三维场景交互接口RflySim3D更新状态获取接口（python getUE4Pos）

1. 实验目的

通过平台提供的 Python 接口获取 RflySim3D 内所有动态创建物体的位置和碰撞数据，用于验证场景中动态对象状态采集与碰撞检测逻辑。

2. 实验要求

- 软件要求：Windows 10 及以上；RflySim 工具链；VS Code 或其它 Python 开发环境。
- 硬件要求：笔记本/台式机 1 台（推荐配置见 RflySim 官网）。

3. 实验地址

例程目录：
[安装目录]\RflySimAPIs\3.RflySim3DUE\0.ApiExps\e6_RflySim3DCtrlAPI\9.RflySim3DPosGet

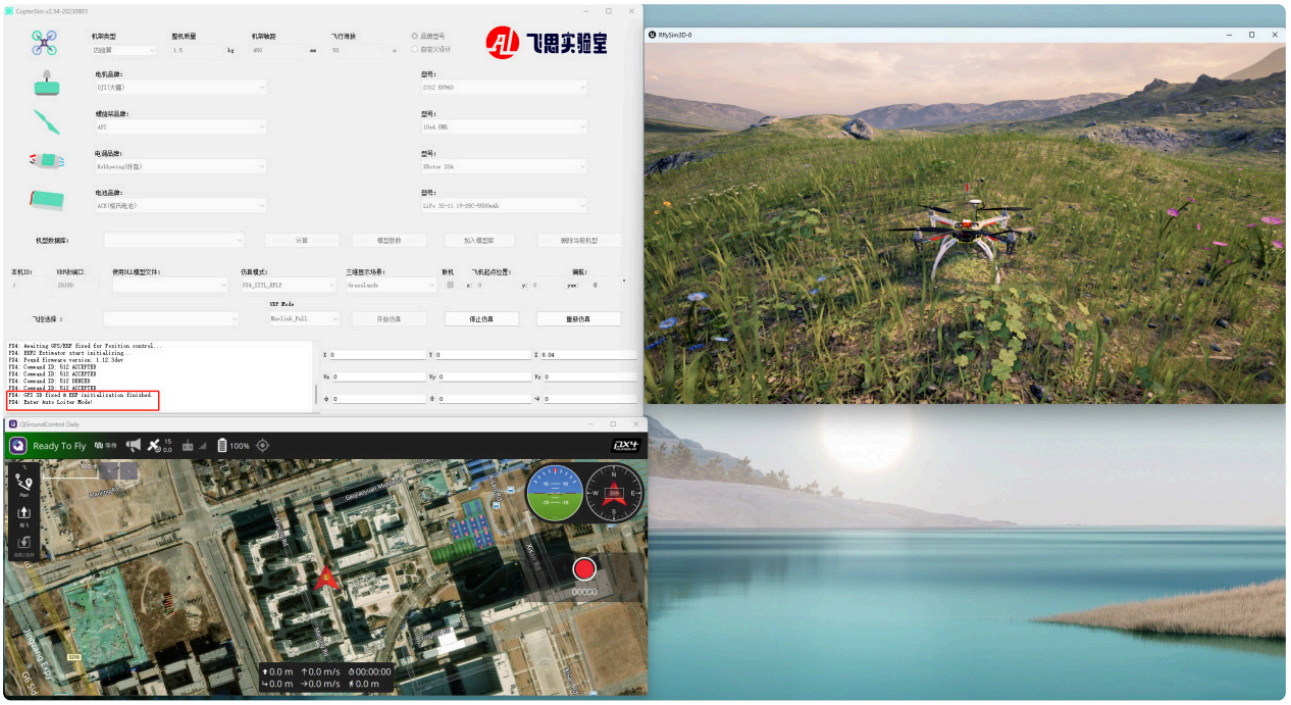
- ./Readme.pdf：RflySim3D 常用功能快速调用接口实验原理
- ./GetUE4PosAPI.bat：软件在环仿真实验脚本（用于启动 QGC、CopterSim、RflySim3D）
- ./GetUE4PosAPI.py：Python 实验脚本（获取场景内物体位置与碰撞信息）
- ./Python38Run.bat：Python 环境启动脚本（打开已集成的 Python 环境）
- ./Ue4.bat：打开 RflySim3D（如果存在此脚本）

