

## 1、实验名称及目的

**RflySim3D 蓝图接口实验：**在进行仿真前，保证 Copter 以“蓝图形式”导入 RflySim3D/RflySimUE5，仿真时通过调用蓝图接口来控制场景中的 Copter。

关于导入自定义的 Copter 的详细流程，可以参考 [RflySim3D 自定义无人机模型加载实验\[1\]](#)，这里只是介绍蓝图控制接口。

## 2、实验原理

三维场景中显示的一个个物体都是一个个的对象，它们也有一个个成员变量、成员函数，我们控制的这些无人机也是这样，这些对象是在 UnrealEngine 中定义的，它们如果是继承于 UE 蓝图类的对象，则平台提供了调用蓝图函数的接口。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。

所谓蓝图接口，就是从外部调用蓝图的函数，从而触发在蓝图中编写的各种功能。这种调用可以通过之前介绍过的两个 RflySim3D 控制台命令：“RflySetActuatorPWMs”与“RflySetActuatorPWMsExt”来实现，也可使用其他程序（如 Python、Simulink）通过 UDP 发送指定的结构体来实现。

## 3、实验效果

本实验利用 RflySim3D 蓝图接口控制了仿真过程中旋翼的转动。



[效果图](#)

## 4、文件目录

文件夹/文件名称	说明
Droneyee_WithBP_cooked_UE4	其内的 Droneyee_WithBP 文件夹包含烘焙完成的 copter（含 XML，可直接导入 RflySim3D）
Droneyee_WithBP_content_UE4	在 UE4.27.2 中编辑好的蓝图模型，可导入 UE4.27.2 及以上的 project 中打开
Droneyee_WithBP_cooked_UE5	其内的 Droneyee_WithBP 文件夹包含烘焙完成的 copter（含 XML，可直接导入 RflySimUE5）
Droneyee_WithBP_content_UE5	在 UE5.1.1 中编辑好的蓝图模型，可导入 UE5.1.1 及以上的 project 中打开
DroneyeeX680Body.FBX	机体模型
DroneyeeX680Prop.FBX	旋翼模型

## 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 <sup>①</sup>	1
2	RflySim 平台完整版		
3	UE4.27.2		
4	UE5.1.1		

推荐配置请见：<https://doc.rflysim.com>

## 6、实验步骤

### Step 1: 在 UE 中创建蓝图类

#### UE4

我们打开 UE4，创建一个空项目，具体步骤可参考[../1.BasicExps/e0\\_StarterContent/Readme.pdf](https://doc.rflysim.com/1.BasicExps/e0_StarterContent/Readme.pdf)

在它的 Content 下创建一个文件夹“Droneyee\_WithBP”，然后将此文档下的“DroneyeeX680Body.FBX”、“DroneyeeX680Prop.FBX”导入 UE4 中，然后创建一个蓝图 Actor，并将无人机的机身与旋翼组装好，用不同颜色的螺旋桨来标记机头朝向：

