

| 基于RflySim3D接口的自定义多旋翼模型加载实验

| 1. 实验目的

了解如何导入基于XML和静态网格的自定义多旋翼模型的流程。

| 2. 实验要求

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

| 3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\3.RflySim3DUE\1.BasicExps\e1_CusLoadDroneyeeX680](#)

- [Intro.pdf](#)：自定义多旋翼三维模型加载实验原理
- [VehicleModel](#)：其内为待处理的模型数据
- [DroneyeeX680](#)：其内为处理完成可直接导入的模型数据

| 4. 实验内容或步骤

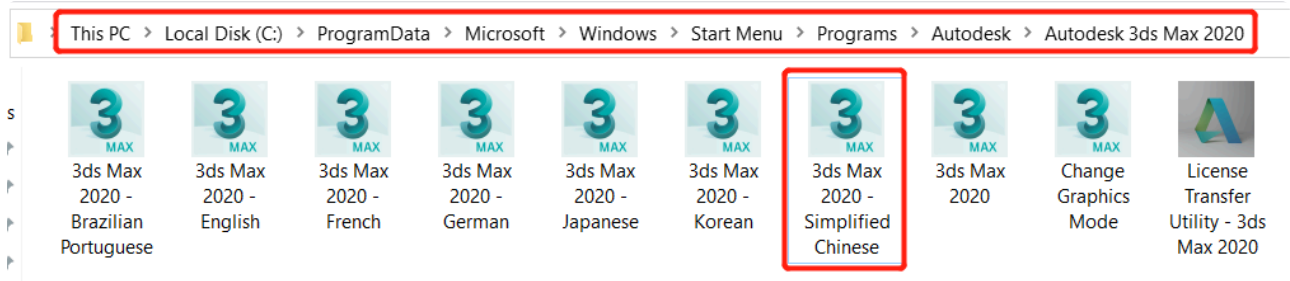
| 4.1 在三维建模软件中处理模型（选做）

说明：本步骤以3ds Max为例进行演示，若已经熟悉相关流程，您也可以使用其他支持的三维建模软件（如Maya、Blender、Cinema 4D等）或CAD软件（如SOLIDWORKS、

Inventor等) 进行模型处理。不同软件的具体操作方法会有所差异, 但核心流程和要求相同。

步骤1: 通过3dsMax打开模型 (其他软件类似)

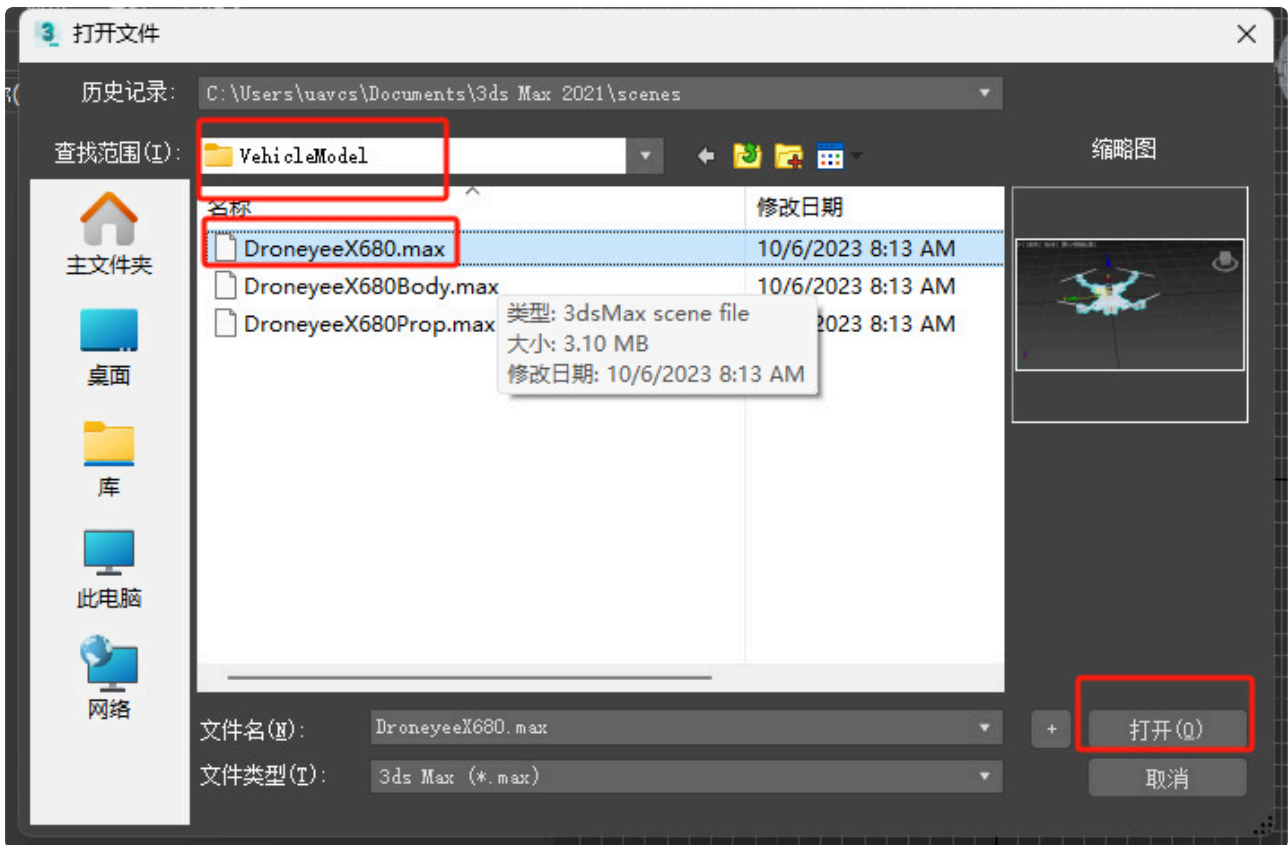
完成3dsMax安装后, 通过如下方式打开3dsMax中文版



打开3dsMax后, 通过下图方式选择打开源文件所在目录 (这里是打开max文件, 若使用其它三维格式, 应通过文件-导入的方式)



