

1、实验名称及目的

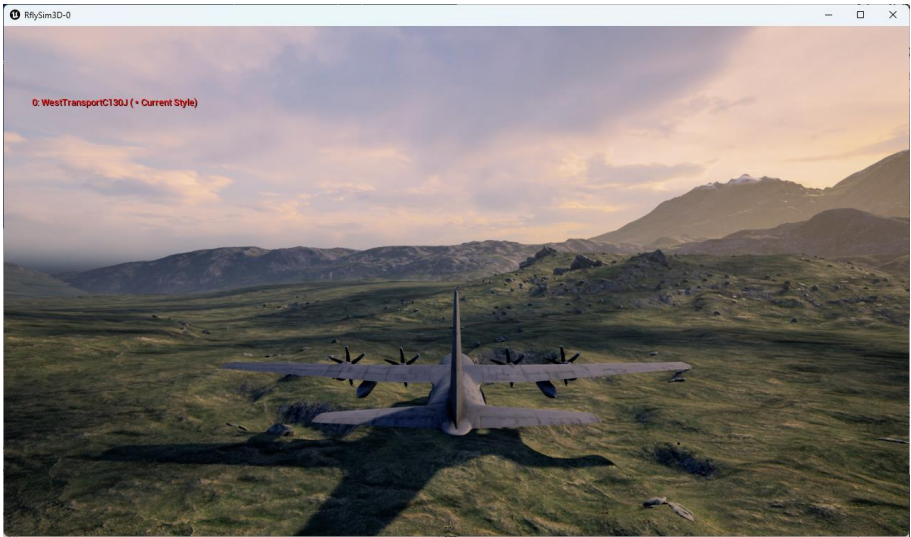
虚幻商城固定翼蓝图模型导入：熟悉将固定翼蓝图模型导入到 RflySim3D, 并能确保其正常工作的流程

2、实验原理

将蓝图模型导入平台前，需要修改它的蓝图事件与平台对应的接口相匹配，还需要编写 XML 文件来帮助 RflySim3D 识别执行器位置、舵面转动方向、材质等。RflySim3D 提供给蓝图的接口主要包括：“ActuatorInputs”与“ActuatorInputsExt”，这两个接口可以将一组数据传输给场景中指定的无人机，可以用于展示一些自定义的效果（该效果需要使用蓝图系统自己编写），例如打开舱门、爆炸、显示文字、切换飞机材质、使机翼旋转等等任何 UE 蓝图系统能做到的事情。它是更高阶的接口，因此也要求基本掌握 UE 的蓝图使用方法。如果想使用蓝图接口来控制场景中的 Copter，需要保证它是以“蓝图形式”导入的 RflySim3D，故使用该接口要求无人机的 XML 文件中的“isAnimationMesh”标记值为 2，即无人机机体 body 使用的不是 mesh 网格，而是蓝图模型。

在导入后，需要进行全面的测试以确保所有执行器都能正确工作，并且飞机的模拟表现与预期一致。可以编写 Simulink、Python 脚本或直接通过 RflySim3D 控制台输入命令来验证各执行器动作是否正确。

3、实验效果



4、文件目录

文件夹/文件名称	说明

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 平台完整版		
3	EpicLauncher		
4	UnrealEngine4.27		