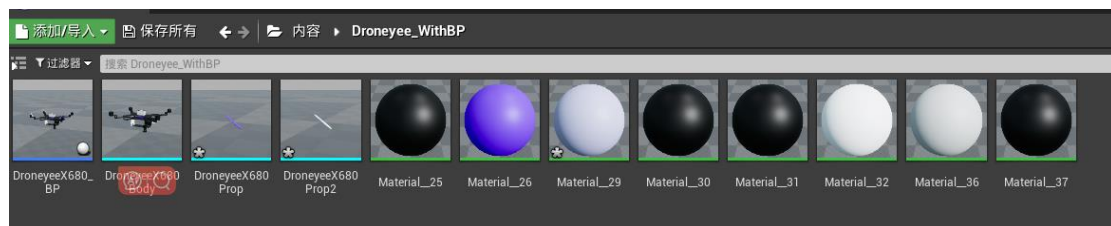


图 1



图 2

UE5

与 UE4 配置相同



图 3

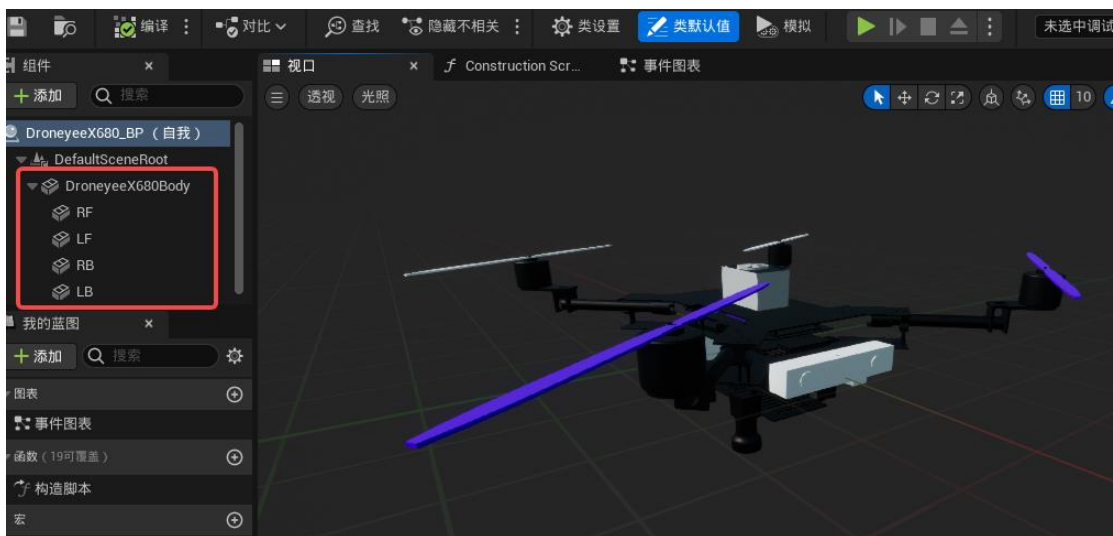


图 4

## Step 2: 定义接口函数并编写功能

### UE4

接下来进入它的蓝图图表界面，右键 **AddCustomEvent** 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。那么当该接口被触发时，就会触发该函数。这里我们接收它的前 4 个值，分别存储起来，作为 4 个旋翼的旋转速度。