4. 实验内容或步骤

本实验通过 Python 脚本获取场景中飞机与障碍物的位置信息及碰撞状态。以下步骤严格对应原文档顺序，且将 Step 改为“步骤”，小节编号以 4.x 表示。

4.1 步骤1：打开平台

以管理员方式运行 ./GetUE4PosAPI.bat，开启软件在环仿真；会启动 QGC 地面站、CopterSim 软件和 RflySim3D。
在 CopterSim 的日志栏中应出现 “GPS 3D fixed & EKF initialization finished” 字样，表示初始化完成，并且 RflySim3D 界面内有 1 架无人机，其 CopterID 为 1。



4.2 步骤2：运行 Python 环境

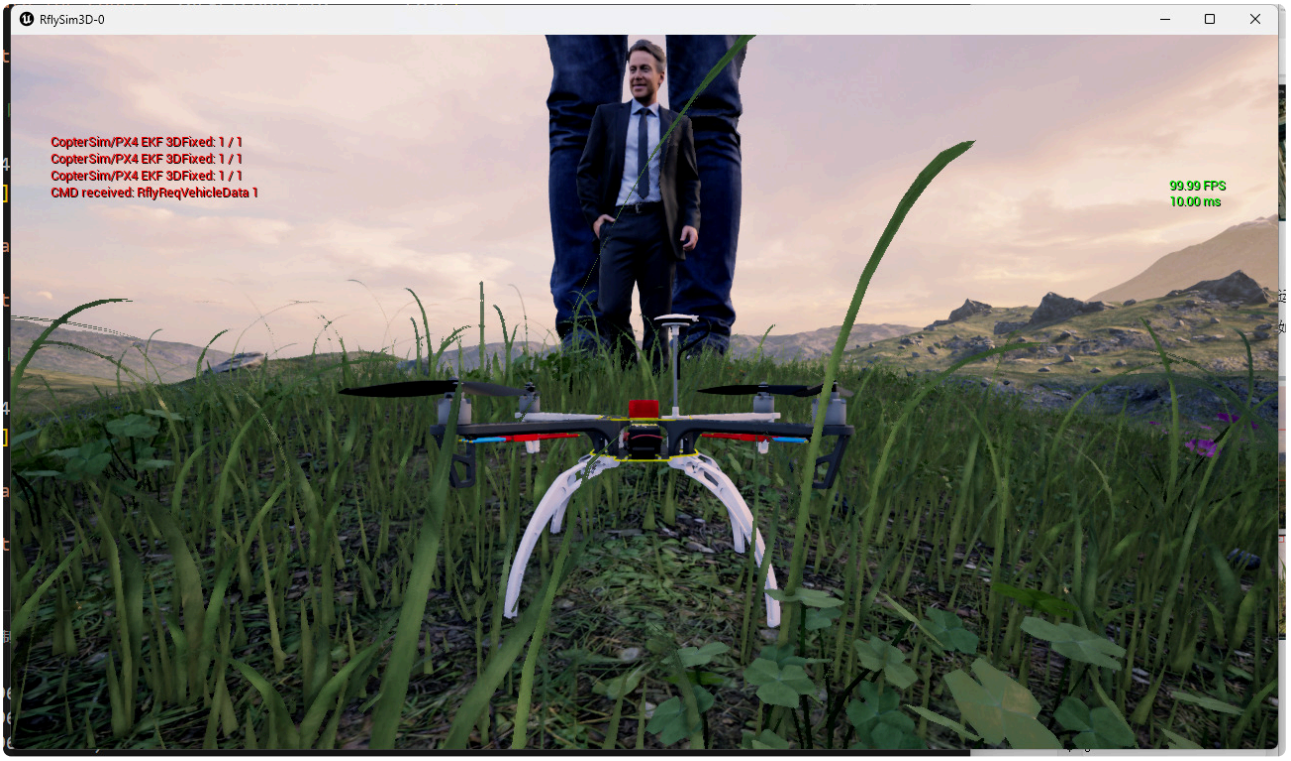
在例程目录下双击 ./Python38Run.bat 打开已集成的 Python 环境。然后在该环境中运行：

- 执行 Python 脚本：运行 `GetUE4PosAPI.py`（可通过在命令行中输入 `python ./GetUE4PosAPI.py` 并回车运行）。

```
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\9.RflySim3DPosGet>python GetUE4PosAPI.py
```

运行后可以获取到场景内飞机和障碍物的信息（程序会打印或记录创建的若干动态物体）。示意图：



此时如果飞机停在地面未起飞，碰撞物体类型可能为 -2（表示地面）：

```
76
问题    输出    终端    调试控制台    端口

1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
1 (0.0, 0.0, -8.039999961853027) -2
```

