*
GearBoxF		*
GearBoxB		*
Fuselage		*
▼ FlapR		*
FlapR2		*
FlapR1		*
▼ FlapL		*
FlapL2		*
FlapL1		*
Engine		*
▼ ElevatorR		*
ElevatorR		*
▼ ElevatorL		*
ElevatorL		*
Antennas		*
▼ AileronR		*
AileronR		*
▼ AileronL		*
AileronL		*

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”

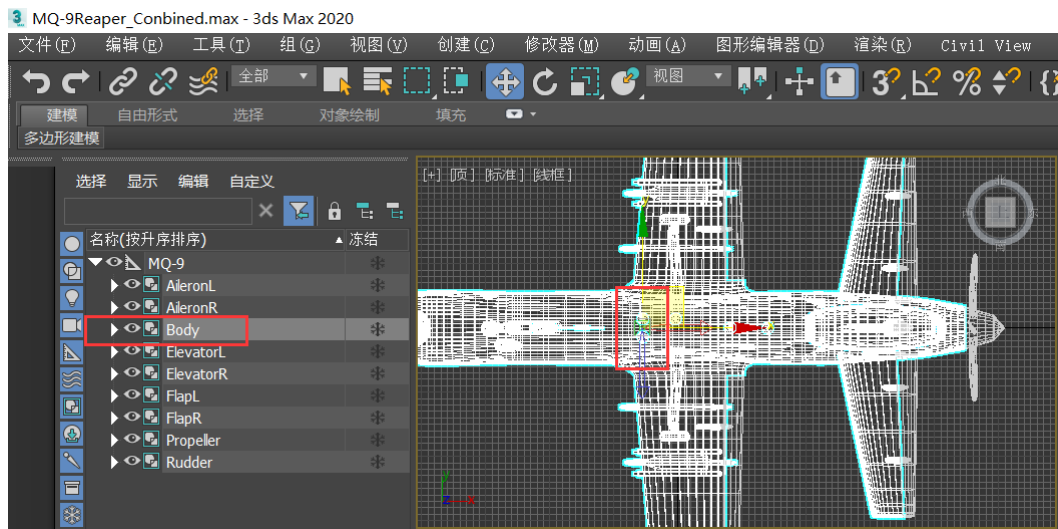
分组得到的名字为：

AileronL 左副翼
AileronR 右副翼
Body 机身
ElevatorsL 左升降舵
ElevatorsR 右升降舵
FlapL 左襟翼
FlapR 右襟翼
Propeller 螺旋桨
Rudder 方向舵

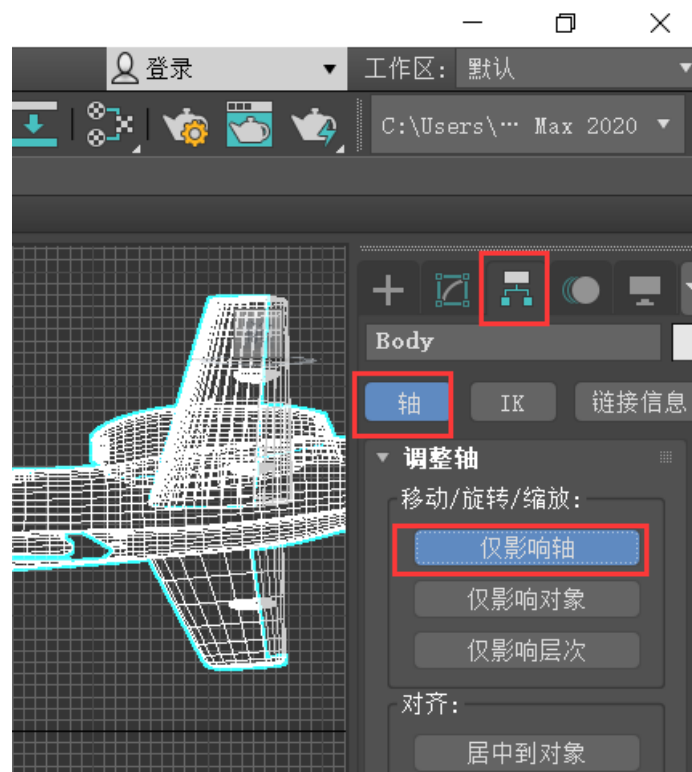
注意：XML 中八个执行器（和运动方式）的定义顺序为：1. Propeller（转动）、2. ElevatorsL（偏转）、3. ElevatorsR（偏转）、4. AileronL（偏转）、5. AileronR（偏转）、6. Rudder（偏转）、7. FlapL（偏转）、8. FlapR（偏转）。

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。



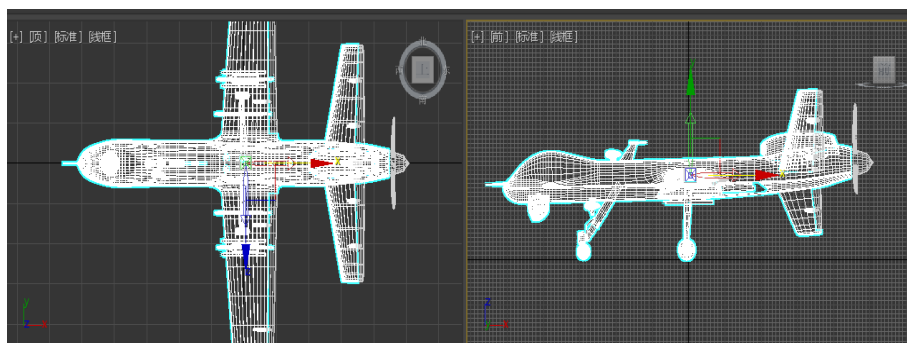
点击右侧工具栏的第三项“层次”-“轴”-“仅影响轴”，然后可以拖动机体的中心轴。



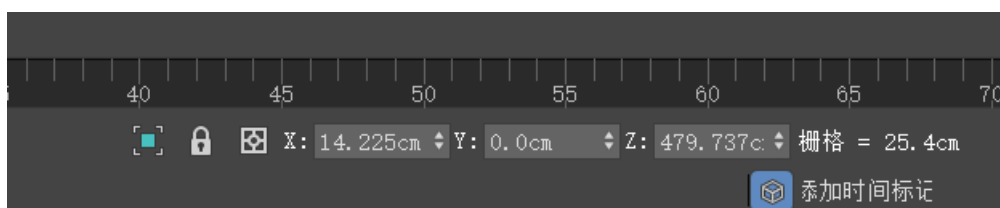
同时点击工具栏的移动图标



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

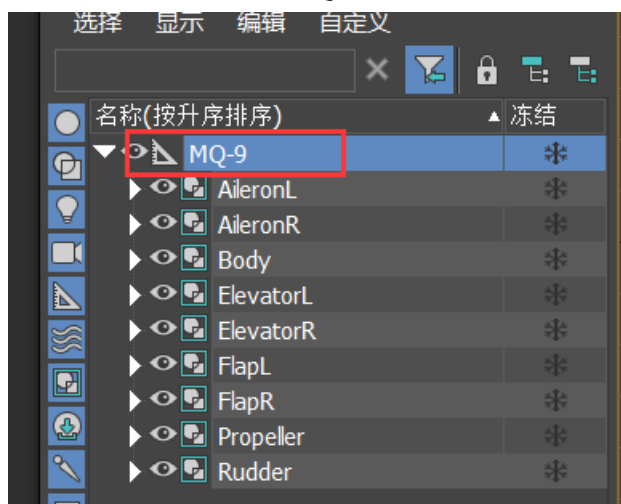


例如，这里为[14.225,0,479.737]

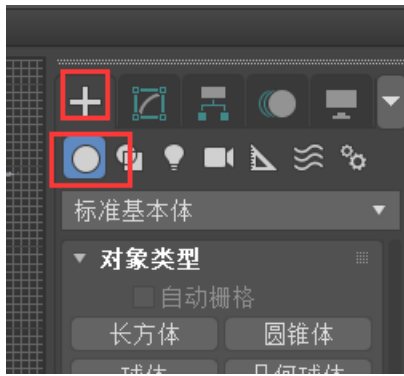
1.3 将飞机位置挪动到场景中心

打开 MQ-9Reaper_Conbined.max 文件并另存为 MQ-9Reaper_Zeroed.max。

选中顶层主物体“MQ-9”