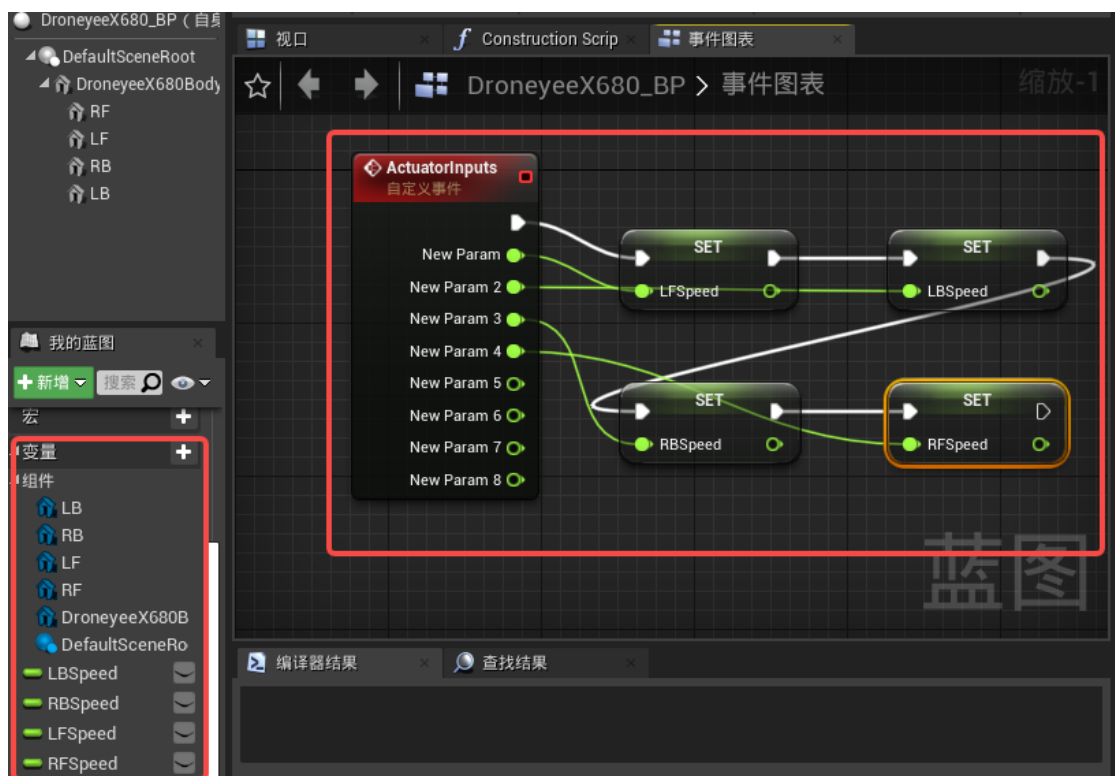


图 5

然后添加 **EventTick** 节点，在 **EventTick** 中对这些旋翼实施旋转。**EventTick** 是场景中每帧都会被调用的事件，如果我们想要旋翼看起来在旋转运动，我们需要每帧都修改它的姿态，当多帧连续起来时它看起来就是在旋转了。

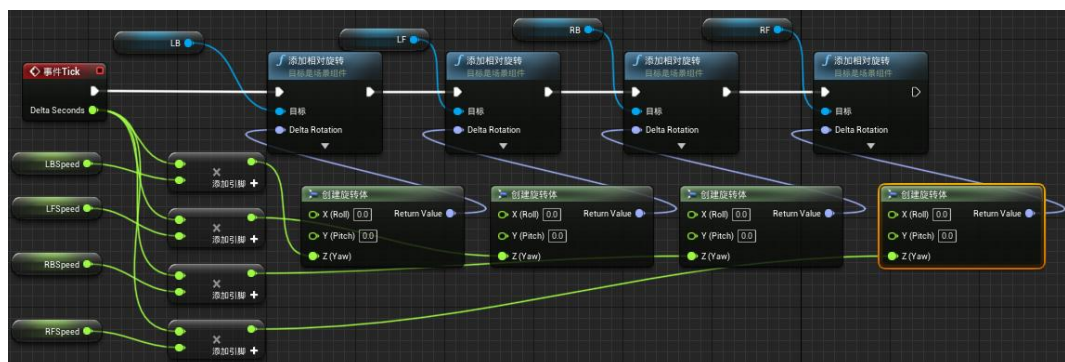
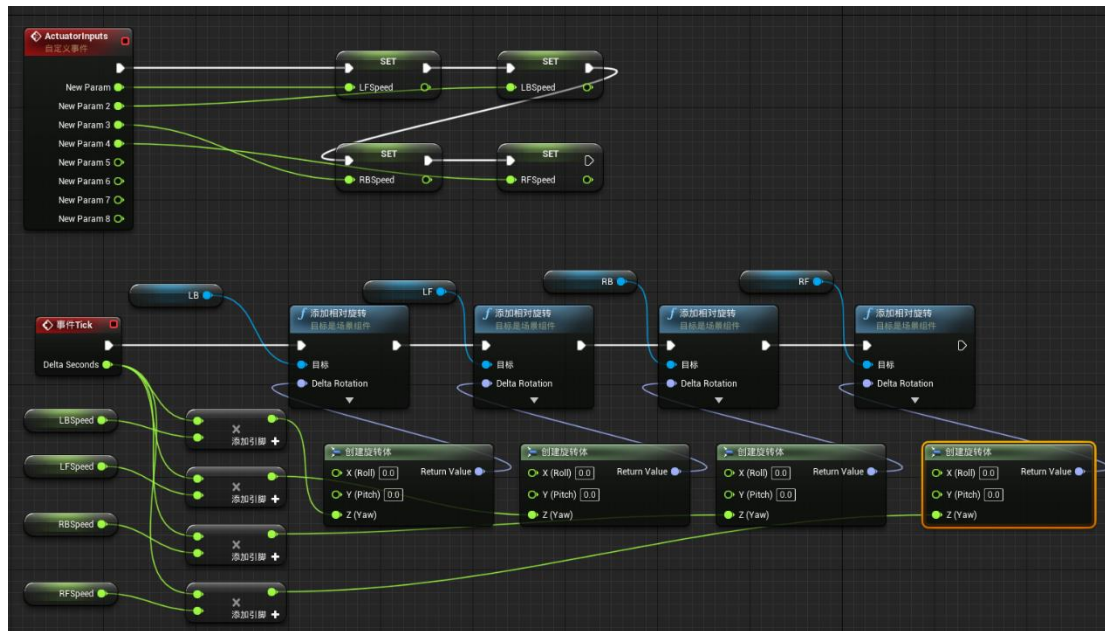