然后在弹出窗口中找到本实验目录下的"VehicleModel"目录并找到"DroneyeeX680.max", 选择打开



步骤2: 模型坐标归零

在3Dsmax中选中所有组件，点击菜单栏"组"->"组"，就可以将所有组件弄成一个整体，这时候可以调整飞机的位置，姿态，轴的位置，轴的方向等。需要满足以下要求：

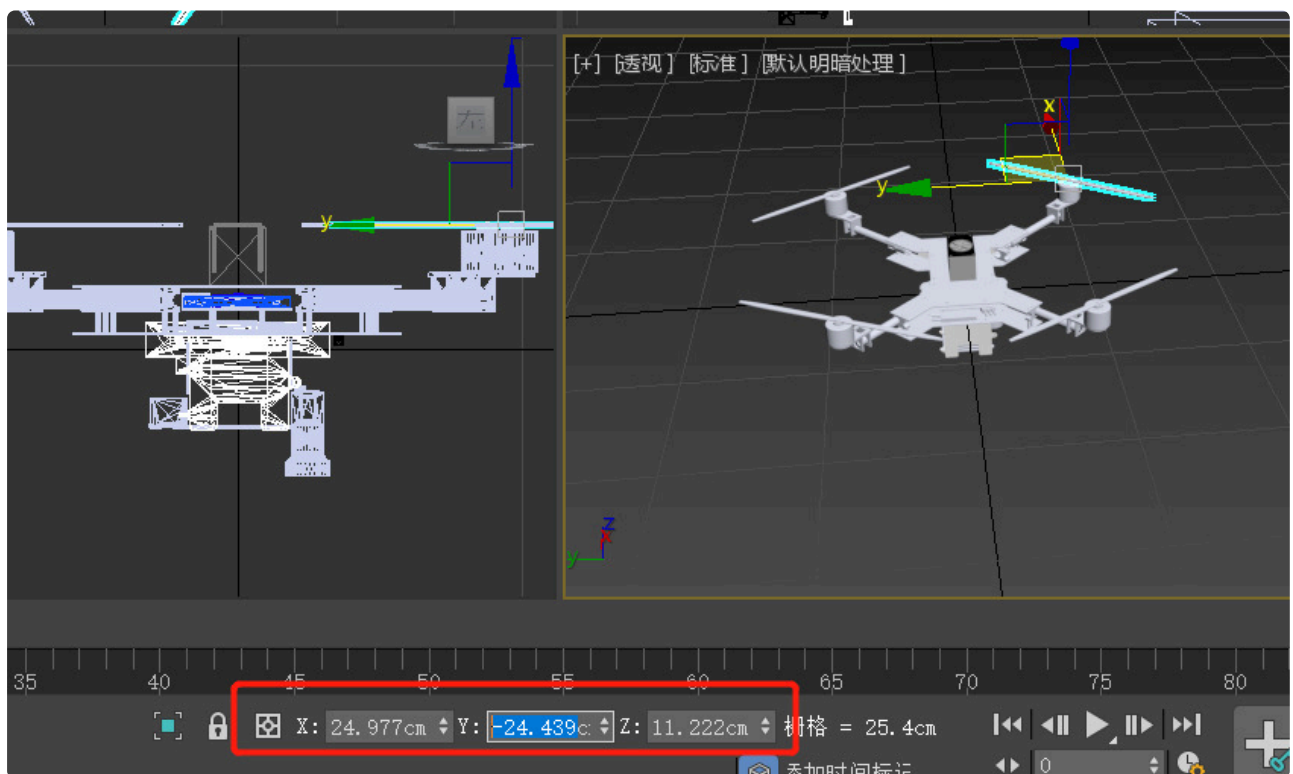
- 1) 确保机头方向指向3Ds Max的x轴正方向，机顶指向z轴正方向（向上）；
- 2) 确保飞机的质心在3Ds Max的中心；
- 3) 确保模型中没有隐形（不显示的）组件，如果有需要删除。

如下图，点击工具栏"移动"按钮，再点击选中机身或螺旋桨对象，就可以在下方状态栏中读取对象的三维坐标（将坐标都输入为0就可以将对象挪到中心）。



步骤3: 记录机身和螺旋桨相对位置

点击机身就可以看到机身的三维坐标 $(x,y,z) = [-0.449, -0.363, 0]$



右上螺旋桨 $[24.977, -24.439, 11.222]$ ，左下螺旋桨 $[-25.99, 22.528, 11.022]$ ，
左上螺旋桨 $[24.977, 24.693, 11.022]$ ，右下螺旋桨 $[-25.99, -22.274, 11.022]$ 。

将各个螺旋桨的坐标减去机体的坐标，就可以得到各个螺旋桨相对机体中心的三维坐标。由于3DsMax是采用前左上的右手坐标系，UE采用前右上的左手坐标系，因此需要对y轴进行反向。

这样就得到了右上、左下、右下、左上的坐标序列为：[25.4260,24.0760,11.2220]，[-25.5410,-22.8910,11.0220]，[25.4260,-25.0560,11.0220],[-25.5410,21.9110,11.0220]，这几个坐标值留着备用。然后，记录飞机质心到地面（机体最低端）的距离，这里大约取8cm。