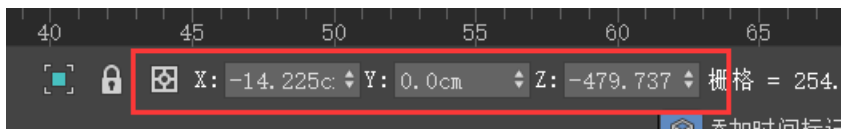
右侧工具栏进入第一个“创建”模式



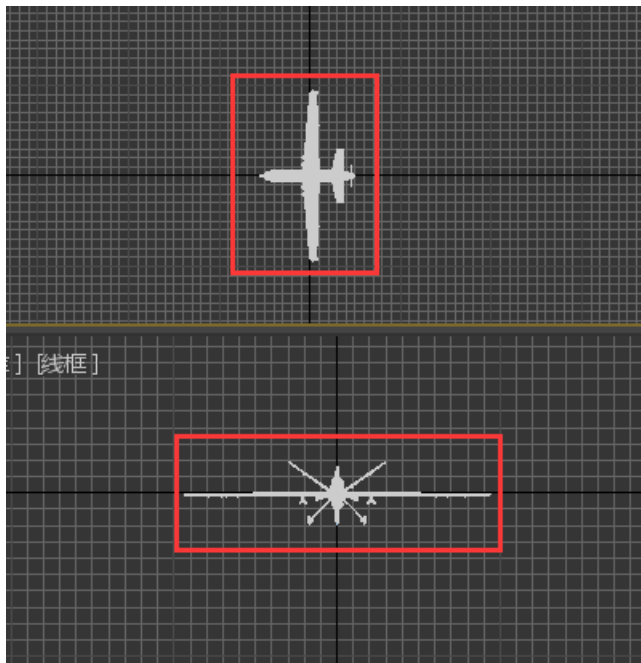
点击菜单栏移动按钮



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）

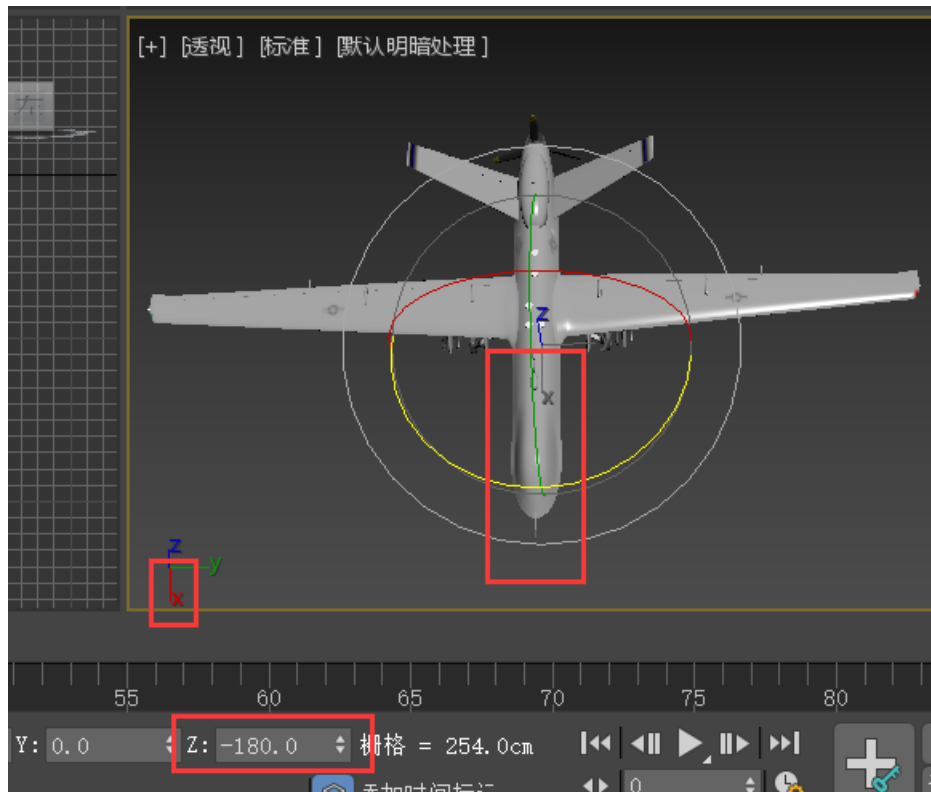


此时，还发现目前飞机的 X 轴与机头方向相反，因此需要将飞机沿 Z 轴旋转 180 度。

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

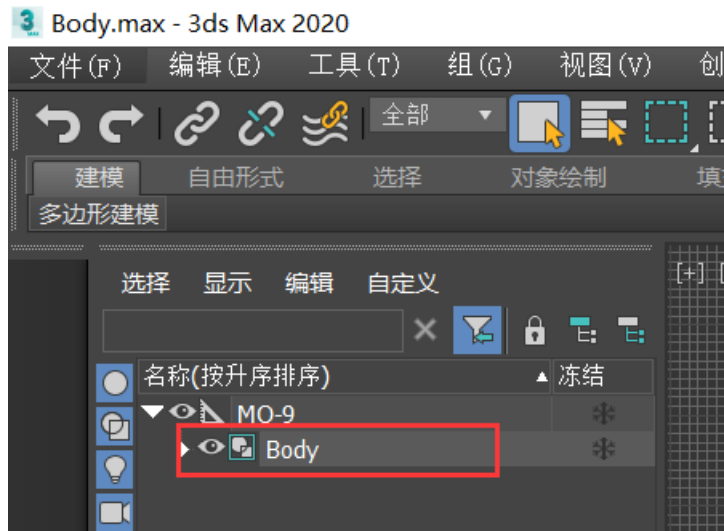
```
<Scale>
```

```
<x>0.1</x>
<y>0.1</y>
<z>0.1</z>
</Scale>
```