图 6

最后完成如下图的蓝图时间图表即可



我们可以再创建一个自定义事件 **“ActuatorInputsExt”**，它是第二个蓝图接口，但是想要触发它需要完整版的 **RflySim**，这里我们将接收到的前 3 维数据作为四个旋翼在 **xyz** 方向的位置偏移。

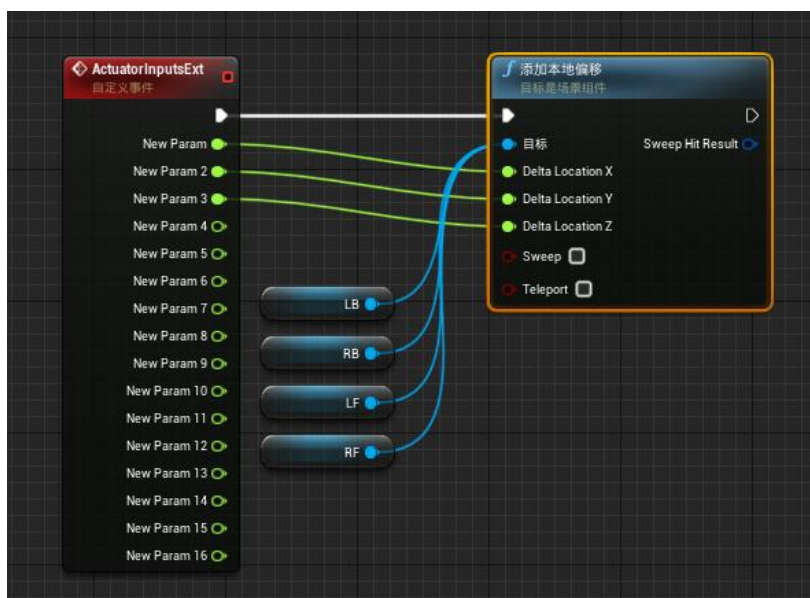


图 8



## UE5

配置步骤与 UE4 相同

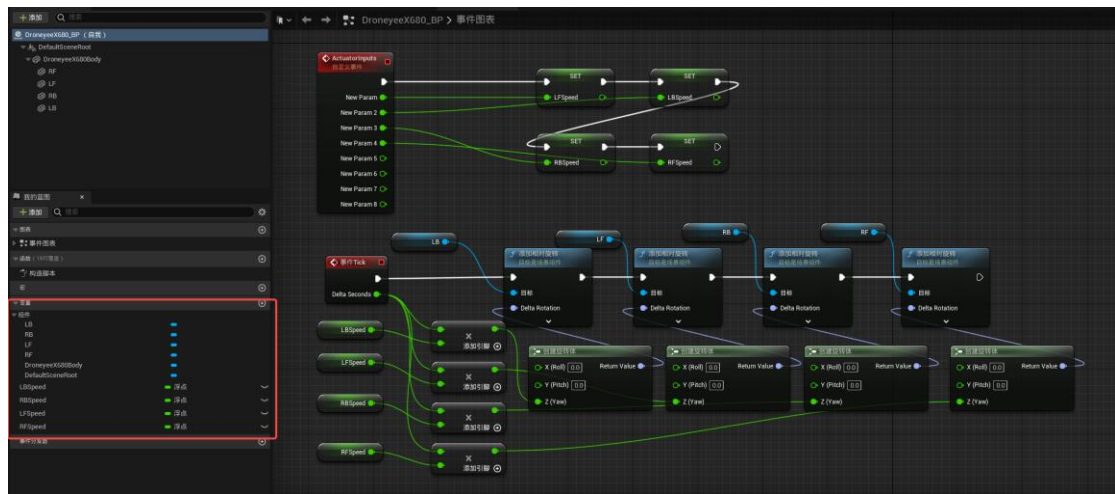


图 9

## Step 3: 导入 RflySim3D/RflySimUE5

### UE4

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

首先是烘培，点击 File->Cook Content for Windows，等待烘培完成后，去项目目录下找到 “\【项目名】\Saved\Cooked\WindowsNoEditor\【项目名】\Content\DroneyeX680\_Wit hBP” 文件夹，将它拷贝出来备用即可。

