

| *此处编写实验名称*

| 1. 实验目的

| 2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[^1]。

```
1 ---
2 title: 三维场景交互接口python获取飞机, 物体, 相机信息接口getCamCoptObj验证实验
3 puppeteer:
4   format: "A4"
5   printBackground: true
6   outline: true
7 ---
8
9 <center><h1>三维场景交互接口 Python 获取飞机、物体、相机信息验证实验</h1></center>
10
11 # 1. 实验目的
12
13 了解如何通过 Python 接口获取飞机、场景物体和相机的信息, 并掌握向 RflySim3D 请求与接收这些
14 数据的基本流程与注意事项。
15
16 # 2. 实验要求
17
18 - 软件要求: Windows 10 及以上版本; RflySim 工具链; VS Code。
19 - 硬件要求: 笔记本/台式电脑 1 台 (推荐配置见下文)。
20
21 [^1]: https://rflysim.com/
22
23 # 3. 实验地址
```

例程目录:

[安装目录]\RflySimAPIs\3.RflySim3DUE\0.ApiExps\e6_RflySim3DCtrlAPI\10.GetCamObjDemo

- ./GetCamObjDemo.bat：软件在环仿真实验脚本
- ./GetCamObjDemo.py：Python 实验脚本（请求飞机与物体信息）
- ./GetCamObjDemoWithCam.py：Python 实验脚本（请求飞机、物体与相机信息）

- [./Python38Run.bat](#)：集成的 Python 环境启动脚本
- [./Config.json](#)：取图与相机配置文件
- [./assets/](#)：示例图片及资源目录

4. 实验内容或步骤

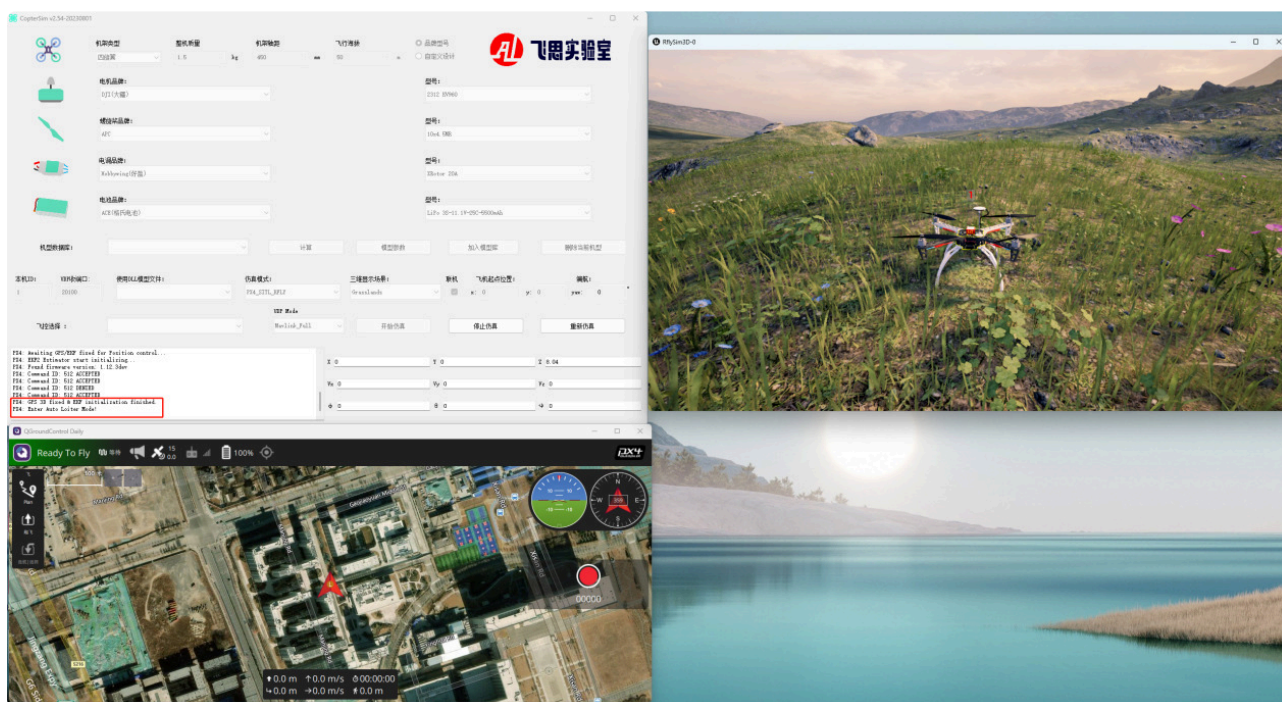
本节将原文档实验步骤按模板编号迁移，保留所有图片与操作要点。所有 Step 均已改为“步骤”。