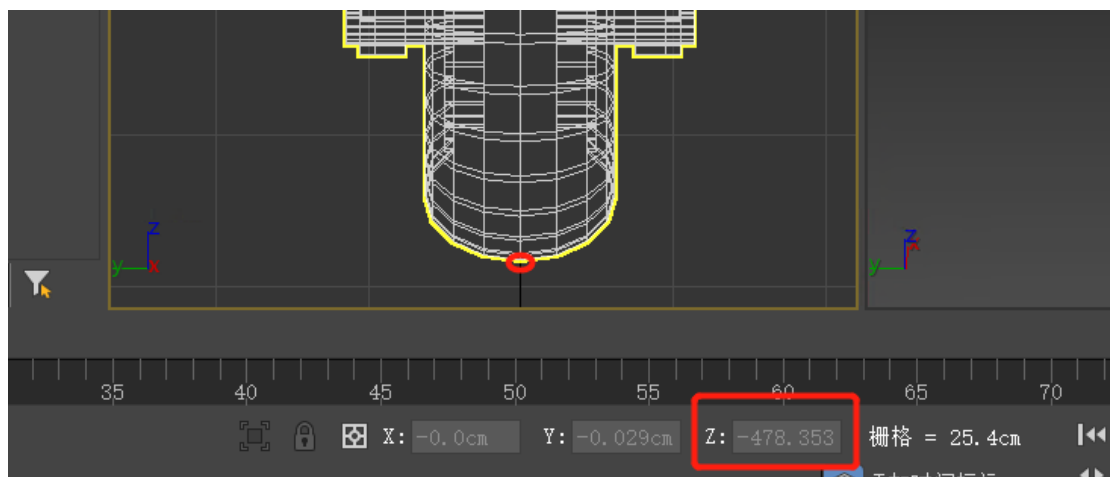
Step2: 机体模型导出

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>
  <isAnimationMesh>0</isAnimationMesh>
  <MeshPath>/Game/MQ-9Reaper/Body </MeshPath>
  <MaterialPath></MaterialPath>
  <AnimationPath></AnimationPath>
  <CenterHeightAboveGroundCm>478</CenterHeightAboveGroundCm>
  <NumberHeigthAboveCenterCm>700</NumberHeigthAboveCenterCm>
  <NumberSizeScale>50</NumberSizeScale>
</body>
```

其中，isAnimationMesh 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；NumberHeigthAboveCenterCm 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

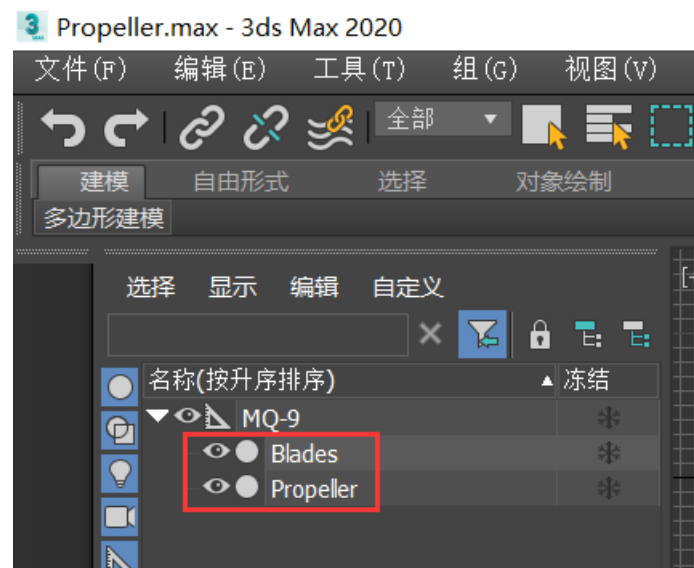
尺寸，一般根据飞机实际尺寸来，这里取 50 倍，可根据实际显示效果来调整倍数。

Step3: 螺旋桨模型的导出

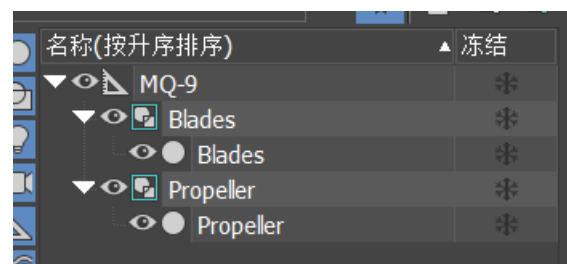
第 3.1 步，提取螺旋桨模型

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



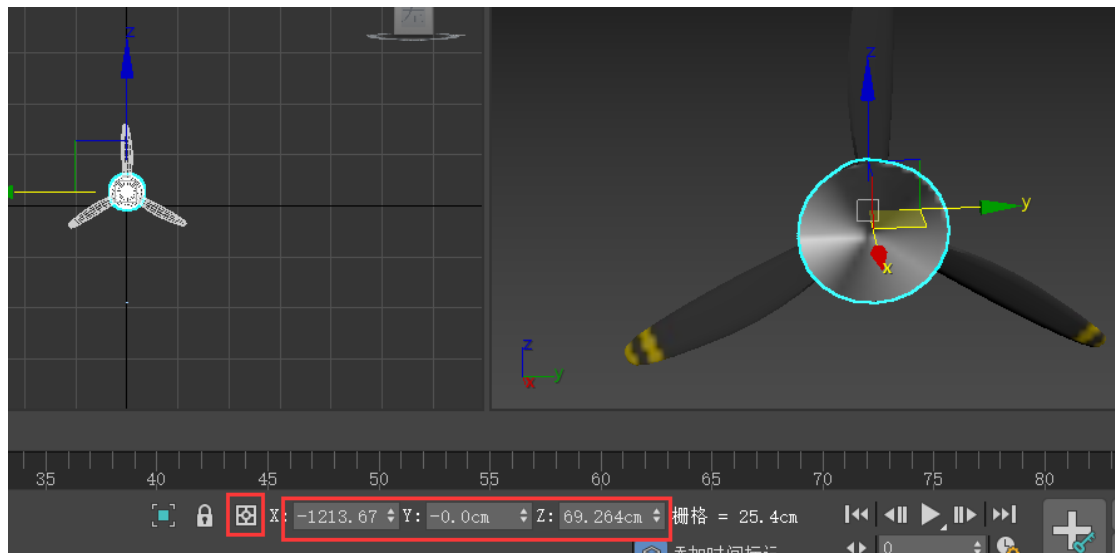
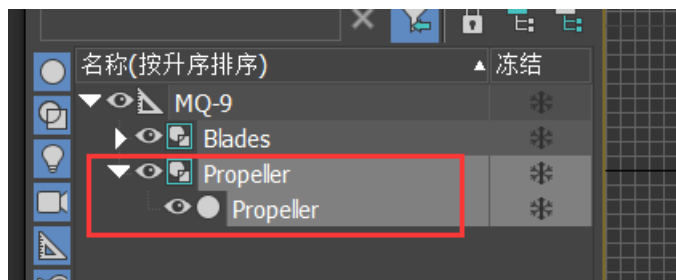
然后，如下图将两个子组件单独组成组，确保物体轴心位于物体中心



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

ax 2020

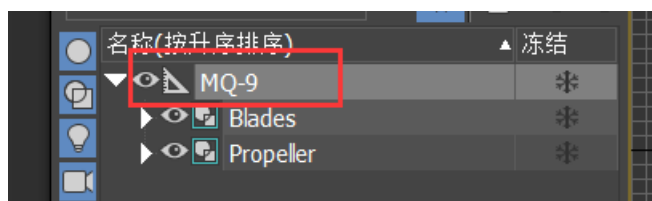




读数为-1213.674 0 69.264

步骤 3.2：调整轴心对齐

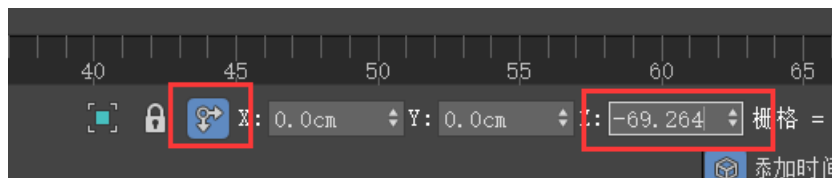
全选整体“MQ-9”，对上述坐标取反可知，需要将整体沿 X 移动 1213.674，以及沿 Z 移动-69.264 来保证螺旋桨移动到场景中心。



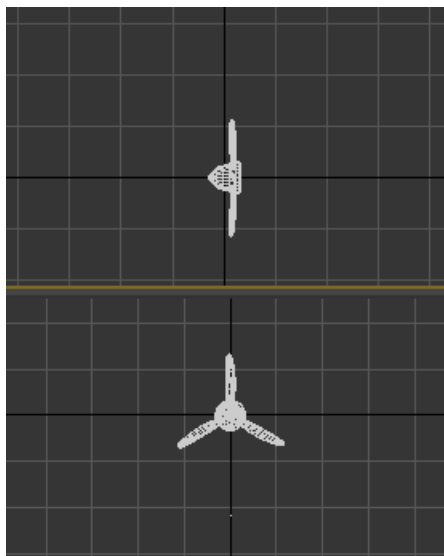
点击移动按钮，并在坐标输入栏左侧选择相对移动图标，在 X 处输入 1213.674，表示向 X 轴移动



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】

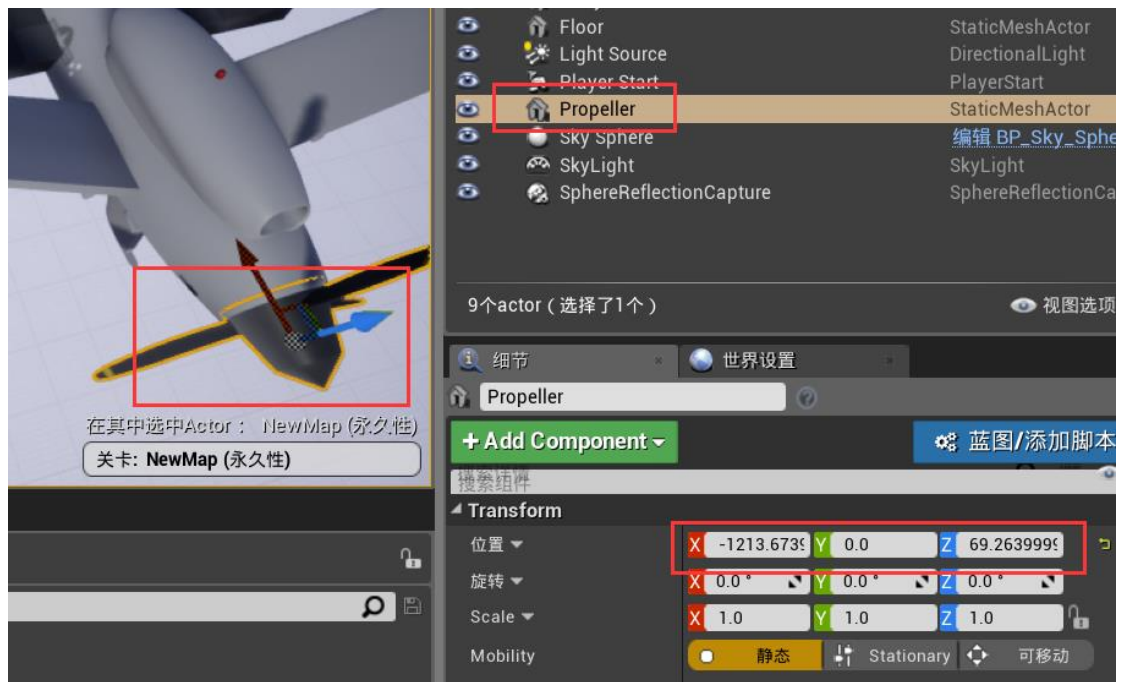
注意，UE4 的坐标 y 轴向右为正，与 3Ds Max 的 y 轴方向相反，因此 y 轴坐标需要取反。这里 y 为 0，因此不需要操作。

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>
</Actuator>
```

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。

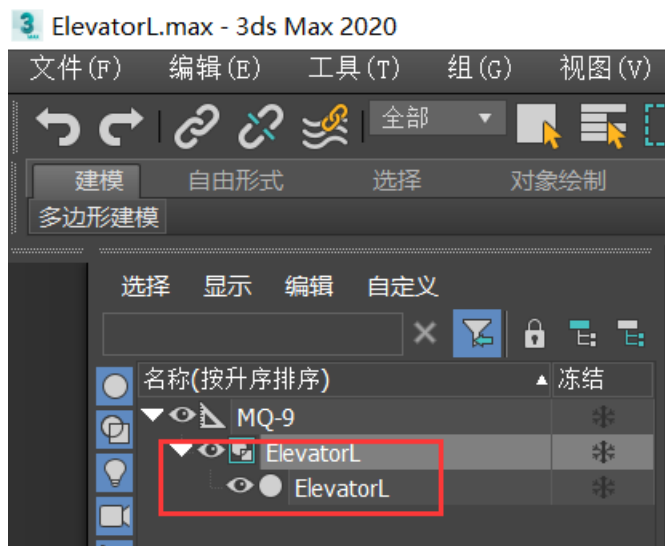
注：将机身和螺旋桨都导入 UE4 场景中，并拖入关卡，将上述位置的值输入 Propeller 的 Transform 位置框，可以预览相对位置是否正确。



