

1. 实验名称及目的

1.1. 实验名称

双无人机协同侦察打击实验

1.2. 实验目的

本次实验的目的是模拟一个双无人机协同作战场景，完整地呈现从侦察到打击的全过程。具体来说，首先2号无人机在接收到指令后，独自前往指定区域进行侦察任务，在飞行过程中主动寻找作为目标物的红色气球。当2号无人机成功抵达指定位置并发现红色气球后，会立即将目标的相关信息发送给1号无人机。1号无人机在接收到这些信息后，迅速起飞并准确地飞向目标物所在位置，对红色气球实施打击，以此来验证双无人机系统在协同作战中的信息传递、目标识别与精准打击能力。

注意：此实验应在ROS1中运行。

1.3. 关键知识点

关键知识点1: ReqCopterSim接口使用（自动获取ip接口）

```
1 req = ReqCopterSim.ReqCopterSim() # 获取局域网内所有CopterSim程序电脑IP列表
2
3 TargetIP = req.getSimIpID(StartCopterID) #获取CopterSim的1号程序所在电脑的IP，作为目标IP
4
5 req.sendReSimIP(StartCopterID) # 请求回传数据到本电脑
```

关键知识点2: 视觉接口使用

```
1 vis = VisionCaptureApi.VisionCaptureApi(TargetIP) #创建一个视觉传感器实例，这个实例对应的ip
2
3 vis.jsonLoad(jsonPath = "/root/catkin_two_ws/src/detect/sensor_pkg/Config.json")# 加载
4
5 isSuss = vis.sendReqToUE4(0, TargetIP) # 向 RflySim3D 发送取图请求
6
7 vis.startImgCap() #开启取图循环，执行本语句后，已经可以通过vis.Img[i]读取到图片
8
9 vis.sendImuReqCopterSim(StartCopterID,TargetIP) #发送请求，从目标飞机CopterSim读取IMU数据
```

关键知识点3：相机接收图像数据，将检测结果发布到ROS话题上

```
1 | rospy.init_node('det_node')\# 初始化ROS节点det_node
2 |
3 | self.img1_sub = rospy.Subscriber(img1_topic, Image, partial(self.img_cb,idx=1))\# 订阅
4 |
5 | self.img2_sub = rospy.Subscriber(img2_topic, Image, partial(self.img_cb,idx=2))\# 订阅
6 |
7 | self.task_sub = rospy.Subscriber("/task_state",MissionState,self.task_cb)\#订阅任务状态话
8 |
9 | self.ret_pub = rospy.Publisher("/objects",Objects,queue_size=10)\#发布目标检测结果
10 |
11 | img_yolo, det, dt = self.yolo_detector(self.color_img) \#调用了Yolo_Detect类的实例，进行
12 |
13 | self.ret_pub.publish(self.objs)\# 将检测结果发布到ROS话题上
```

关键知识点4：UE控制

```
1 | ue = UE4CtrlAPI.UE4CtrlAPI()\#创建 UE 控制实例
2 |
3 | ue.sendUE4Cmd('r.setres 720x405w',0) \#设置UE4窗口分辨率，注意本窗口仅限于显示，取图分辨率在js
4 |
5 | ue.sendUE4Cmd('t.MaxFPS 20',0) \# 设置UE4最大刷新频率，同时也是取图频率
6 |
7 | ue.sendUE4PosNew(copterID=i+10002,vehicleType=pos[i][0], PosE=pos[i][1:4],AngEuler=an
8 |
9 | ue.sendUE4ExtAct(10002,[0.5,0.5,0.5,1,0,0,0,0,0,0,0,0,0,0,0])\#触发扩展蓝图接口
```