步骤4: 分别导出机身和螺旋桨fbx（适用于各种三维软件）

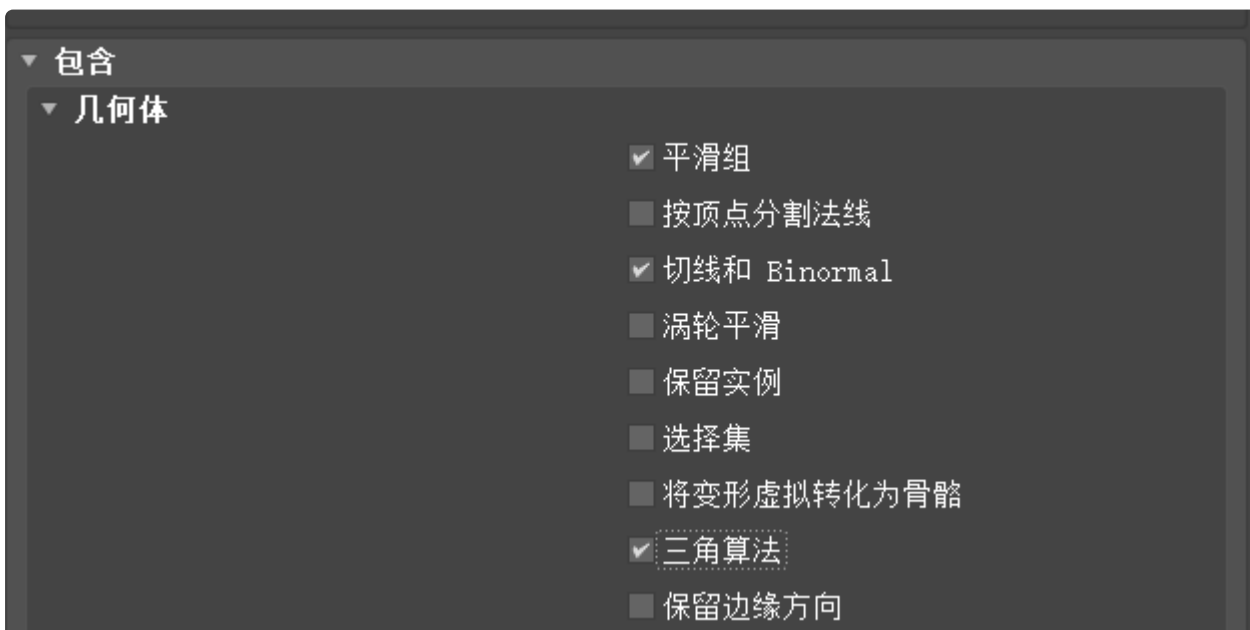
将DroneyeeX680.max复制两份，分别命名为DroneyeeX680Body.max和DroneyeeX680Prop.max。在DroneyeeX680Body.max文件中移除四个螺旋桨，并将机身移动到3DsMax的坐标中心；在DroneyeeX680Prop.max中移除机体组件和三个螺旋桨，只留下一个，然后移动到3Dsmax坐标中心。

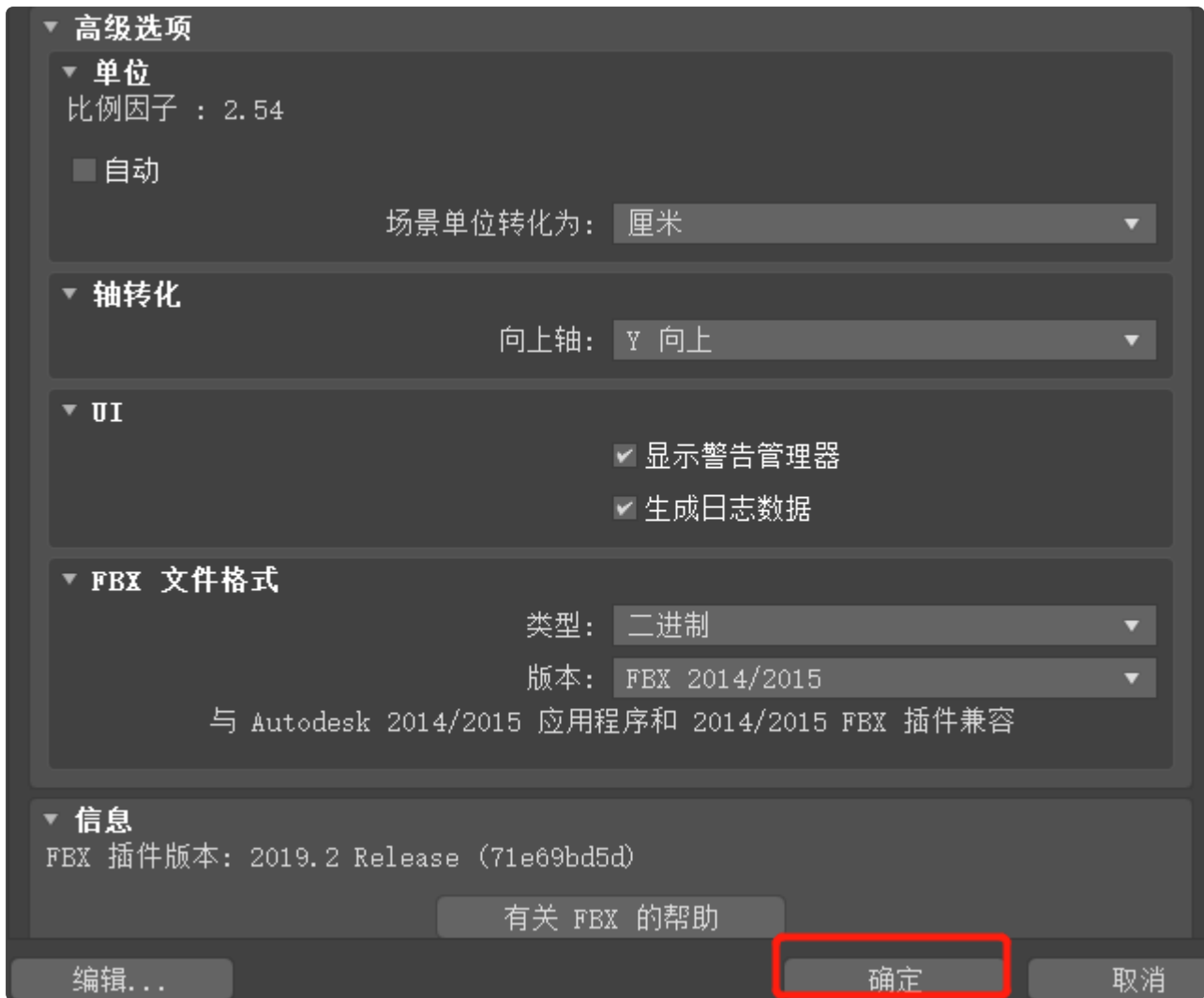
导出得到FBX模型文件，分别点击3DsMax的菜单-文件-导出-导出，按下图所示配置，导出得到DroneyeeX680Body.FBX和DroneyeeX680Prop.FBX文件