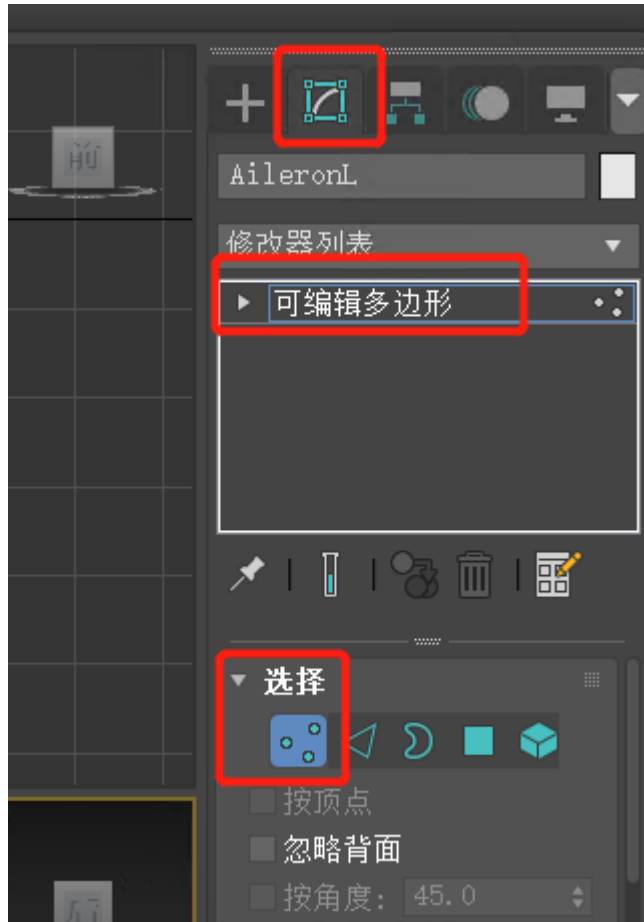
Step4: ElevatorL 左升降舵的导出

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

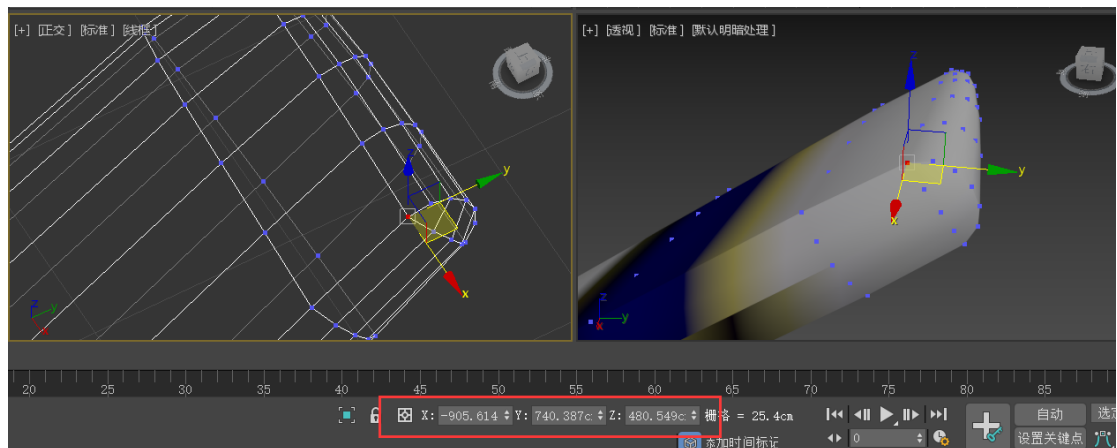
删掉其他物体, 只保留 “ElevatorL”, 并选中它



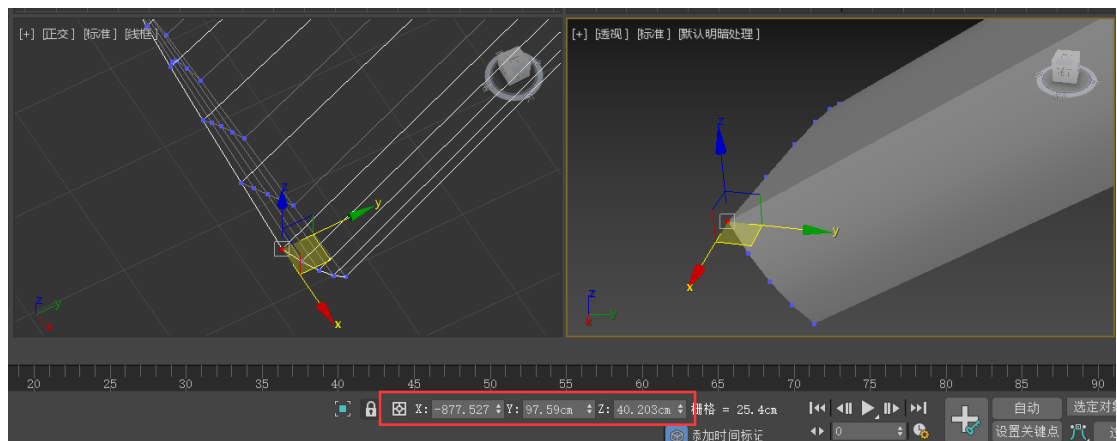
点右侧第二个工具条, 可编辑多边形, 点, 进入点选择模式



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：DD./norm(DD)）

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



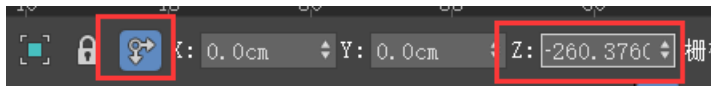
依次，向 X 轴移动 891.5705cm



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>
```

```

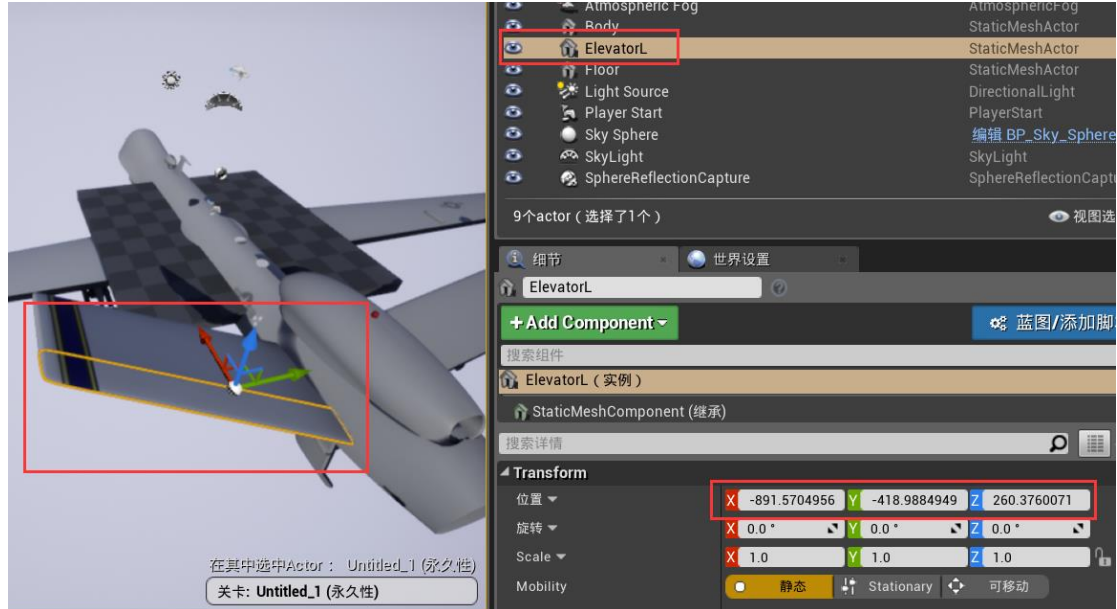
<RotationAxisVectorToBody>
  <x>0.0360</x>
  <y>0.8245</y>
  <z>-0.5648</z>
</RotationAxisVectorToBody>
<RotationModeSpinOrDefect>1</RotationModeSpinOrDefect>
</Actuator>

```

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。



Step5：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]

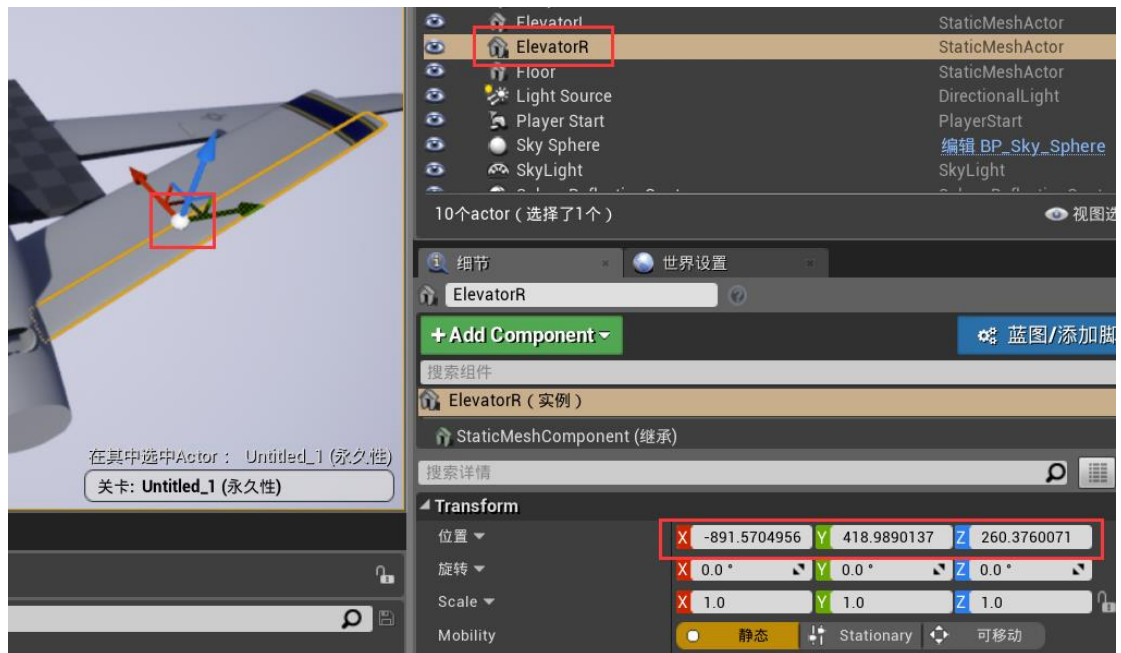
得到 XML 文件数据如下