4.1 Python 请求物体实验（必做）

4.1 步骤1：打开平台

以管理员方式运行 [./GetCamObjDemo.bat](#)，开启软件在环仿真。将会启动 QGC 地面站与 CopterSim，并在 CopterSim 日志栏出现如下字样代表初始化完成：

GPS 3D fixed & EKF initialization finished（表示 RflySim3D 中有 1 架无人机）。



4.1 步骤2：运行 Python 文件

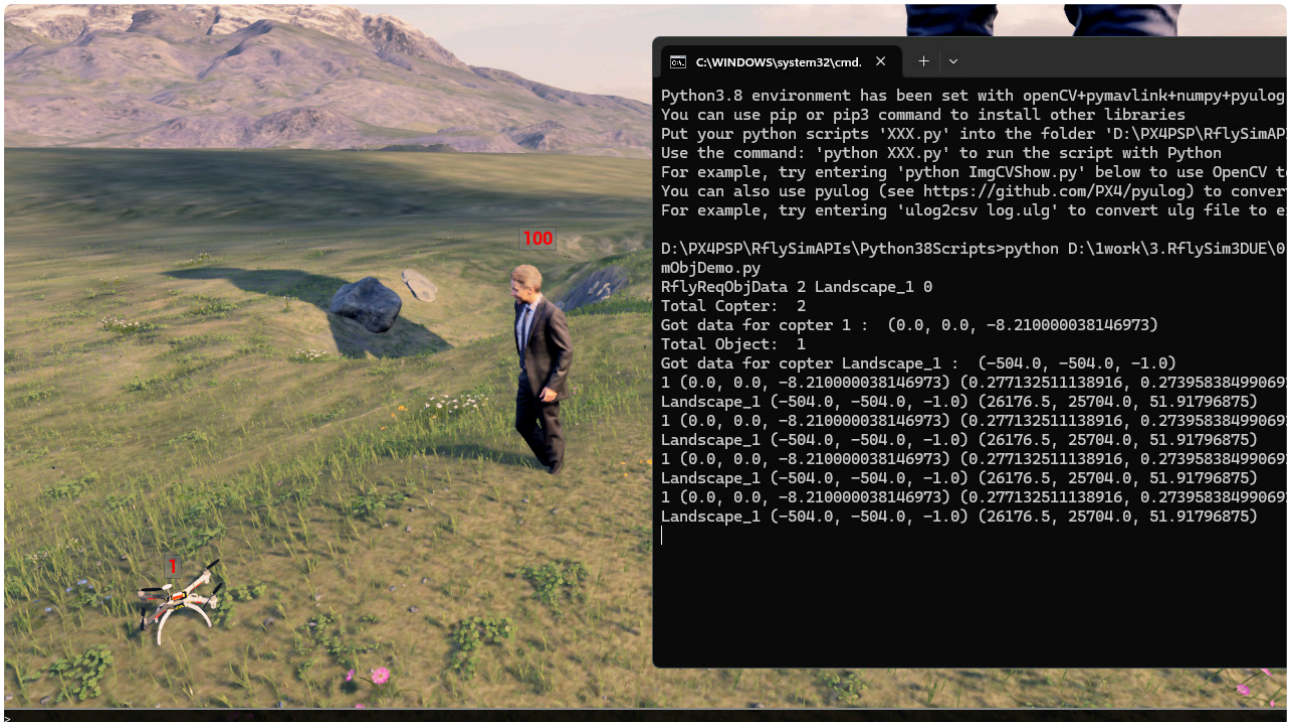
双击 [./Python38Run.bat](#) 打开集成好的 Python 环境，输入并运行：

```
python ./GetCamObjDemo.py
```

运行后可以获取到飞机和场景物体的信息，并弹出摄像头窗口。请查看脚本内注释以理解实现原理。详细协议见参考资料。

```
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

D:\1work\3.RflySim3DUE\0.ApiExps\e6_RflySim3DCtrlAPI\10.GetCamObjDemo>python GetCamObjDemo.py
```