推荐配置请见：<https://doc.rflysim.com>

6、实验步骤

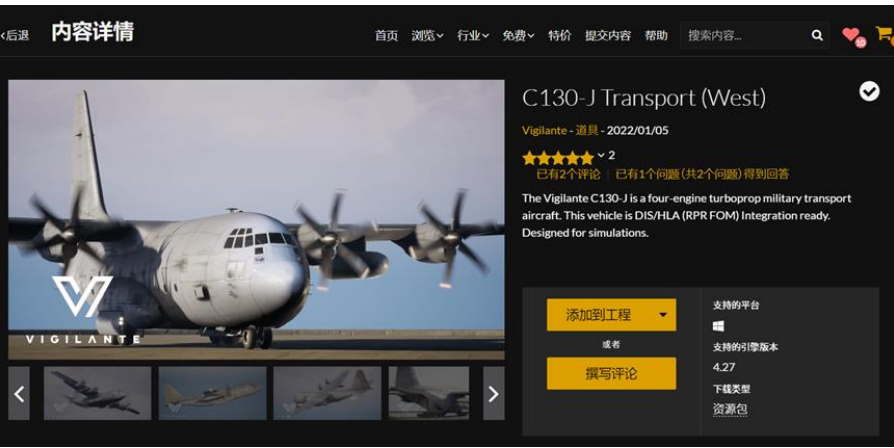
Step 1: 资源添加

UE4

在虚幻商城搜索“Vigilante”。



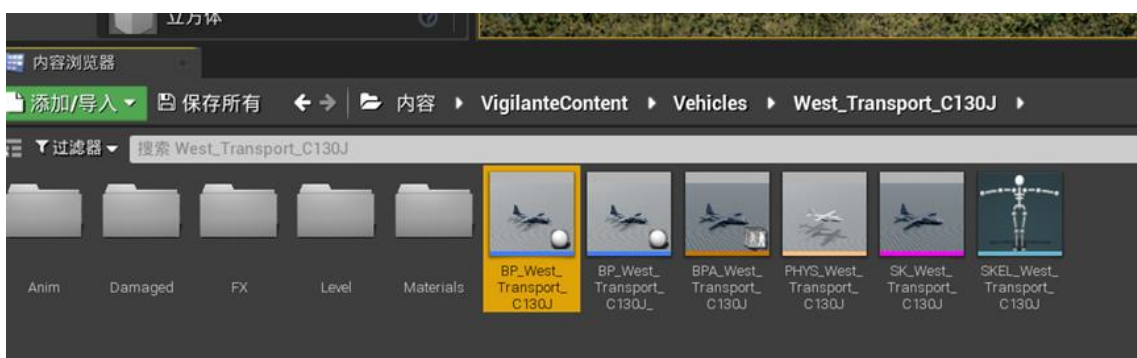
以 C130-J Transport(West) 为例。



使用 4.27 版本的虚幻引擎，创建一个空白工程，将上述的 C130-J 添加到创建好的工程中。（创建工程时打开光线追踪）



在内容浏览器中可以看到 BP_West_Transport_C130J 的蓝图类。



因为只需要一个蓝图类，删除 BP_West_Transport_C130J_Showcase，同时删除多余场景文件 level，另外重命名最外层的 VigilanteContent 文件夹为 WestTransportC130J（便于后续的访问和分享），最后保留如下文件：

Anim：动画序列

Damaged：爆炸毁伤后的网格体和材质

FX：粒子特效

Materials：模型材质

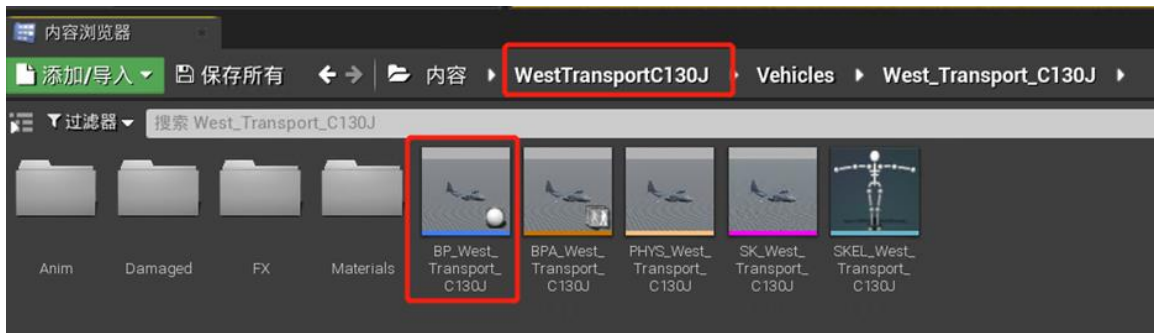
BP_West_Transport_C130J：飞机蓝图类

BPA_West_Transport_C130J：飞机动画蓝图，包含事件图表和动画图表

PHYS_West_Transport_C130J：飞机物理资产，主要是碰撞模型，定义了飞机的碰撞边界。

SK_West_Transport_C130J：飞机骨骼网格体（Skeletal Mesh），一个具有几何体和关联骨骼的 3D 模型，用于呈现具有动画需求的模型。