4.3 步骤3：飞机起飞

在 QGC 中控制飞机起飞，观察程序输出；若无碰撞，碰撞物体类型为 0（表示无碰撞）。

```
1 (0.018122879788279533, 0.014640222303569317, -18.168258666992188) 0
1 (0.01829248107969761, 0.01514009665697813, -18.16765022277832) 0
1 (0.0184441190212965, 0.015618368051946163, -18.167024612426758) 0
1 (0.018617399036884308, 0.01622229814529419, -18.166173934936523) 0
1 (0.018723340705037117, 0.01664811000227928, -18.165538787841797) 0
1 (0.01880689337849617, 0.017049528658390045, -18.1649112701416) 0
1 (0.01888456754386425, 0.01754816435277462, -18.164091110229492) 0
1 (0.018926674500107765, 0.01800825446844101, -18.163293838500977) 0
1 (0.01893548108637333, 0.018327251076698303, -18.162717819213867) 0
1 (0.018924251198768616, 0.01862180419266224, -18.16216468811035) 0
1 (0.01889261044561863, 0.018890563398599625, -18.16164207458496) 0
1 (0.018841609358787537, 0.01913336291909218, -18.16115951538086) 0
1 (0.018773049116134644, 0.019350789487361908, -18.160722732543945) 0
1 (0.018688026815652847, 0.019543945789337158, -18.16033363342285) 0
```

4.4 步骤4：关闭 demo

在由 ./GetUE4PosAPI.bat 启动的命令提示符（CMD）窗口中，按回车（或任意键）可以快速关闭 CopterSim、QGC、RflySim3D 等所有进程，结束实验。

4.5 可选：VS Code 调试运行（选做）

准备工作：确保已按照 RflySimAPIs 中 Python 环境配置步骤完成 VS Code 的 Python 配置（参见：`../1.RflySimIntro/2.AdvExps/e3.PythonConfig/Readme.pdf`），或者使用已配置的 PyCharm 等开发环境。

操作要点：

- 使用 VS Code 打开 `GetUE4PosAPI.py`，可设置断点、逐步调试，或修改打印内容以查看更多字段。
- 建议阅读例程源码并结合 API 手册了解每条指令的含义。

5. 关键知识点

以下关键知识点从原文档移植并做为二级标题呈现；如果有多个关键知识点则按编号逐项列出。

关键知识点1：UE4MsgRecLoop（UDP 消息接收循环与碰撞检测）

UE4MsgRecLoop 定义了一个死循环，用于不断监听来自 UE4 的 UDP 数据：

主要逻辑：

- 循环判断 `self.stopFlagUE4`，若为 True 则退出循环，停止监听。
- 使用 `buf, addr = self.udp_socketUE4.recvfrom(65500)` 接收 UDP 数据包（buf 为数据，addr 为发送端地址）。

碰撞监听示例（CopterSimCrash 结构体）：

结构体定义：

struct CopterSimCrash {

int checksum;

int CopterID;

int TargetID;

}

当接收到长度为 12 字节的数据包时，按三个整数进行解包：

checksum, CopterID, targetID = struct.unpack("iii", buf[0:12])

若 checksum 等于预定义值 1234567890，且 targetID 是有效目标且 CopterID 与当前飞行器 ID 匹配，则设置碰撞标志 `self.isVehicleCrash` 并记录 `self.isVehicleCrashID`，用于后续处理或打印碰撞信息。

6. 参考资料

1. RflySim3D 快捷键接口总览 ../../../API.pdf
2. RflySim3D 控制台命令接口总览 ../../../API.pdf

7. 常见问题

Q1: 无法接收到 UDP 数据，程序一直阻塞或没有输出?

A1: 请检查是否已正确运行 `GetUE4PosAPI.bat` 并确保 QGC、CopterSim 与 RflySim3D 已启动且初始化完成（日志中出现 “GPS 3D fixed & EKF initialization finished”）。另外检查防火墙设置是否阻止了本地 UDP 端口，或端口是否被占用。

Q2: 程序提示 checksum 不匹配或没有检测到碰撞信息?

A2: 校验和 `checksum` 为预定义值（示例 1234567890），若收到数据包但校验失败，可能是协议不匹配或字节序问题。确认发送端与接收端使用相同的 struct 打包格式和大小端（endianness），并保证接收缓冲区长度与发送一致。

Q3: 运行 `GetUE4PosAPI.py` 时找不到某些模块或 Python 环境问题?

A3: 请使用 `Python38Run.bat` 启动示例自带的 Python 环境，或确保本机 Python 环境已安装所需依赖；同时在 VS Code 中选择正确的 Python 解释器。若仍有问题，查看 `GetUE4PosAPI.py` 顶部 import 列表并逐个安装缺失包。