其他软件导出FBX的方法：

- **Maya**: File > Export All > FBX Export, 注意坐标轴设置为Z-Up
- **Blender**: File > Export > FBX, Forward设置为-Y Forward, Up设置为Z Up
- **Cinema 4D**: File > Export > FBX, 确保坐标系正确转换
- **SOLIDWORKS**: 通过Datasmith插件导出或使用第三方FBX导出插件

关键导出配置如下图所示。注：摄像机、动画、灯光都不要勾选

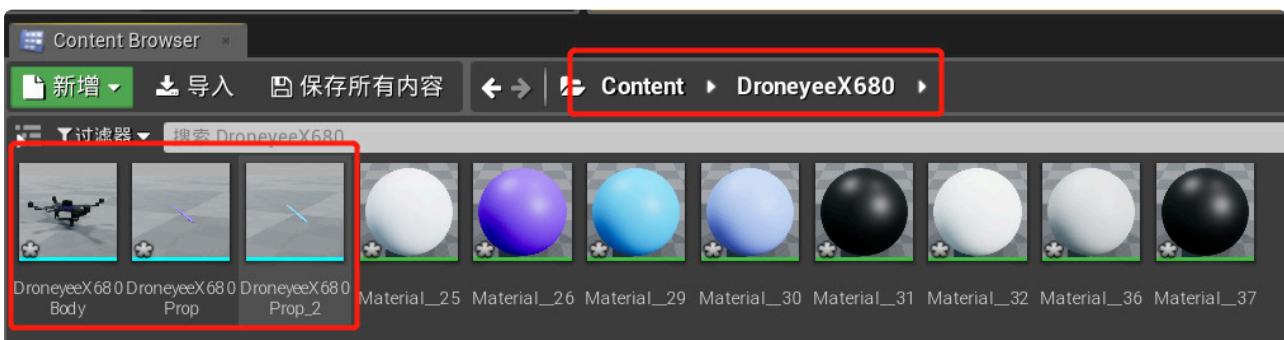


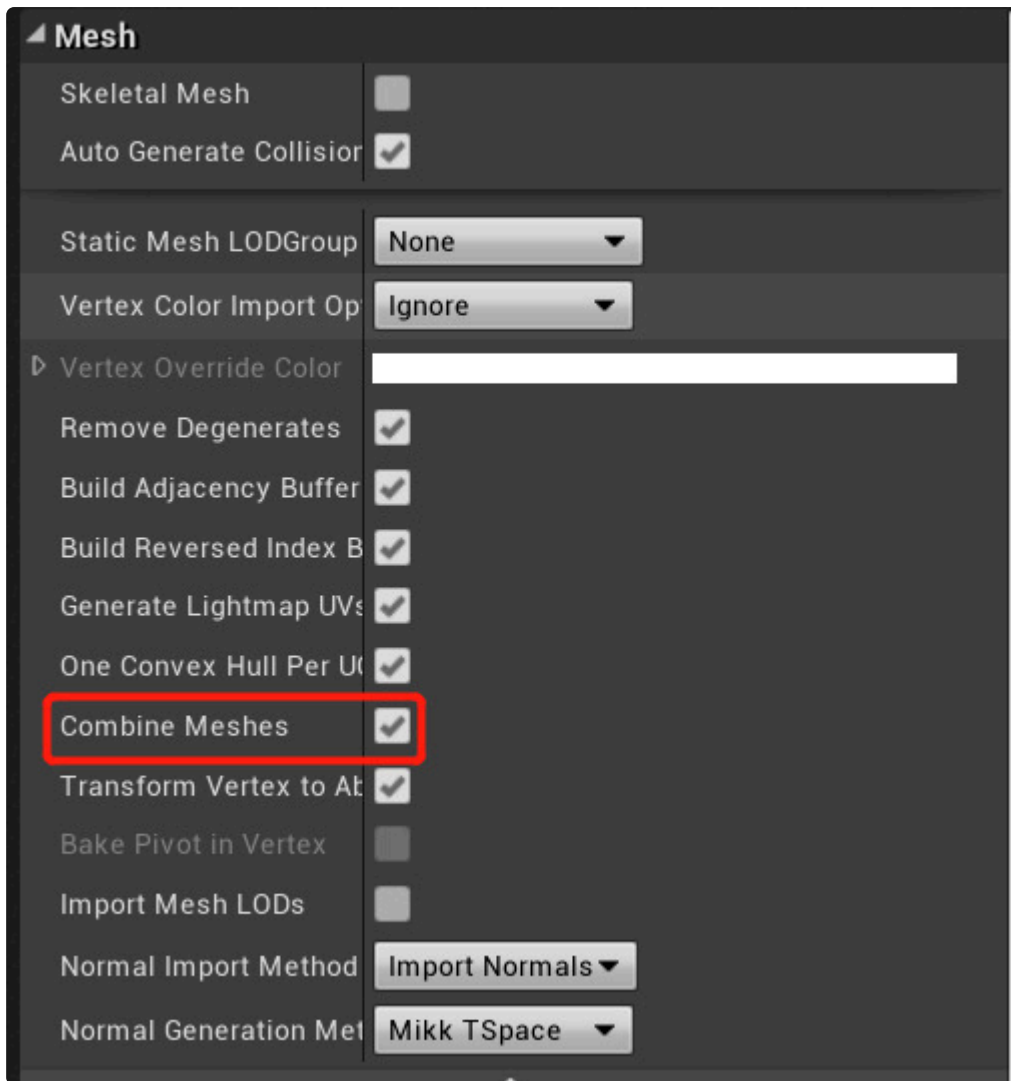


4.2 将该fbx模型导入UE处理并烘培（选做，以UE4为例）

步骤1: fbx模型导入UE

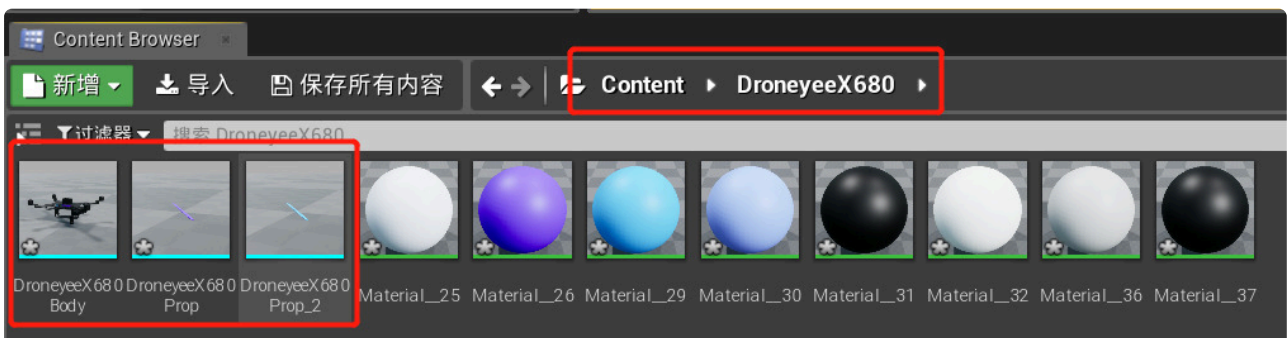
任意打开一个UE工程，在其Content目录新建一个文件夹（这里命名为DroneyeeX680），如下图进入该文件夹后，点击导入按钮，依次将DroneyeeX680Body.FBX和DroneyeeX680Prop.FBX导入进来





步骤2: 处理模型材质

由于我们的fbx模型没有定义材质，因此导入进来是白色的，并不美观。需要双击 DroneeyeX680Body网格体资产，选择资产详细查看材质面板，在其中对关键部件添加颜色材质，例如碳纤维的黑色（**这里需要引用到当前目录下定义的材质，材质及其引用资产的路径只能相对于当前的内容目录，不能引用到UE引擎自带的材质，否则最终导入 RflySim3D无法正确显示**）。同时，复制一个DroneeyeX680Prop组件命名为 DroneeyeX680Prop_2，将其材质设置为不同的颜色，以区分飞机的靠前的两个螺旋桨和靠后的两个螺旋桨。如果导入其他的其他模型本身拥有材质贴图，则需要保证贴图纹理贴在正确的位置上。



步骤3: 项目打包设置与烘焙

在上图中点击"保存所有内容", 在编辑->项目设置->打包中确保如下设置:



然后在文件->烘焙Windows的内容, 进行烘焙:



烘焙完成后，可以

在 `【该工程文件夹】\Saved\Cooked\WindowsNoEditor\【工程名】\Content\` 下，将整个"DroneyeeX680"目录复制出来备用。

4.3 烘焙完成的模型导入RflySim3D

步骤1: 编写XML文件

导出模型后，我们还需要编写XML文件来帮助RflySim3D识别螺旋桨位置、转动方向、材质等。

```

1  <?xml version="1.0"?>
2  <vehicle>
3      <ClassID>3</ClassID>          <!-- 类型, 3表示四旋翼无人机-->
4      <DisplayOrder>1015</DisplayOrder> <!-- 显示顺序, 越小在切换列表 (c键) 越靠前-->
5      <Name>DroneyeeX680</Name>      <!-- 显示名称-->
6      <Scale>          <!-- 缩放-->
7          <x>1</x>
8          <y>1</y>
9          <z>1</z>
10     </Scale>
11     <AngEulerDeg> <!-- 创建时的初始姿态角-->
12         <roll>0</roll>
13         <pitch>0</pitch>
14         <yaw>0</yaw>