SKEL_West_Transport_C130J：骨架（Skeleton），描述骨骼层次结构和关节运动的数据结构，用于驱动骨骼网格的动画。

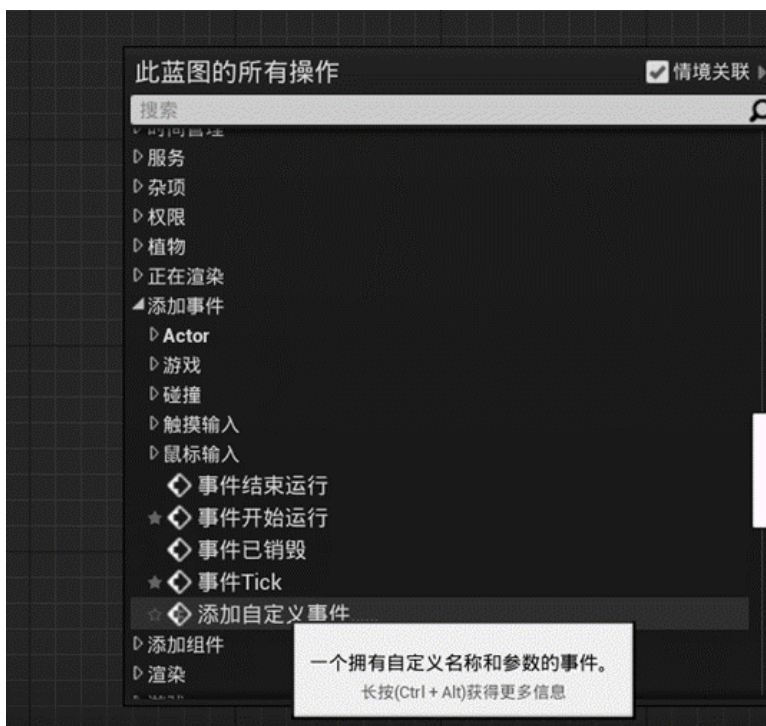


UE5

Step 2: 修改蓝图事件

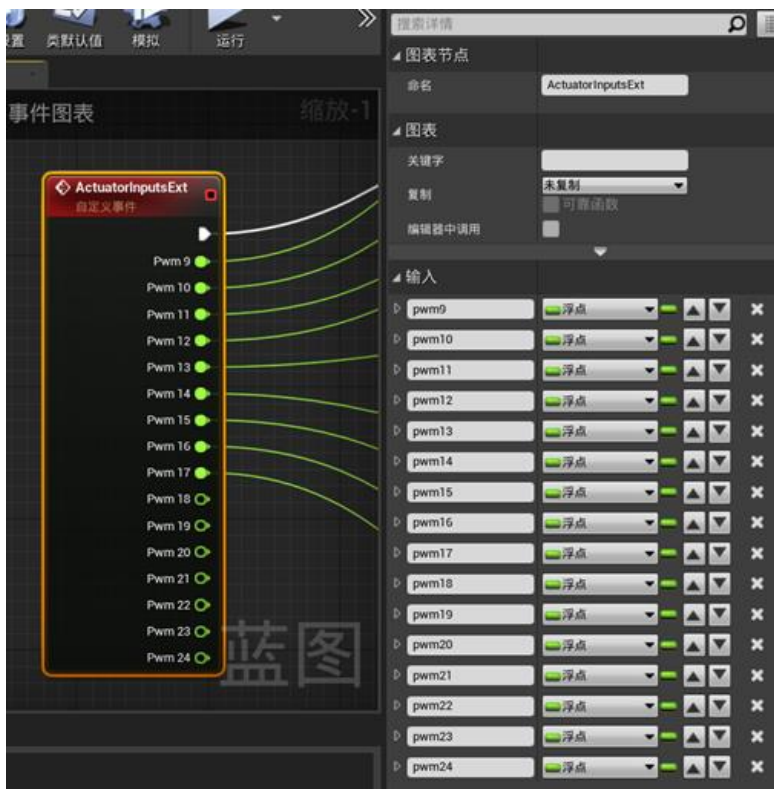
UE4

接下来进入它的蓝图图表界面，右键添加 CustomEvent 节点，为其创建一个自定义事件“ActuatorInputs”（必须是这个名字），并且给它添加 8 个 float 的参数，作为基础 8 维的执行器输入信号。