```
<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

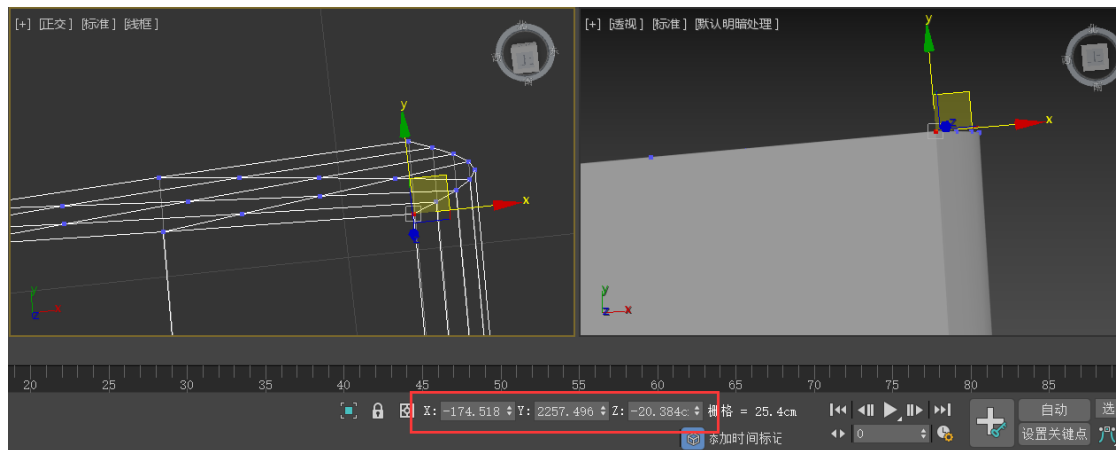


Step6: 左侧副翼 AileronL 的执行器导出方法

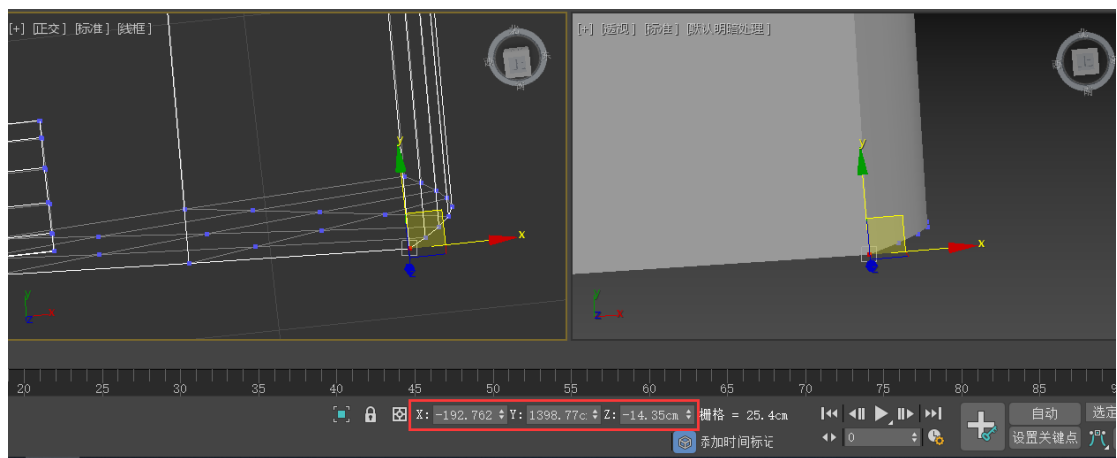
打开 MQ-9Reaper-Zeroed.max, 另存一个文件为 AileronL.max
删掉其他物体, 只保留 “AileronL”, 并选中它, 开始下面操作

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]

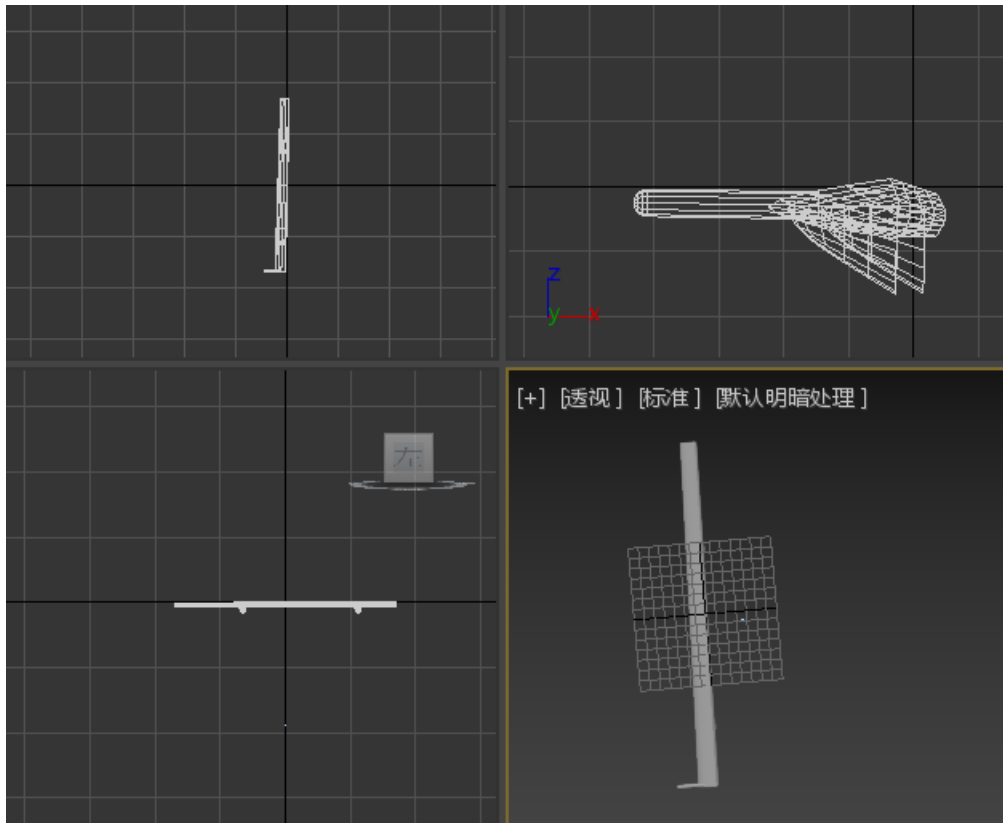


中点: [-183.64 1828.133 -17.367]

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。

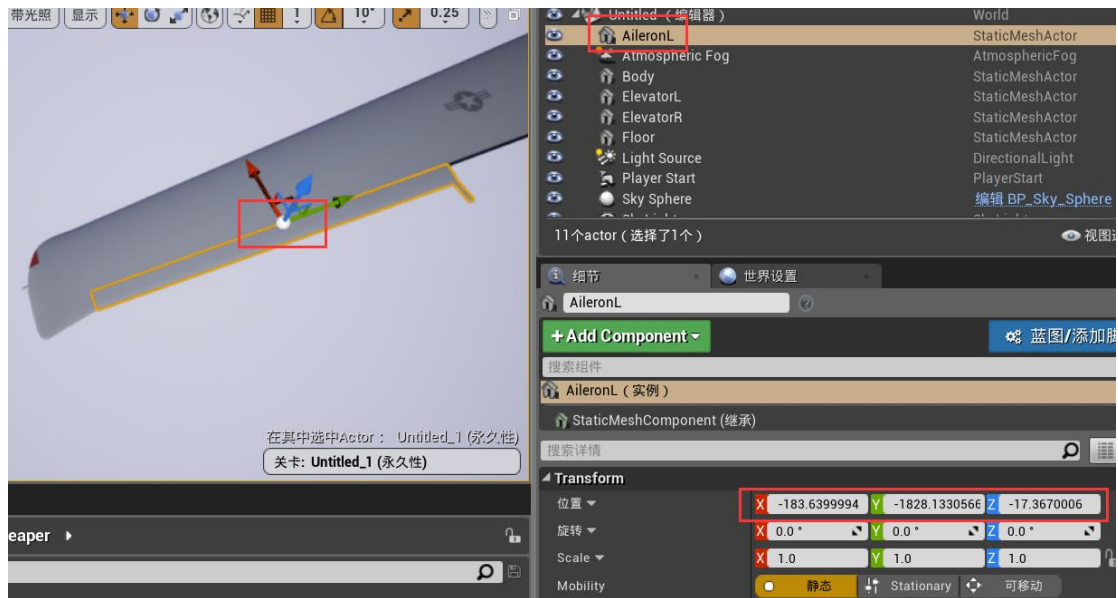
步骤 6.3：编写 XML 文件

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



Step7: 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。

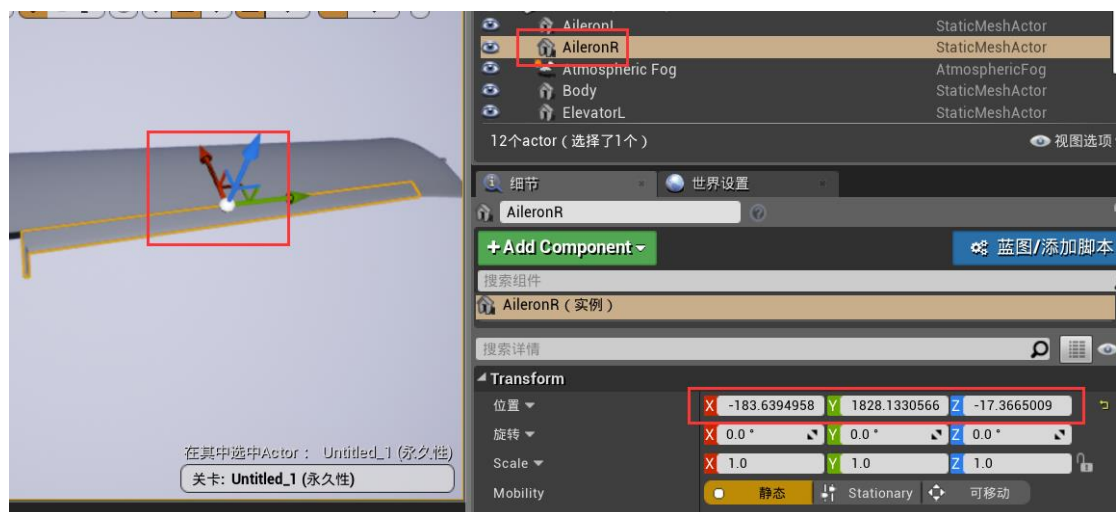
保存文件，并导出成 fbx，再导入到 UE4 中。

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: [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