15     </AngEulerDeg>
16     <body>
17         <ModelType>0</ModelType> <!-- 0表示静态网格, 1表示动画, 2-4表示蓝图-->
18         <MeshPath>/Game/DroneyeeX680/DroneyeeX680Body</MeshPath> <!-- mesh路径, 鼠标悬
19 停在资产上会显示资产路径, 资产完整路径=显示的资产路径+资产名称-->
20         <MaterialPath></MaterialPath>
21         <AnimationPath></AnimationPath>
22         <CenterHeightAboveGroundCm>8</CenterHeightAboveGroundCm> <!-- 质心离地面的高
23 度, 从图4下面知道为8-->
24         <NumberHeigthAboveCenterCm>20</NumberHeigthAboveCenterCm> <!-- 标签离地面的高
25 度-->
26     </body>
27     <ActuatorList> <!-- 是mesh类型, 有执行器列表-->
28         <Actuator>
29             <MeshPath>/Game/DroneyeeX680/DroneyeeX680Prop</MeshPath> <!-- 执行器mesh
30 路径, 参考body路径获取-->
31             <MaterialPath></MaterialPath>
32             <RelativePosToBodyCm> <!-- 执行器相对于Body的偏移, 由图4知道
33 [25.4260,24.0760,11.2220],[ -25.5410,-22.8910,11.0220], [25.4260,
34 -25.0560,11.0220], [-25.5410,21.9110,11.0220]-->
35                 <x>25.4260</x>
36                 <y>24.0760</y>
37                 <z>11.2220</z>
38             </RelativePosToBodyCm>
39             <RelativeAngEulerToBodyDeg> <!-- 执行器相对于body的姿态角-->
40                 <roll>0</roll>
41                 <pitch>0</pitch>
42                 <yaw>0</yaw>
43             </RelativeAngEulerToBodyDeg>
44             <RotationAxisVectorToBody>          <!-- 执行器相对于body的旋转轴, 显示是z轴-->
45             >
46                 <x>0</x>
47                 <y>0</y>
48                 <z>1</z>
49             </RotationAxisVectorToBody>
50             <RotationModeSpinOrDefect>0</RotationModeSpinOrDefect>
51         </Actuator>