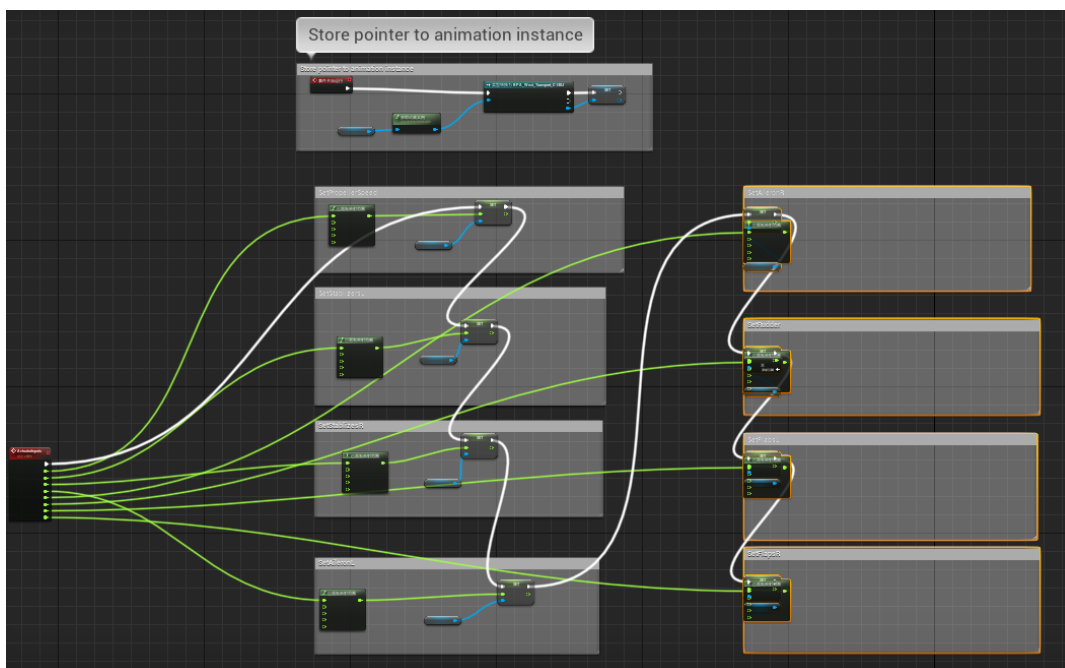
再创建一个名为“ActuatorInputsExt”的自定义事件，添加 pwm9~pwm24 的 16 维输入信号，作为 9~24 维扩展执行器输入信号接收端。



注意：创建的两个自定义事件的名称必须为“ActuatorInputs”和“ActuatorInputsExt”，维度必须为 8 维和 16 维，如果没有这两个自定义事件，飞机无法触发执行器的动作或动画。

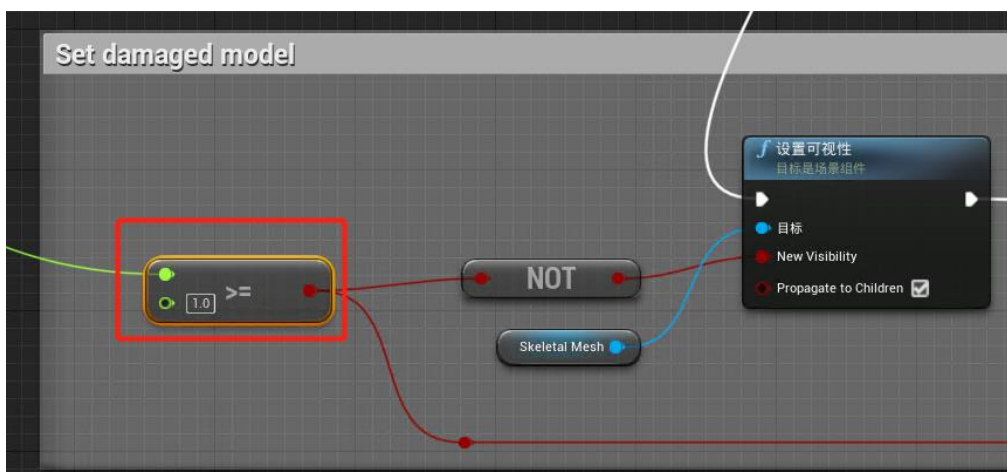
将 ActuatorInputs 的 pwm 输出端依次连接到各个舵面控制输入上，8 维信号连接顺序如下图（白色的线确定了代码执行顺序），（推荐顺序：SetPropellerSpeed、SetStabilizersL、SetStabilizersR、SetAileronL、SetAileronR、SetRudder、SetFlapsL、SetFlapsR）