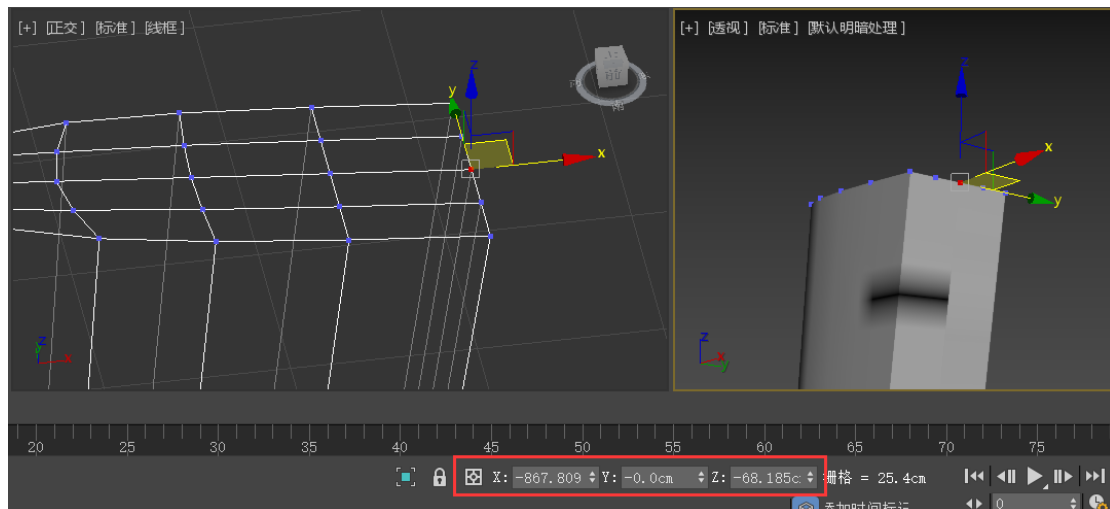
Step8: 方向舵 Rudder 的执行器导出

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

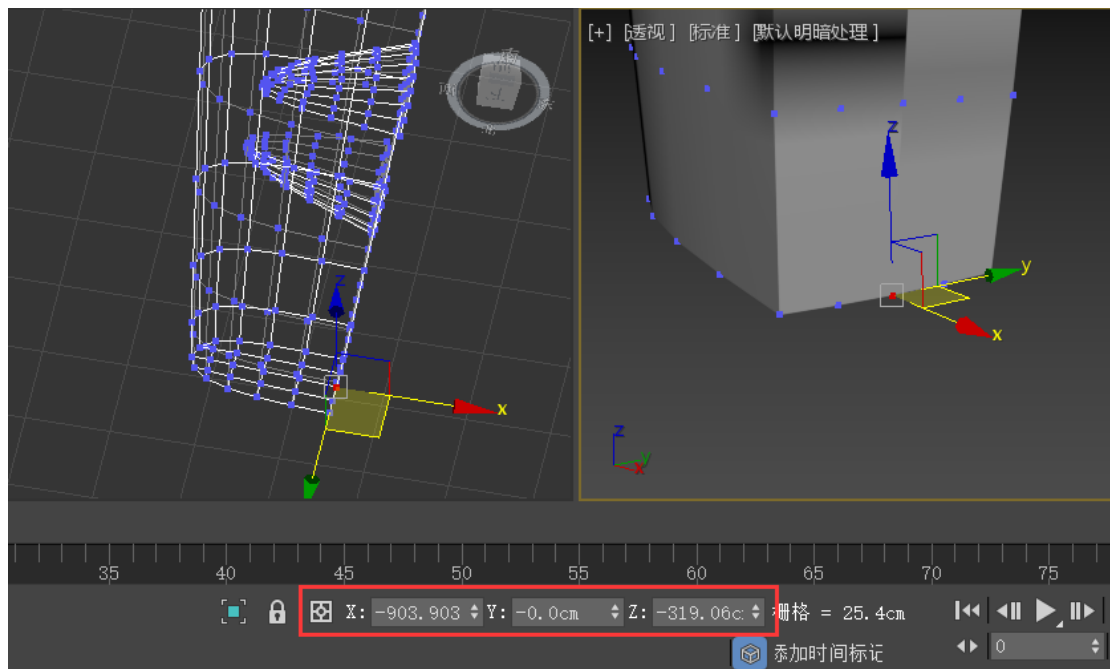
删掉其他物体，只保留“Rudder”，并选中它，开始下面操作

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]

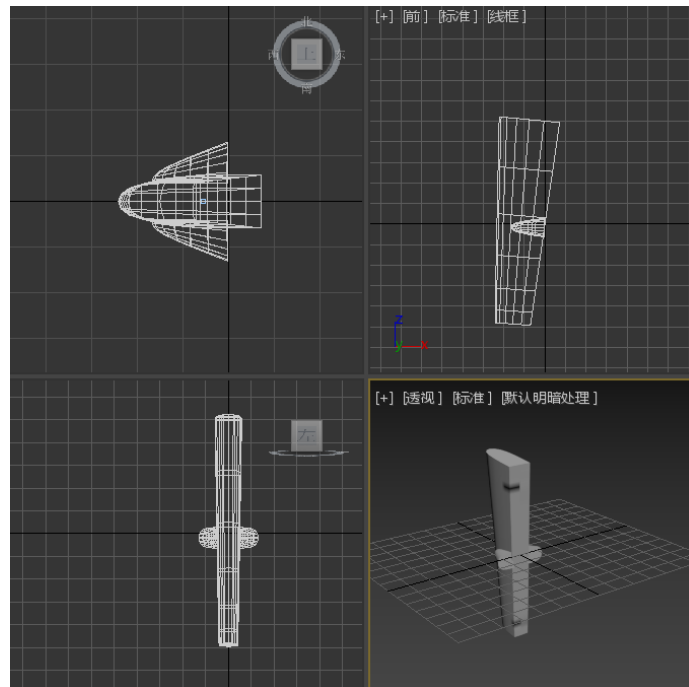


中点：[-885.8560 0 -193.6225]

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]