```

```

53 <Actuator>
54   <MeshPath>/Game/DroneyeeX680/DroneyeeX680Prop_2</MeshPath>
55   <MaterialPath></MaterialPath>
56   <RelativePosToBodyCm>
57     <x>-25.5410</x>
58     <y>-22.8910</y>
59     <z>11.0220</z>
60   </RelativePosToBodyCm>
61   <RelativeAngEulerToBodyDeg>
62     <roll>0</roll>
63     <pitch>0</pitch>
64     <yaw>0</yaw>
65   </RelativeAngEulerToBodyDeg>
66   <RotationAxisVectorToBody>
67     <x>0</x>
68     <y>0</y>
69     <z>1</z>
70   </RotationAxisVectorToBody>
71   <RotationModeSpinOrDefect>0</RotationModeSpinOrDefect>
72 </Actuator>
73 <Actuator>
74   <MeshPath>/Game/DroneyeeX680/DroneyeeX680Prop</MeshPath>
75   <MaterialPath></MaterialPath>
76   <RelativePosToBodyCm>
77     <x>25.5410</x>
78     <y>-25.0560</y>
79     <z>11.0220</z>
80   </RelativePosToBodyCm>
81   <RelativeAngEulerToBodyDeg>
82     <roll>0</roll>
83     <pitch>0</pitch>
84     <yaw>0</yaw>
85   </RelativeAngEulerToBodyDeg>
86   <RotationAxisVectorToBody>
87     <x>0</x>
88     <y>0</y>
89     <z>1</z>
90   </RotationAxisVectorToBody>
91   <RotationModeSpinOrDefect>0</RotationModeSpinOrDefect>
92 </Actuator>
93 <Actuator>
94   <MeshPath>/Game/DroneyeeX680/DroneyeeX680Prop_2</MeshPath>
95   <MaterialPath></MaterialPath>
96   <RelativePosToBodyCm>
97     <x>-25.528</x>
98     <y>21.9110</y>
99     <z>11.0220</z>
100  </RelativePosToBodyCm>
101  <RelativeAngEulerToBodyDeg>
102    <roll>0</roll>
103    <pitch>0</pitch>
104
105

```

```

106         <yaw>0</yaw>
107     </RelativeAngEulerToBodyDeg>
108     <RotationAxisVectorToBody>
109         <x>0</x>
110         <y>0</y>
111         <z>1</z>
112     </RotationAxisVectorToBody>
113     <RotationMode>0</RotationMode>
114 </Actuator>
115 </ActuatorList>
116 <OnboardCameras> <!-- 飞行器上的相机，从其他xml复制保持不变 -->
117     <camera>
118         <name>Chase_Camera</name>
119         <RelativePosToBodyCm>
120             <x>-100</x>
121             <y>0</y>
122             <z>5</z>
123         </RelativePosToBodyCm>
124         <RelativeAngEulerToBodyDeg>
125             <roll>0</roll>
126             <pitch>0</pitch>
127             <yaw>0</yaw>
128         </RelativeAngEulerToBodyDeg>
129     </camera>
130     <camera>
131         <name>Front_Camera</name>
132         <RelativePosToBodyCm>
133             <x>20</x>
134             <y>0</y>
135             <z>0</z>
136         </RelativePosToBodyCm>
137         <RelativeAngEulerToBodyDeg>
138             <roll>0</roll>
139             <pitch>0</pitch>
140             <yaw>0</yaw>
141         </RelativeAngEulerToBodyDeg>
142     </camera>
143     <camera>
144         <name>Back_Camera</name>
145         <RelativePosToBodyCm>
146             <x>-20</x>
147             <y>0</y>
148             <z>0</z>
149         </RelativePosToBodyCm>
150         <RelativeAngEulerToBodyDeg>
151             <roll>0</roll>
152             <pitch>0</pitch>
153             <yaw>180</yaw>
154         </RelativeAngEulerToBodyDeg>
155     </camera>
156 </camera>
157 </OnboardCameras>
158