可参考[机架参考 | PX4 自动驾驶用户指南](#)进行连接



ActuatorInputsExt 的输出口可以连接其他的执行器。为了方便和我们的例程对应，这里的推荐顺序为：RetractFrontWheel、RetractRearLWheel、RetractRearRWheel、OpenRamp、OpenRearLDoor、OpenRearRDoor、OpenFrontDoor、Set damaged model（毁伤效果接到 24 位）。

注意：毁伤模型里面，要通过浮点数来触发毁伤，需要增加浮点数判断（右键，搜索“大于”，找到“浮点数>=浮点数”）模块，设定大于等于 1 才触发毁伤。其他数据默认即可。



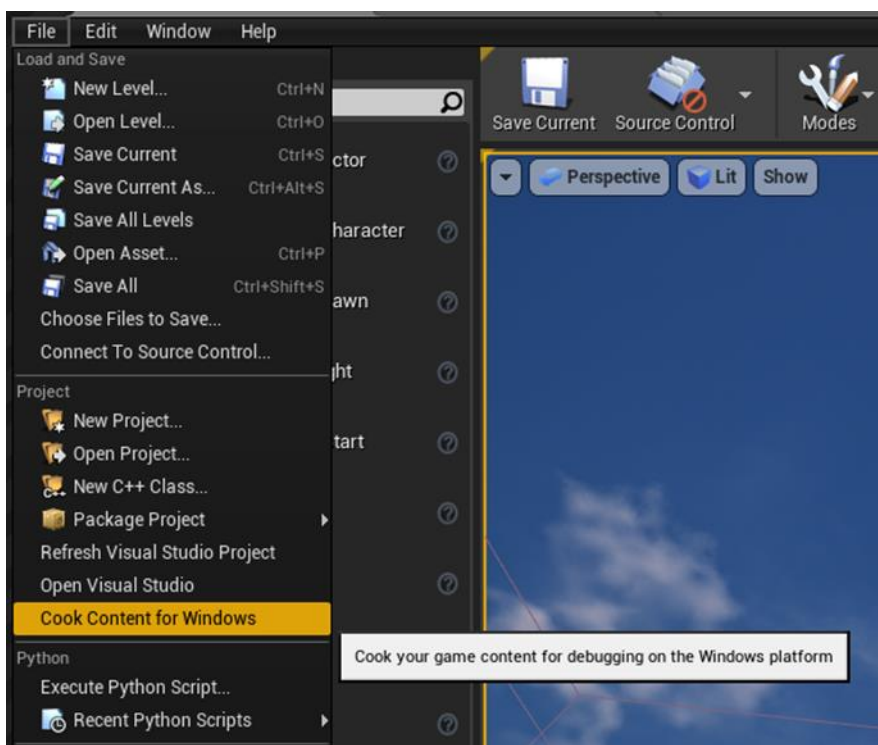
在细节面板启用碰撞事件，其碰撞边界是在模型物理资产中定义好的。



点击“保存”和“编译”，使蓝图生效，然后关闭蓝图窗口，



烘焙文件，点击 File->Cook Content for Windows



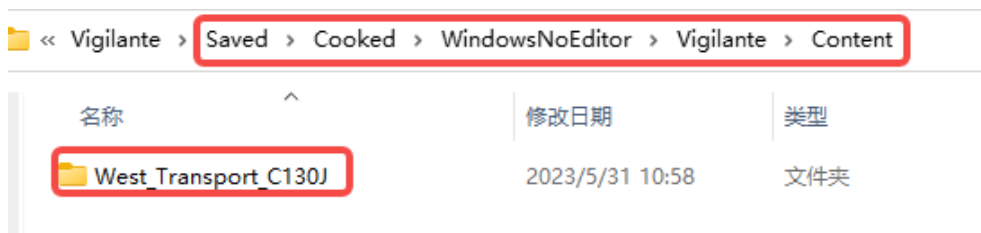
UE5

Step 3: 导入 RflySim3D

UE4

接下来需要进行两个操作将它导入 RflySim3D，使我们可以在 RflySim3D 中创建它：

首先是在【项目文件夹】-> Saved -> Cooked -> WindowsNoEditor -> 【项目名】 -> Content 目录中，可以拷贝出刚才烘焙的“WestTransportC130J”文件夹备用。