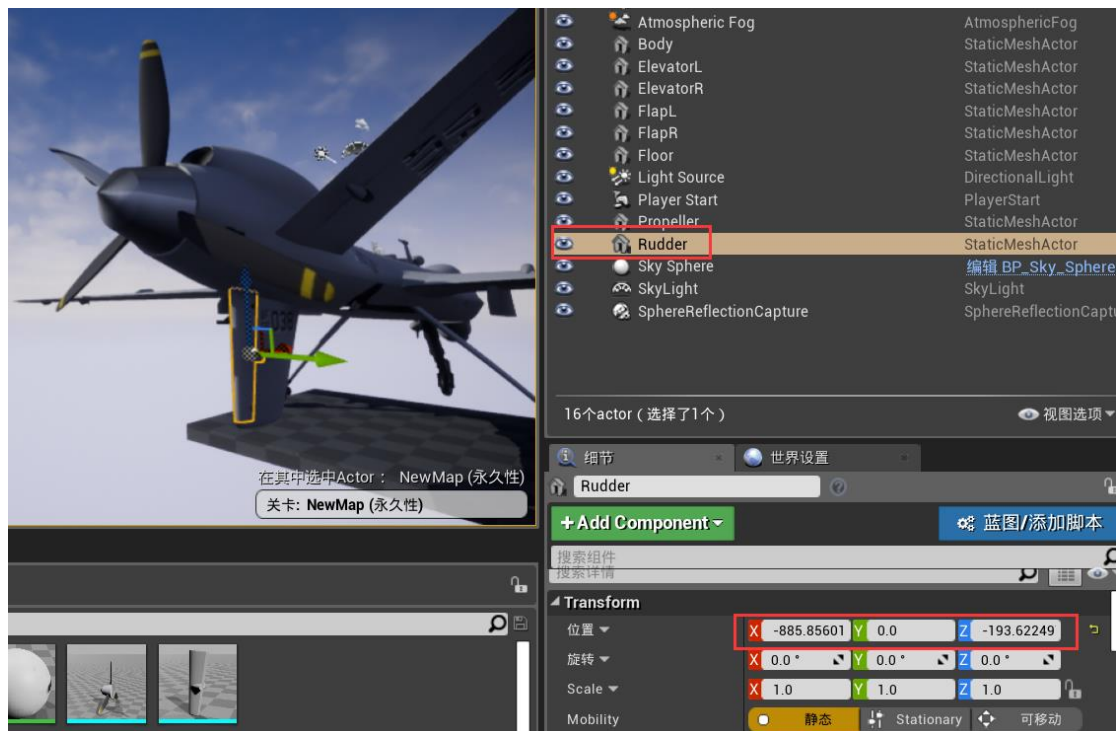
```
<Actuator>
  <MeshPath>/Game/MQ-9Reaper/Rudder</MeshPath>
  <MaterialPath></MaterialPath>
  <RelativePosToBodyCm>
    <x>-885.8560</x>
    <y>0</y>
    <z>-193.6225</z>
  </RelativePosToBodyCm>
  <RelativeAngEulerToBodyDeg>
    <roll>0</roll>
    <pitch>0</pitch>
    <yaw>0</yaw>
  </RelativeAngEulerToBodyDeg>
  <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



Step9: 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

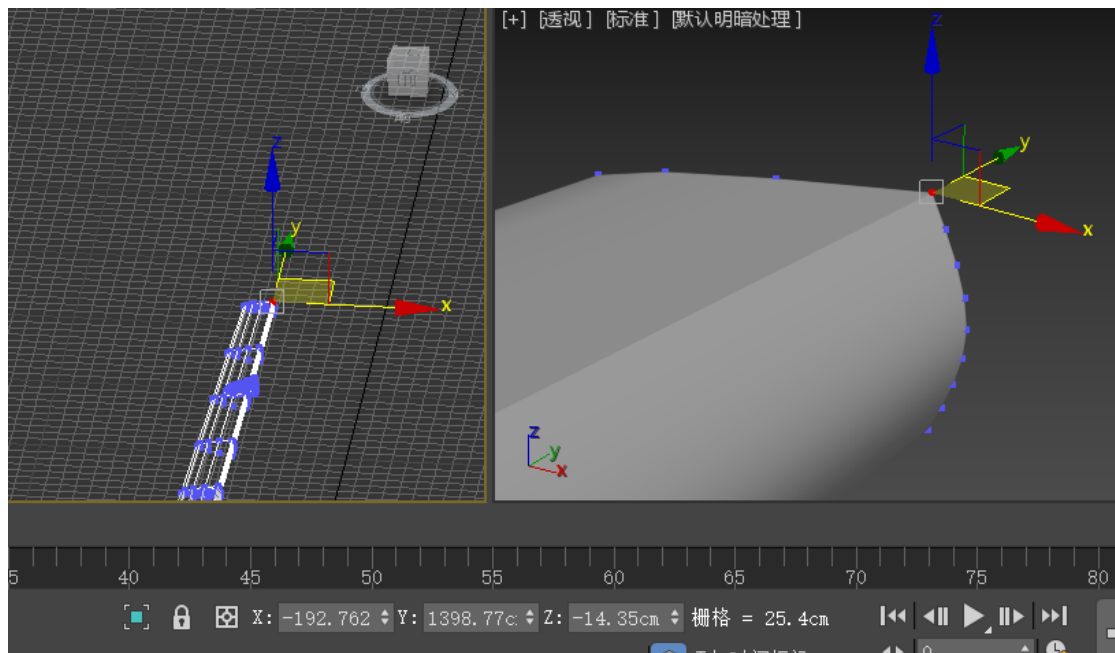
删掉其他物体，只保留“FlapL”，

步骤 9.1: 获取转轴信息

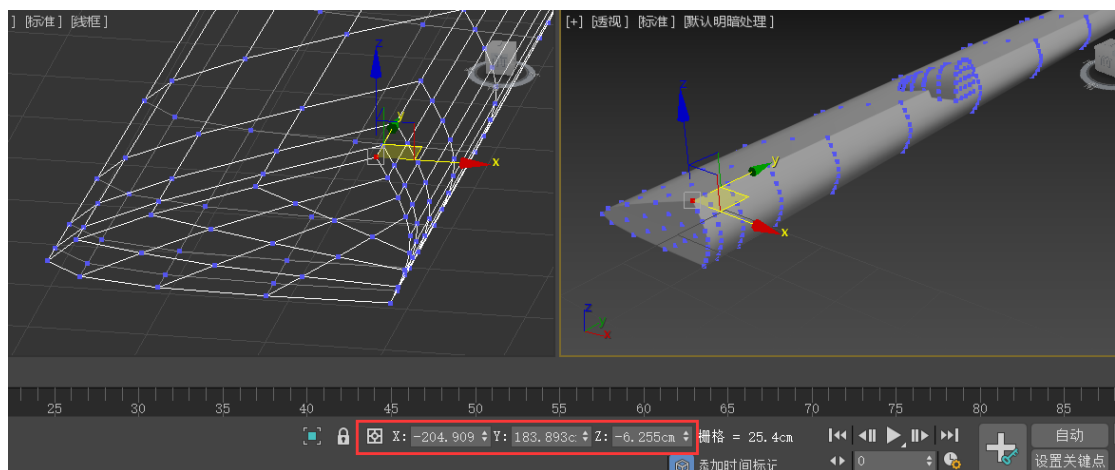
由于 FlapL 中包含了两个子物体，FlapL1 和 FlapL2，需要将 FlapL 先解组，才能进行点拾取操作。

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]

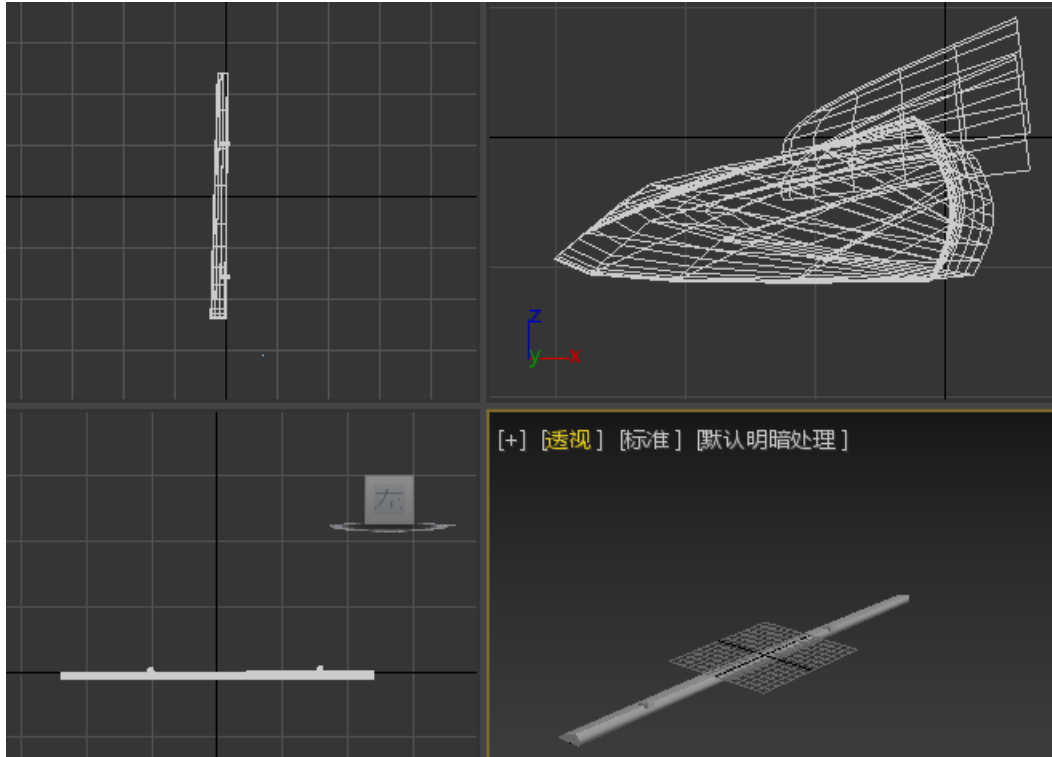


中点: [-198.8355 791.3315 -10.3025]

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。



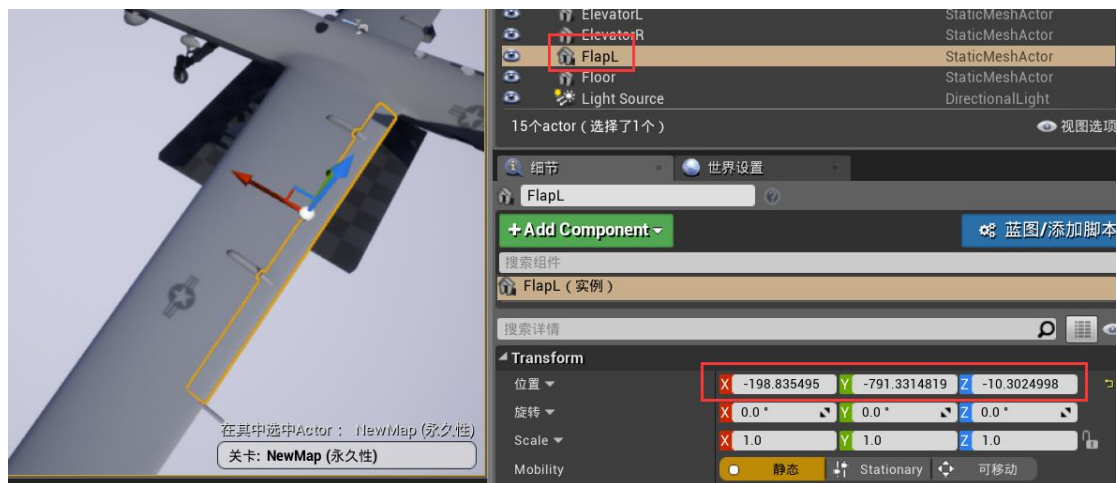
保存文件，并导出成 **fbx**，再导入到 UE4 中。

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：[-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



Step10: 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。

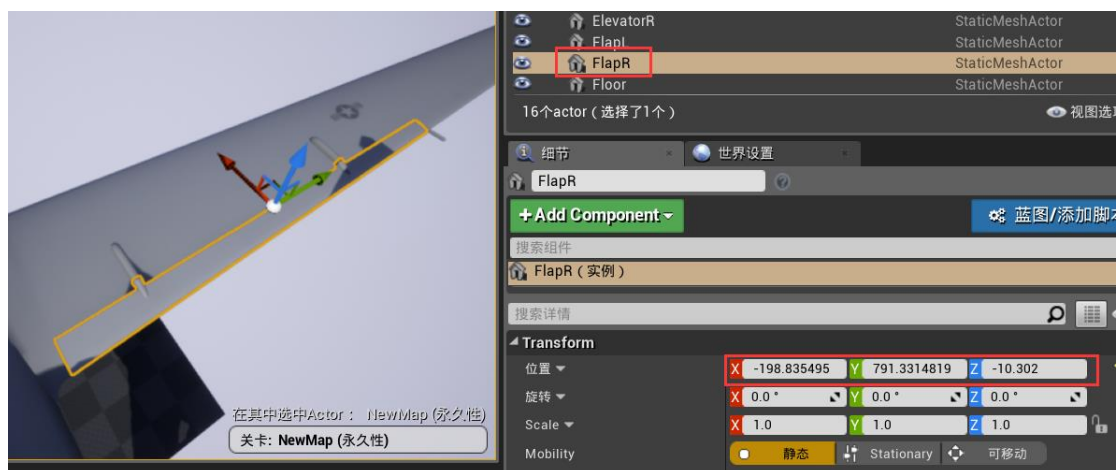
保存文件，并导出成 fbx，再导入到 UE4 中。

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：[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>
```