```

```

159     <name>Right_Camera</name>
160     <RelativePosToBodyCm>
161         <x>0</x>
162         <y>20</y>
163         <z>0</z>
164     </RelativePosToBodyCm>
165     <RelativeAngEulerToBodyDeg>
166         <roll>0</roll>
167         <pitch>0</pitch>
168         <yaw>90</yaw>
169     </RelativeAngEulerToBodyDeg>
170 </camera>
171 <camera>
172     <name>Left_Camera</name>
173     <RelativePosToBodyCm>
174         <x>0</x>
175         <y>-20</y>
176         <z>0</z>
177     </RelativePosToBodyCm>
178     <RelativeAngEulerToBodyDeg>
179         <roll>0</roll>
180         <pitch>0</pitch>
181         <yaw>-90</yaw>
182     </RelativeAngEulerToBodyDeg>
183 </camera>
184 <camera>
185     <name>Down_Camera</name>
186     <RelativePosToBodyCm>
187         <x>0</x>
188         <y>0</y>
189         <z>-20</z>
190     </RelativePosToBodyCm>
191     <RelativeAngEulerToBodyDeg>
192         <roll>0</roll>
193         <pitch>-90</pitch>
194         <yaw>0</yaw>
195     </RelativeAngEulerToBodyDeg>
196 </camera>
197 <camera>
198     <name>Up_Camera</name>
199     <RelativePosToBodyCm>
        <x>0</x>
        <y>0</y>
        <z>20</z>
    </RelativePosToBodyCm>
    <RelativeAngEulerToBodyDeg>
        <roll>0</roll>
        <pitch>90</pitch>
        <yaw>0</yaw>
    </RelativeAngEulerToBodyDeg>
</camera>

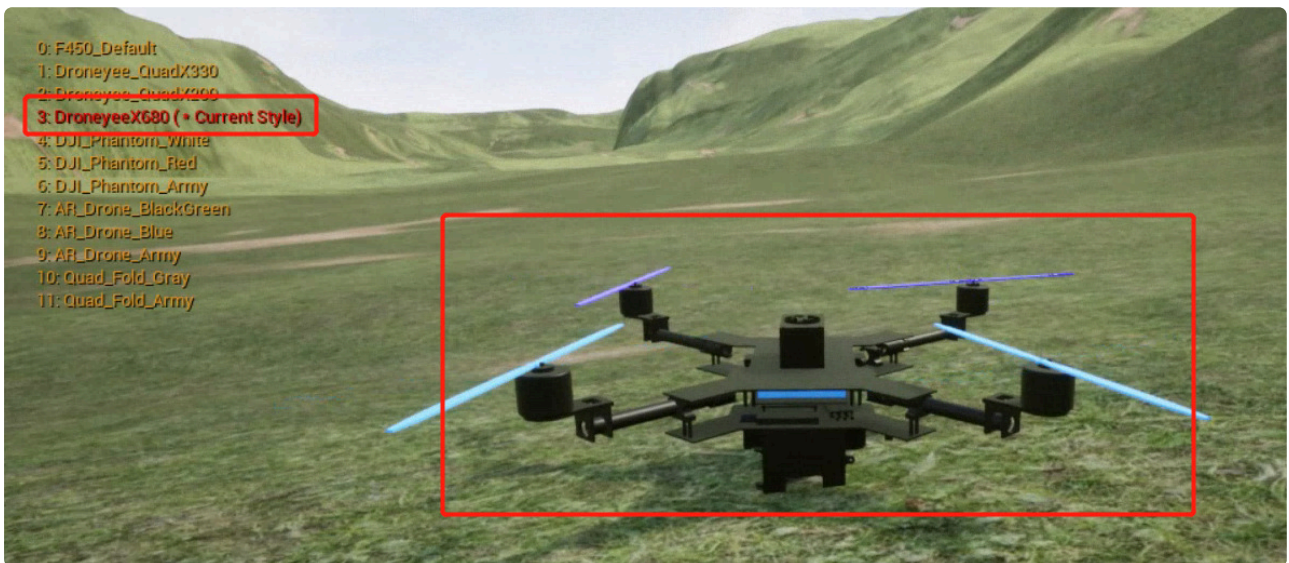
```

```
</OnboardCameras>  
</vehicle>
```

步骤2: 导入RflySim3D

将编写好的XML文件拷贝到之前生成的DroneyeeX680文件夹中，再一起拷贝到 RflySim3D\RflySim3D\Content 目录

打开RflySim3D，鼠标双击地面+O+3创建一架四旋翼，按下C键可以切换飞机样式，看看自己的飞机模型是否已经导入。



注：XML中DisplayOrder数字小于1000就会超过内置模型排到最前边，变成默认显示飞机。详细教程如下：

RflySim:如何将你的飞机三维模型导入到基于UE的RflySim3D程序中并用于视景仿真，本视频观看地址：

优酷：https://v.youku.com/v_show/id_XNDcwNjA4NzIxMg==.html

YouTube：<https://youtu.be/mKUehJwqqSU>

B站：<https://www.bilibili.com/video/BV13a411i7sH?p=9>

步骤3: 最终效果

此文件目录下有一个已经完成的例程，找到DroneyeeX680文件夹，将文件夹复制到PX4PSP\RflySim3D\RflySim3D\Content目录下。最后打开RflySim3D，鼠标双击地面+O+3，就能创建出这个ClassID为3的飞机，然后按C键切换到刚刚导入的DroneyeeX680飞机即可。



5. 关键知识点

关键知识点1：XML配置文件格式

有关XML配置文件的格式，请参考

[\[RflySim安装目录\]/RflySimAPIs/3.RflySim3DUE/API.pdf](#)中的8.1节

关键知识点2：支持的三维建模软件和文件格式

本实验不仅支持3ds Max，还支持多种主流的三维建模和CAD软件进行自定义载具开发：

主要支持的三维建模软件

- **Autodesk 3ds Max (2016-2022)**: 主要演示软件，支持直接导出FBX
- **Autodesk Maya**: 通过FBX格式支持，业界标准建模软件
- **Blender**: 开源免费建模软件，可导出FBX格式
- **Maxon Cinema 4D**: 专业建模渲染软件，通过Datasmith支持
- **McNeel Rhinoceros**: 工业设计软件，支持导出插件
- **Trimble SketchUp Pro**: 建筑设计软件，支持导出插件

主要支持的CAD软件

- **SOLIDWORKS**: 工业设计标准软件，通过Datasmith导出插件

- **Autodesk Inventor**: 机械设计软件，直接支持CAD格式
- **CATIA V5**: 航空航天工业软件，直接支持CAD格式
- **PTC Creo (Pro/ENGINEER)**: 机械设计软件，直接支持CAD格式
- **Siemens NX**: 高端CAD软件，直接支持CAD格式