4.1 步骤3: 关闭 demo

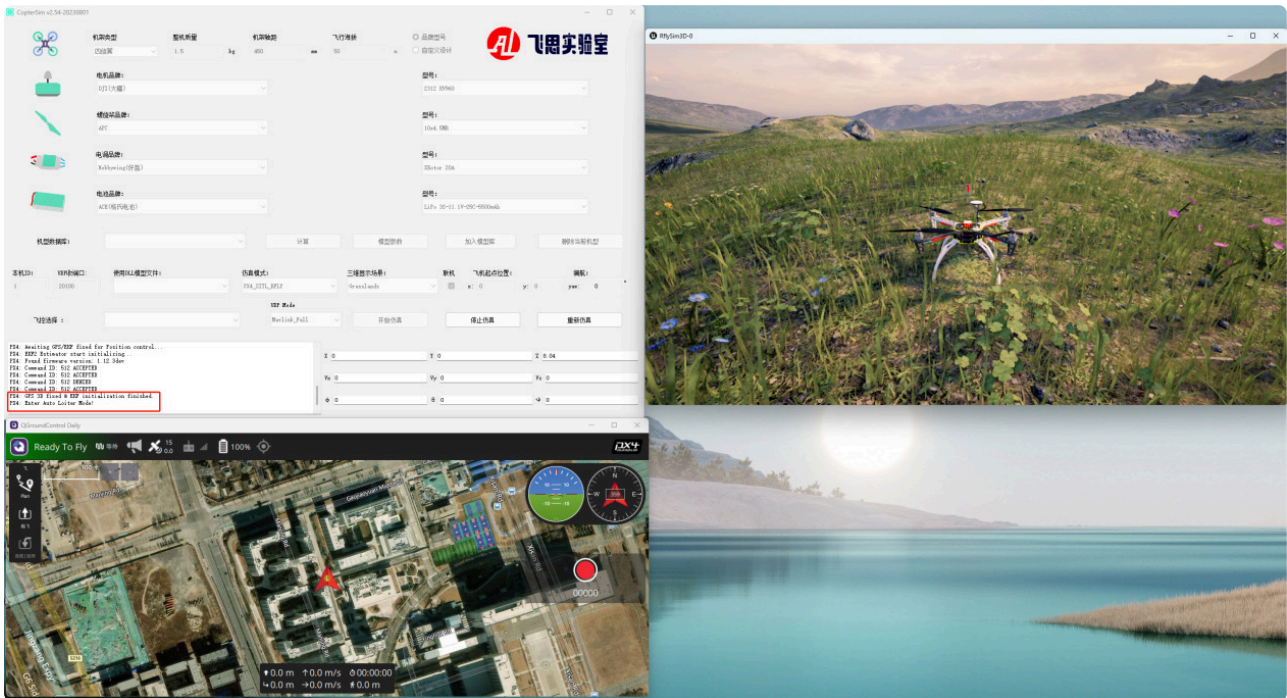
在显示平台的 CMD 窗口中按任意键可关闭平台，关闭用于运行 Python 的 CMD（或直接关闭 Python 进程）以退出实验。

```
-----
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../../logs
Using Airframe File: 10016_iris
starting instance 1 in /mnt/d/PX4PSP/Firmware/build/px4_sitl_default/instance_1
Copying rcS files
PX4 instances start finished
Press any key to exit
_
```

4.2 Python 请求物体和相机实验（必做）

4.2 步骤1：打开平台

以管理员方式运行 [GetCamObjDemo.bat](#)，开启一个飞机的软件在环仿真。将会启动1个QGC地面站，1个CopterSim软件且其软件下侧日志栏必须打印出GPS 3D fixed & EKF initialization finished字样代表初始化完成，并且RflySim3D软件内有1架无人机。



4.2 步骤2：运行 Python 文件

```
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

D:\1work\3.RflySim3DUE\0.ApiExps\66_RflySim3DCtrlAPI\10.GetCamObjDemo>python GetCamObjDemoWithCam.py
```

在文件夹下，双击Python38Run.bat，打开集成好的python环境，输入 `python GetCamObjDemoWithCam.py`，回车运行。

可以获取到飞机和物体的信息，以及摄像头窗口，创建的几个物体，如下图所示。请查看注释理解python实现原理。详细原理见“求取标志点协议v3.txt”。



4.2 步骤3: 关闭 demo

同 4.1 步骤3, 在对应的 CMD 中退出或关闭运行环境。

```
-----  
Start QGroundControl  
Kill all CopterSims  
Starting PX4 Build  
[1/1] Generating ../../logs  
Using Airframe File: 10016_iris  
starting instance 1 in /mnt/d/PX4PSP/Firmware/build/px4_sitl_default/instance_1  
Copying rcS files  
PX4 instances start finished  
Press any key to exit  
_
```

