Binding Experiment Between 3D Scene Interactive Interface Models (Python interface sendUE4Attatch)

1. Objective

Learn how to bind UAVs in Rflysim3D via Python.

2. Requirements

- Software requirements: Windows 10 or higher; RflySim toolchain [1]; Python 3.8.
- Hardware requirements: One laptop/desktop computer [2].

3. Experiment Location

Routine directory:
[Installation Directory]\RflySimAPIs\3.RflySim3DUE\0.ApiExps\e4_UAVCtrl\4.VehicleAttachPy

- Ue_python.bat: Start RflySim3D and Python
- VehicleAttachAPI.py: Calls functions from the main interface

4. Experiment Content or Steps

4.1 Step 1: Python UAV Binding Experiment (Required)

Double-click ue_python.bat to start RflySim3D and the Python environment.

In the folder, double-click Python38Run.bat to open the integrated Python environment. Run the VehicleAttachAPI.py file in this environment, enter python VehicleAttachAPI.py

```
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\e4_UAVCtrl\4.VehicleAttachPy>python VehicleAttachAPI.py
```

The position of Model 1 updates continuously, and you can observe that the relative positions and attitudes of two models remain unchanged in RflySim3D

Open the Python file with any text editor and change the pitch angle of Model 1. Re-run the file according to Step 2.

```
59    time.sleep(2)
60    ue.sendUE4PoAsScale(1,2030,0,[12,0,-8],[0,30,0],[1,1,1])
```

It can be seen that Aircraft 2 changes its attitude and position centered on Model 1.

4.2 Step 2: VSCode Debugging and Running Experiment (Required)

- First, ensure that the VS Code environment has been properly configured according to the steps in RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig\Readme.pdf. Or configure your own custom Python environment such as PyCharm, etc.
- Other steps are the same as above, when running VehicleAttachAPI.py in Step 2, you can use VS
  Code (or tools like PyCharm) to open the VehicleAttachAPI.py file, read the code, modify the code, debug and execute, etc.
- Please use VS
  Code to read the source code of VehicleAttachAPI.py on your own, understand

the execution principle of each line of code through program navigation; then verify the execution effect of each instruction through debugging tools.

```python
# Create MAVLink control API instance
ue = UE4CtrlAPI.UE4CtrlAPI()


ue.sendUE4Cmd('RflyChangeMapbyName Grasslands')
time.sleep(2)

# Vehicle type 200030 means 2(model: style = 2) * 100000 + 30(Type: a man)
man_vehicale_type = 200030
ue.sendUE4PosScale(1,man_vehicale_type,0,[0,0,-8],[0,0,0],[1,1,1])
# Send and generate a 3D object to RflySim3D, where: the vehicle ID is CopterID=1;
# Vehicle type VehicleType=200030 means Type=30 (a man) with model = 2; RotorSpeed=
time.sleep(0.5)

# Vehicle type VehicleType=3 means a Quadrotor Copter
ue.sendUE4PosScale(2,3,0,[-2,0,0],[0,0,0],[1,1,1])
time.sleep(0.5)

# Let Copter #2 (CopterIDs) attatch to vehicle #1 (AttatchIDs) with attatch Type 3
ue.sendUE4Attatch(2,1,3)
```

- Try to modify the code and experiment with other binding relationships.

5. Key Knowledge Points

Key Point 1: Inter-Vehicle Attachment Relationship Structure

Send the structure defining inter-vehicle attachment relationship to RflySim3D using Python, and observe the attachment relationship in RflySim3D. The structure is defined as follows:

struct VehicleAttatch25 {

int checksum; // Checksum, used to verify data integrity

int CopterIDs[25]; // ID of aircraft

int AttatchIDs[25]; // ID of aircraft being attached to

int AttatchTypes[25]; // Style of attachment, including different modes

}

The API for sending attachment relationship in Python interface is as follows:

ue.sendUE4Attatch(copterIds, attachIds ,attachTypes)

CopterIDs(Any): This parameter represents the ID of the aircraft serving as the attachment point, or the ID of the aircraft actively attaching to another aircraft. It can be a list with a maximum length of 25, specifying up to 25 aircraft IDs.

AttatchIDs(Any): This parameter represents the ID of the aircraft being attached to, or the ID of the aircraft passively receiving the attachment. Similar to CopterIDs, it can also be a list with a maximum length of 25.

AttatchTypes(Any): This parameter defines the type of each attachment. It represents the way the attached aircraft connect, including:

0: Normal mode, relative position and attitude remain unchanged.

1: Relative position mode, relative position remains unchanged, but attitude can change.

2: Relative position + yaw mode, relative position remains unchanged, but pitch and roll remain the same, only yaw can change.

3: Relative position + full attitude mode, both relative position and attitude can change.

6. References

1. XML File Rules
2. RflySim3D Shortcut Keys Interface Overview
3. RflySim3D Console Commands Interface Overview

7. Frequently Asked Questions

Q1: ***

A1: ***

1. https://rflysim.com/ ↩
2. Recommended configuration please see:
   https://rflysim.com/doc/zh/HowToInstall.pdf ↩