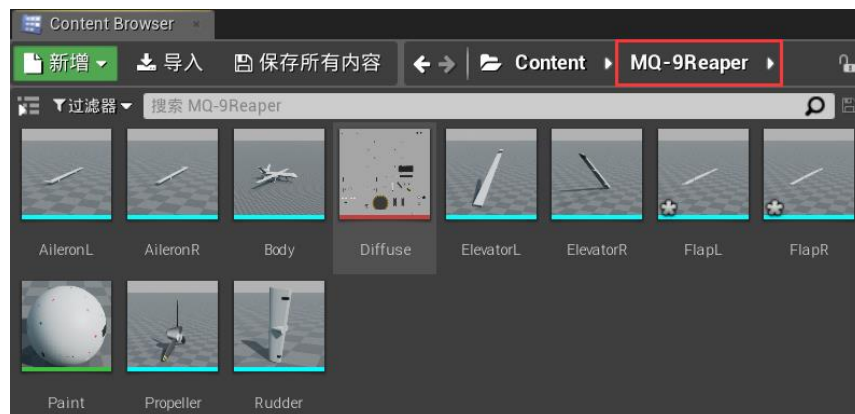
步骤 10.4：导入 UE4，拖入关卡中，输入位置坐标，查看拼合效果如下



Step11: 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

```
<?xml version="1.0"?>
<vehicle>
  <ClassID>100</ClassID>
  <DisplayOrder>900</DisplayOrder>
  <Name>MQ-9Reaper</Name>
  <Scale>
    <x>0.1</x>
    <y>0.1</y>
    <z>0.1</z>
  </Scale>
  <AngEulerDeg>
    <roll>0</roll>
    <pitch>0</pitch>
    <yaw>0</yaw>
  </AngEulerDeg>

  <body>
    <isAnimationMesh>0</isAnimationMesh>
    <MeshPath>/Game/MQ-9Reaper/Body </MeshPath>
    <MaterialPath></MaterialPath>
```

```
<AnimationPath></AnimationPath>
<CenterHeightAboveGroundCm>478</CenterHeightAboveGroundCm>
<NumberHeightAboveCenterCm>700</NumberHeightAboveCenterCm>
<NumberSizeScale>50</NumberSizeScale>
</body>
```

```
<ActuatorList>
```

```
<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>
```

```
</Actuator>
```

```
<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>
```

```
<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>
```

```
<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>
```

```
<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>

<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>
```

<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>
```

<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>
```

```
</ActuatorList>
```

```
<OnboardCameras>
```

```
    <camera>
        <name>Chase_Camera</name>
        <RelativePosToBodyCm>
            <x>-2500</x>
            <y>0</y>
            <z>500</z>
        </RelativePosToBodyCm>
        <RelativeAngEulerToBodyDeg>
            <roll>0</roll>
            <pitch>0</pitch>
            <yaw>0</yaw>
        </RelativeAngEulerToBodyDeg>
```

```
    </camera>
```

```
    <camera>
        <name>Front_Camera</name>
        <RelativePosToBodyCm>
            <x>1000</x>
            <y>0</y>
            <z>0</z>
        </RelativePosToBodyCm>
        <RelativeAngEulerToBodyDeg>
            <roll>0</roll>
            <pitch>0</pitch>
            <yaw>0</yaw>
        </RelativeAngEulerToBodyDeg>
```

```
    </camera>
```

```
    <camera>
        <name>Back_Camera</name>
        <RelativePosToBodyCm>
            <x>-1000</x>
            <y>0</y>
            <z>0</z>
        </RelativePosToBodyCm>
        <RelativeAngEulerToBodyDeg>
            <roll>0</roll>
            <pitch>0</pitch>
            <yaw>180</yaw>
        </RelativeAngEulerToBodyDeg>
```

```
    </camera>
```

```
</camera>
```

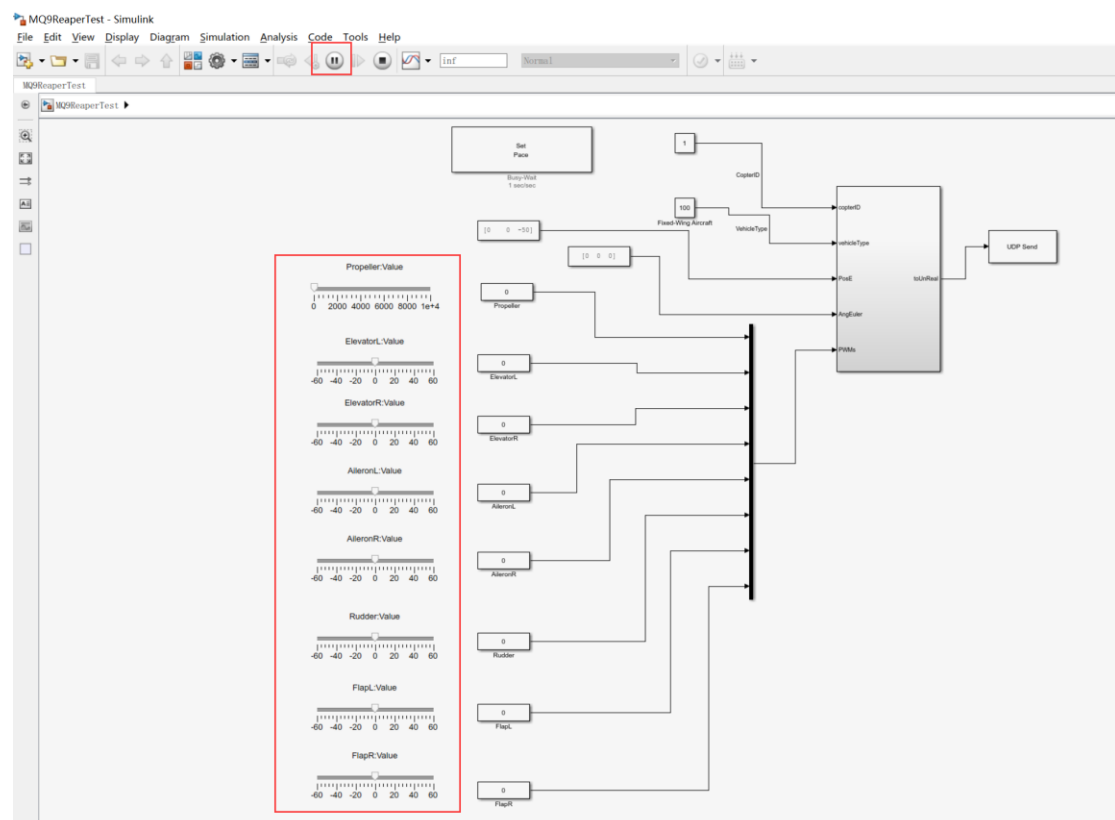
```
<name>Right_Camera</name>
<RelativePosToBodyCm>
  <x>0</x>
  <y>2000</y>
  <z>0</z>
</RelativePosToBodyCm>
<RelativeAngEulerToBodyDeg>
  <roll>0</roll>
  <pitch>0</pitch>
  <yaw>90</yaw>
</RelativeAngEulerToBodyDeg>
</camera>
<camera>
  <name>Left_Camera</name>
  <RelativePosToBodyCm>
    <x>0</x>
    <y>-2000</y>
    <z>0</z>
  </RelativePosToBodyCm>
  <RelativeAngEulerToBodyDeg>
    <roll>0</roll>
    <pitch>0</pitch>
    <yaw>-90</yaw>
  </RelativeAngEulerToBodyDeg>
</camera>
<camera>
  <name>Down_Camera</name>
  <RelativePosToBodyCm>
    <x>0</x>
    <y>0</y>
    <z>-500</z>
  </RelativePosToBodyCm>
  <RelativeAngEulerToBodyDeg>
    <roll>0</roll>
    <pitch>-90</pitch>
    <yaw>0</yaw>
  </RelativeAngEulerToBodyDeg>
</camera>
<camera>
  <name>Up_Camera</name>
  <RelativePosToBodyCm>
    <x>0</x>
    <y>0</y>
    <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

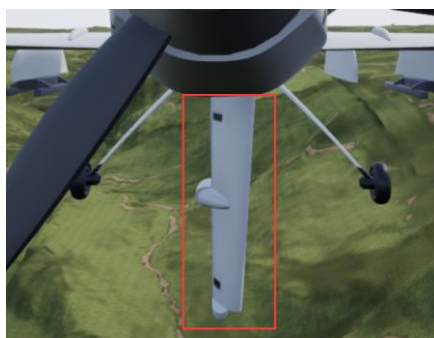
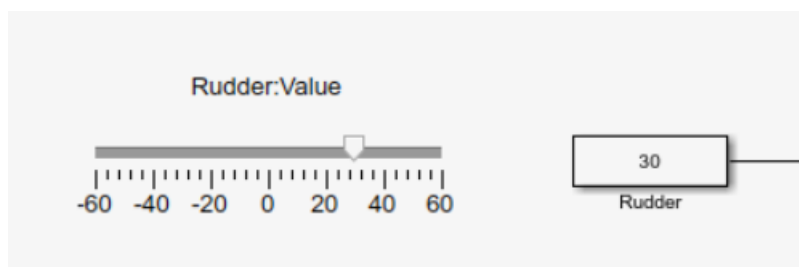
打开一个 RflySim3D 程序，再用 MATLAB 打开“MQ9ReaperTest.slx”（例程见 UE4 Cooked\MQ-9Reaper\MQ9ReaperTest.slx 文件），运行之后可以看到一个固定翼无人机在场景中。



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



Step12: 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。



7、参考资料

- [1]. XML 文件规则 ([见 API 文档](#))
- [2]. RflySim3D 快捷键接口总览 ([见 API 文档](#))

[3]. RflySim3D 控制台命令接口总览 ([见 API 文档](#))

8、常见问题

1. 无