4.3 Vscode 调试运行实验 (选做)

准备工作:

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\3.PythonConfig\Readme.pdf](#) 步骤, 正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同, 在运行python文件时, 可使用VS Code (或Pycharm等工具) 来打开python文件文件, 并阅读代码, 修改代码, 调试执行等。

扩展实验:

- 请自行使用VS Code阅读例程中的python源码, 通过程序跳转, 了解每条代码的执行原理; 再通过调试工具, 验证每条指令的执行效果。
- 请尝试修改代码, 修改Config.json的相机设置测试不同的相机配置。

5. 关键知识点

关键知识点1: type 与 objName 的含义

type 字段表示请求对象的类型:

- 0: 相机 (此时 objName 对应相机的 seqID, int 或 list)
- 1: 飞行器 (此时 objName 对应 CopterID, int 或 list)
- 2: 场景物体 (此时 objName 对应场景内物体名称, string 或 list)

此外, windowID 表示要向哪个 RflySim3D 窗口发送消息, 默认为 0。

关键知识点2: UE4CtrlAPI.getCamCoptObj 接口

函数签名: UE4CtrlAPI.getCamCoptObj(type: int = 1, objName: (int, str) = 1)

该函数从 RflySim3D 获取指定的数据。接收端依赖于后台的消息监听循环

(UE4MsgRecLoop) 将收到的相机、飞机和物体数据分别存入三个列表中; 因此在调用本函数之前, 需要先使用 reqCamCoptObj (或通过 RflyReqObjData 命令) 向 RflySim3D 请求相关数据。

参数说明: type=0 表示相机, 1 表示飞机, 2 表示物体; objName 对应各类型的标识 (见关键知识点1)。

关键知识点3: RflyReqObjData 命令与 opFlag

通过 UE 命令行调用: RflyReqObjData opFlag objName colorflag

opFlag 含义:

- 0: 创建一个相机; 1: 创建一个飞机; 2: 创建一个物体
- 10: 删除一个相机; 11: 删除一个飞机; 12: 删除一个物体
- 20: 清除所有相机; 21: 清除所有飞机; 22: 清除所有物体; 23: 清除物体+飞机;
24: 清除物体+飞机+相机

objName: 飞机时为 CopterID (如 1、2、3), 场景物体为对象名称 (如 Landscape_1), 相机为相机序号 (如 0、1、2)。

colorflag: 用于设置显示颜色 (例如用于分割图), 可为字符串 ("red")、整数索引或 RGB 字符串 (如 "255:0:255")。

6. 参考资料

1. [./求取标志点协议v3.txt](#): 实验涉及的协议与数据结构详解 (包含 CoptReqData、ObjReqData、CameraData 等结构说明)。
2. [RflySim 官方文档](#)
3. 本例程中的脚本文件:
 - [./GetCamObjDemo.py](#)
 - [./GetCamObjDemoWithCam.py](#)

7. 常见问题

Q1: 运行 `GetCamObjDemo.bat` 后看不到 RflySim3D 中的无人机或在 CopterSim 日志未出现初始化完成信息。

A1: 请以管理员权限运行 bat 文件; 确认本机已安装 RflySim 工具链并按推荐配置启动; 检查防火墙或网络设置 (本实验使用本机组播/UDP 通信), 必要时临时关闭防火墙以排除干扰。

Q2: 运行 Python 脚本时报错: 找不到 Python 或模块/依赖错误。

A2: 请使用 `Python38Run.bat` 启动集成环境, 或在系统中安装与脚本兼容的 Python (建议 Python 3.8)。确认所需第三方包已安装 (查看脚本顶部的 import 列表), 若使用 VS Code, 请确保选择正确的 Python 解释器并在终端中运行 `python GetCamObjDemo.py`。

Q3: 图片或资源无法显示（例如 assets 下图片无法加载）。

A3: 请确认当前工作目录为本例程目录（即包含 `assets` 文件夹的目录），并使用相对路径访问图片。若通过其他编辑器或环境打开，确保渲染器支持相对路径的资源引用。