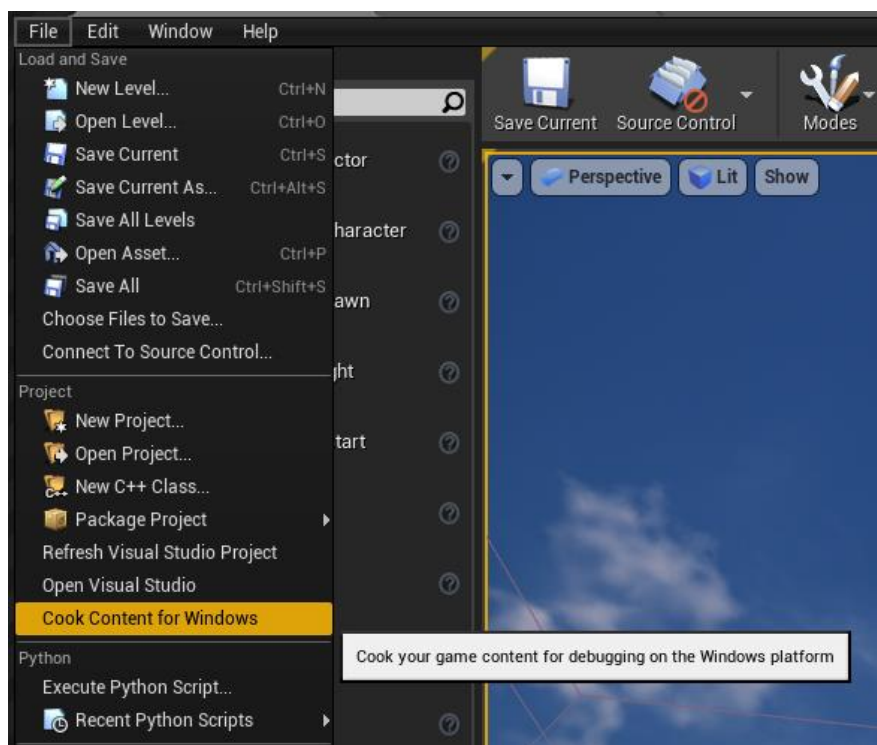


图 10

第二步是编写 XML 文件，通常可以拿其他无人机的 XML 文件稍作修改即可，此文档目录下已经写好一个“DroneyeeX680\_WithBP.xml”文件了，它就是通过“\PX4PSP\RflySim3D\RflySim3D\Plugins\Rfly3DSimPlugin\Content\XML\F450\_Default.xml”修改而来的。