Unreal Engine支持的三维模型格式

原生支持格式（无需插件）：

- **FBX**: 主要推荐格式，支持网格、动画、材质、纹理的完整导入
- **OBJ**: 基础网格格式，仅支持几何体，需要单独处理材质

通过Datasmith支持的格式（需要插件）：

- **STEP** (*.step , *.stp): 国际标准CAD交换格式，支持AP203、AP214、AP242
- **IGES** (*.iges , *.igs): 早期CAD交换格式，支持5.1、5.2、5.3版本
- **3DXML**: Dassault Systèmes的3D交换格式
- **JT Open**: Siemens PLM的轻量化格式
- **Parasolid** (*.x_t): 几何建模内核格式
- **3DS**: 早期3ds Max格式（已较少使用）
- **Alembic** (*.abc): 动画几何体缓存格式
- **glTF**: 现代Web 3D标准格式（实验性支持）

关键知识点3：坐标系转换差异

不同建模软件和三维格式使用不同的坐标系统，在导入UE时需要注意转换：

坐标系对照表

软件/格式	坐标系类型	前进方向	向上方向	向右方向	备注
3ds Max	右手坐标系	+X	+Z	+Y	本例程使用
Maya	右手坐标系	+Z	+Y	+X	需要转换
Blender	右手坐标系	+Y	+Z	+X	需要转换

软件/格式	坐标系类型	前进方向	向上方向	向右方向	备注
Cinema 4D	左手坐标系	+Z	+Y	+X	需要转换
Unreal Engine	左手坐标系	+X	+Z	+Y	目标格式

转换注意事项

1. **从3ds Max到UE**：需要对Y轴进行反向（本例程已处理）
2. **从Maya到UE**：需要旋转90度并调整轴向
3. **从Blender到UE**：导出时选择正确的轴向设置
4. **模型朝向**：确保机头始终指向前进方向（UE中的+X方向）
5. **缩放单位**：注意不同软件的单位制（厘米、米、英寸）转换

关键知识点4：其他三维模型导入方式参考教程

官方教程链接

- **UE官方FBX导入教程**：
<https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Importing/FBX/>
- **UE官方Datasmith教程**：
<https://docs.unrealengine.com/5.0/en-US/datasmith-supported-software-and-file-types/>
- **Blender到UE工作流程**：
<https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Importing/>
- **Maya到UE工作流程**：
<https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Importing/FBX/FBXBestPractices/>

第三方参考教程

- **Blender建模导入UE教程**：B站搜索"Blender UE4 导入教程"
- **SOLIDWORKS到UE工作流程**：YouTube搜索"SOLIDWORKS to Unreal Engine"
- **通用CAD软件导入教程**：各软件官方网站的UE集成文档

坐标轴处理参考

- **3ds Max坐标轴设置**：File > Export > FBX Export Options > Axis Conversion

- **Maya坐标轴设置**: FBX Export > Advanced Options > Axis Conversion
- **Blender坐标轴设置**: Export > FBX > Transform > Forward/Up Axis设置

6. 参考资料

6.1 RflySim3D相关文档

1. XML文件规则 [../API.pdf](#)
2. RflySim3D快捷键接口总览 [../API.pdf](#)
3. RflySim3D模型导入总览 [../API.pdf](#)

6.2 三维建模软件官方文档

建模软件

- 3ds Max官方文档: <https://help.autodesk.com/view/3DSMAX/2023/CHS/>
- Maya官方文档: <https://help.autodesk.com/view/MAYAUL/2023/CHS/>
- Blender官方文档: <https://docs.blender.org/manual/zh-hans/latest/>
- Cinema 4D官方文档: <https://help.maxon.net/c4d/>

CAD软件

- SOLIDWORKS用户指南: <https://help.solidworks.com/>
- Autodesk Inventor帮助: <https://help.autodesk.com/view/INVNTOR/2023/CHS/>
- CATIA用户文档: <https://help.3ds.com/catia/>
- PTC Creo帮助中心: <https://support.ptc.com/help/creo/>

6.3 Unreal Engine相关资源

- UE官方中文文档: <https://docs.unrealengine.com/4.27/zh-CN/>
- FBX导入最佳实践:
<https://docs.unrealengine.com/4.27/en-US/WorkingWithContent/Importing/FBX/FBXBestPractices/>

- **Datasmith支持的软件列表:**

<https://docs.unrealengine.com/5.0/en-US/datasmith-supported-software-and-file-types/>

- **UE坐标系统说明:**

<https://docs.unrealengine.com/4.27/en-US/Basics/Coordinates/>

6.4 格式转换工具

- **Autodesk FBX Converter:**

<https://www.autodesk.com/developer-network/platform-technologies/fbx-converter-archives>

- **Open3D:** 开源三维数据处理库，支持多种格式转换
- **MeshLab:** 开源网格处理软件，支持多种格式

7. 常见问题

Q1: 使用其他三维建模软件（如Blender、Maya）时，导出的FBX模型在UE中显示方向不正确怎么办？

A1: 这是由于不同软件使用不同的坐标系统造成的。解决方法：

- **Blender:** 导出FBX时，将Forward设置为"-Y Forward"，Up设置为"Z Up"
- **Maya:** 导出FBX时，在Advanced Options中将Axis Conversion设置为正确的坐标系
- **通用解决方案:** 在UE导入时，可以在Import Options中调整Transform设置

Q2: 使用CAD软件（如SOLIDWORKS）导出的模型文件很大，导入UE后性能很差怎么办？

A2: CAD模型通常精度很高，面数较多。建议：

1. 在CAD软件中简化模型，去除不必要的细节
2. 使用抽取工具降低面数
3. 考虑制作LOD（细节层次）模型
4. 将复杂模型分解为多个简单部件

Q3: 模型导入UE后材质丢失或显示为白色, 怎么处理?

A3: 不同软件材质系统差异导致。解决方法:

1. **FBX导出时**: 确保勾选"Embed Media"选项, 将贴图嵌入FBX文件
2. **手动重新指定材质**: 在UE中为模型重新创建和指定材质
3. **使用Datasmith**: 对于复杂材质, 建议使用Datasmith插件导入

1. <https://rflysim.com/> ↩

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