第二步是编写 XML 文件，通常可以拿其他无人机的 XML 文件（例如 F450_Default.xml）稍作修改即可。

主要修改的地方如下（更完整的 XML 文件规则见参考资料[1]）：

```
<ClassID>206</ClassID>
<DisplayOrder>1000</DisplayOrder>
<Name>WestTransportC130J</Name>  这里是飞机的名字
<Scale>  这里设置飞机的缩放尺寸
  <x>1</x>
```



```

        <y>1</y>
        <z>1</z>
    </Scale>

```

```

<body>
    <isAnimationMesh>2</isAnimationMesh> 取值 (0: StaticMesh 1: Animation 2: Blueprints
3: Vehicle)
    <MeshPath>/Game/WestTransportC130J/Vehicles/West_Transport_C130J/BP_West_Transport
_C130J </MeshPath> 飞机蓝图路径
    <MaterialPath></MaterialPath>
    <AnimationPath></AnimationPath>
    <CenterHeightAboveGroundCm>237</CenterHeightAboveGroundCm>
    <isMoveBodyCenterAxisCm>0,0,-237</isMoveBodyCenterAxisCm>
    <NumberHeightAboveCenterCm>200</NumberHeightAboveCenterCm>
    <NumberSizeScale>10</NumberSizeScale>
</body>

```

因为导入蓝图默认坐标中心在地面，导致飞机进行欧拉角旋转时，以地面中心为轴而不是以机体为中心。isMoveBodyCenterAxisCm 可以将飞机进行实体进行平移 0,0,-237（z轴向上为正）相当于飞机网格整体向下移动 237cm，使机体中心挪到地面中心，两者轴心合一。CenterHeightAboveGroundCm 是告诉 RflySim3D 飞机中心距底端的距离，确保发送飞机位置 0 0 0 时，能底端触地

```

<RelativePosToBodyCm>
    <x>-4000</x>
    <y>0</y>
    <z>1000</z>
</RelativePosToBodyCm>

```

这里设定了跟随视角的位置，z 轴向上为正，单位 cm。

完成修改后，将该 xml 文件放入之前烘焙好的“WestTransportC130J”文件夹中。一并拷贝到 C:\PX4PSP\RflySim3D\RflySim3D\Content 下

UE5

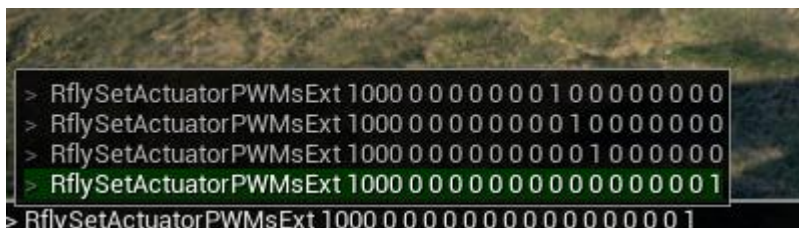
Step 4: 测试

打开 RflySim3D，鼠标双击地面+字母 O+数字 206，创建一个固定翼无人机，然后按 C 键切换到刚刚烘焙的“WestTransportC130J”飞机。

按下键盘“V”键，调整到合适视野

按下键盘左上角“`”的按键，进入命令行

输入 RflySetActuatorPWMsExt 1000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1（可以直接复制）



键入上述命令后回车可以看到飞机爆炸场景



上述指令向 1000 号 ID 的飞机（双击+O 按键创建物体的 ID 默认 1000 开头），发送到第 9 到 24 位执行器数据，这里我们将第 24 位赋值为 1（对应蓝图爆炸场景），可以触发爆炸动画。也可以通过 `RflySetActuatorPWMs 1000 pwm1 ... pwm8` 来控制前 8 维执行器

注意：

`ActuatorInputs` 事件并不是只能通过“`RflySetActuatorPWMs`”命令触发，事实上只要接收到无人机的数据，该事件就会被触发（无人机是蓝图类的实例、并且该事件存在），有可能使用了“`RflySetActuatorPWMs`”命令设置了无人机旋翼的速度后，该值立刻又被飞机的 UDP 数据给覆盖掉了。

免费版只能接收前 8 维执行器数据，无法接收 9 到 24 维数据，因此无法直接观察到效果。免费版用户，可以在 UE4 中将爆炸的端口移到 `ActuatorInputs` 的第 8 维，然后修改 `Simulink`，`Python` 和 `UE4` 命令，即可观察效果。

7、参考资料

- [1]. RflySim3D 模型导入总览 [（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

Q1: ****

A1: ****