修改的地方主要有：

<isAnimationMesh>标签的值修改为 2 了，表示声明该飞机是一个蓝图飞机。

<body>标签中的<MeshPath>的路径修改了，表示这个蓝图的路径。

删除了执行器标签<ActuatorList>，因为蓝图模型不需要依靠 XML 文件来设置执行器。

将该 xml 文件放入之前烘焙好的“DroneyeeX680\_WithBP”文件夹中。

将这个“DroneyeeX680\_WithBP”文件夹拷贝到“\PX4PSP\RflySim3D\RflySim3D\Content”路径下。

## UE5

烘焙

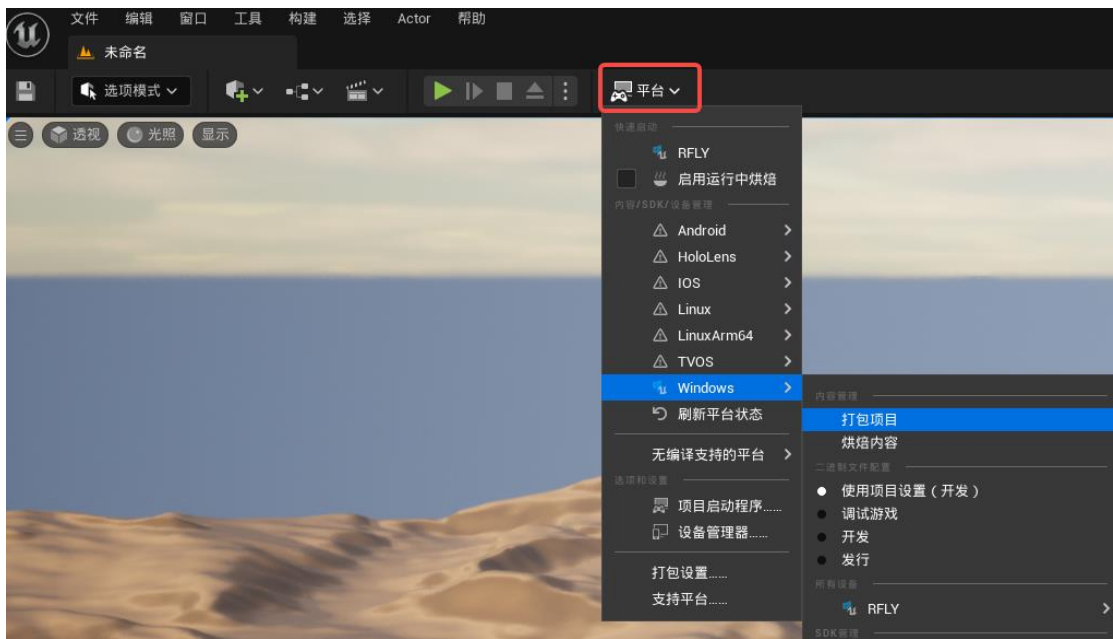


图 11

XML 文件编写与 UE4 相同

将这个“DroneyeeX680\_WithBP”文件夹拷贝到“\PX4PSP\RflySimUE5\RflySim3D\Content”路径下。

## Step 4: 调用蓝图接口[2][3]

### UE4

打开 RflySim3D，鼠标双击地面+字母 O+数字 3，创建一个四旋翼无人机，然后按 C 键切换到刚刚烘培的“DroneyeeX680\_WithBP”无人机。



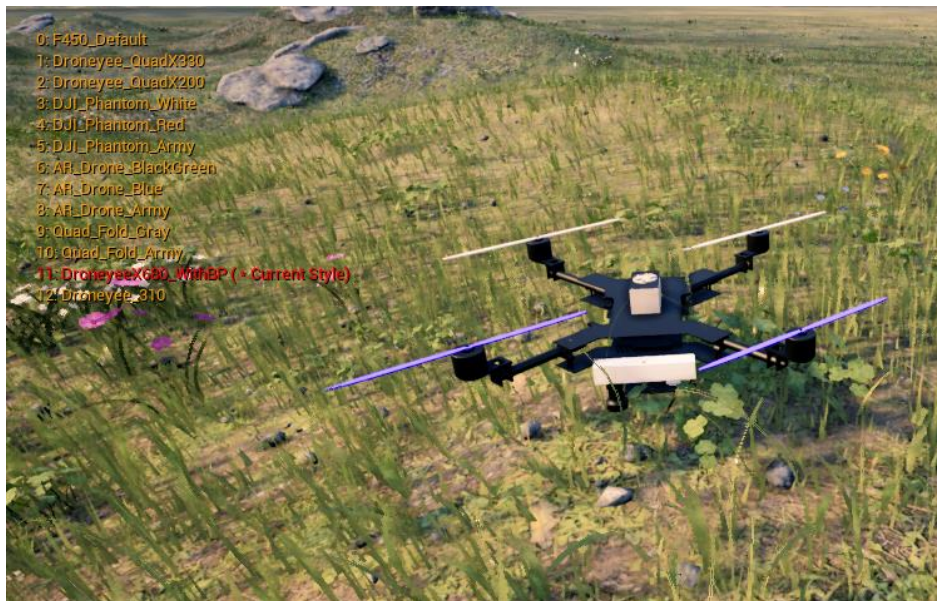


图 12

这样创建出来的无人机 ID 是从 1000 往后递增（也可以按 S 查看它的 ID）。

然后使用命令：“RflySetActuatorPWMs 1000 60 80 100 120 0 0 0”，可以看见四个旋翼开始旋转，分别以每秒  $60^\circ$ 、 $80^\circ$ 、 $100^\circ$ 、 $120^\circ$  在旋转。



效果图

现象解释：使用 RflySetActuatorPWMs 命令时，如果目标无人机是一个蓝图类的实例，那么就会调用它的 ActuatorInputs 事件，并把这 8 个参数全都传给它，而我们在 ActuatorInputs 事件中编写的就是前 4 个值分别控制 4 个旋翼的旋转速度。

我们还另写了一个 “ActuatorInputsExt” 事件，当使用 RflySetActuatorPWMsExt 命令时，如果目标无人机是蓝图类的实例、并且 RflySim 是完整版的，则该命令是有效的。如果我们输入命令 “RflySetActuatorPWMsExt 1000 0 0 20 0 0 0 0 0 0 0 0 0 0 0 0”：



---

[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: \*\*\*\*