关键知识点5：其余代码说明

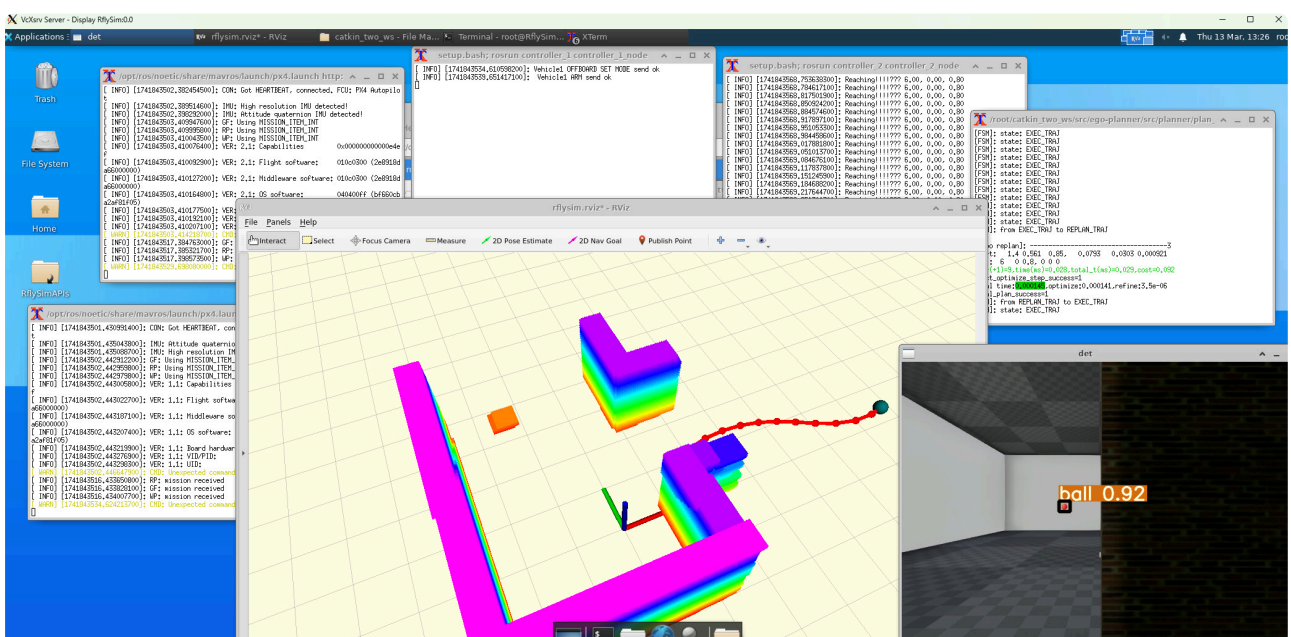
```

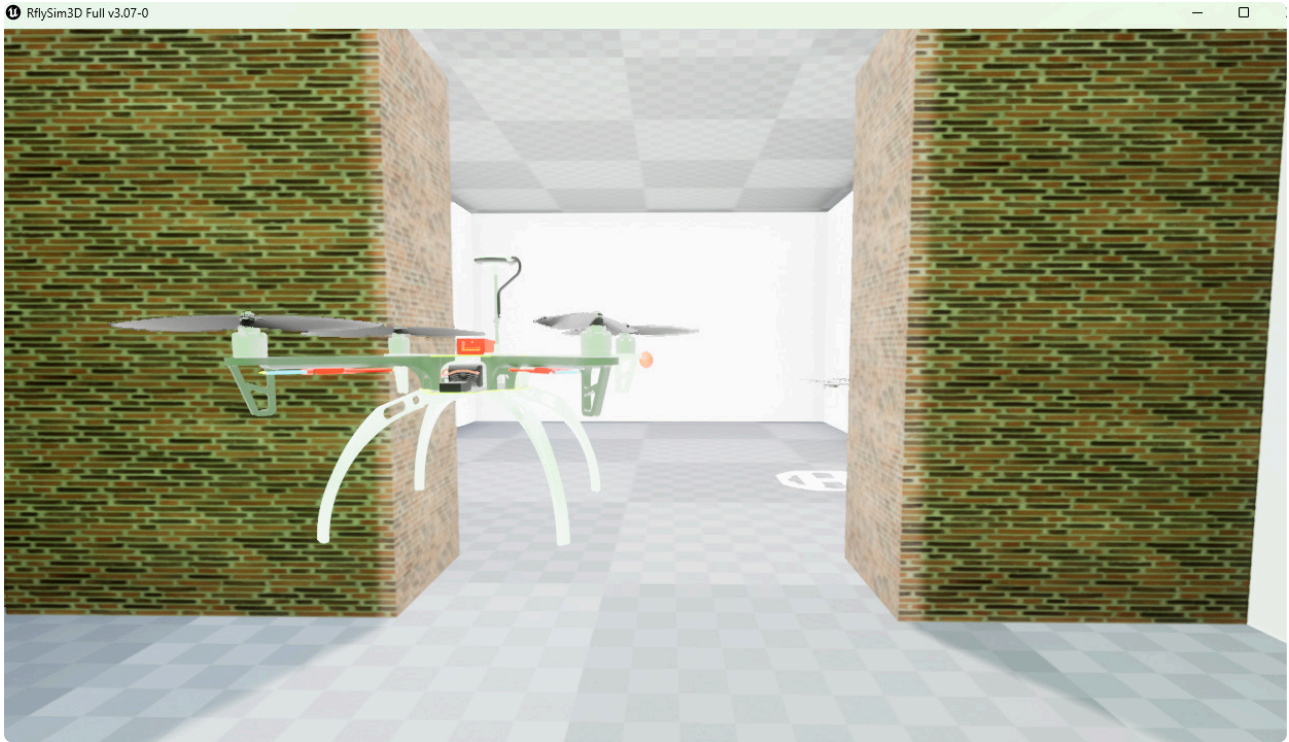
1 // 订阅 MAVROS 状态和本地位置话题
2
3 ros::Subscriber state_sub = nh.subscribe("mavros1/state", 10, state_cb);
4
5 ros::Subscriber local_pos_sub =nh.subscribe\("mavros1/local_posi
6 local_position_cb);
7
8 ros::Subscriber tra_sub = nh.subscribe("/planning/pos_cmd", 10, recv_tra_cb);
9
10 ros::Subscriber flag_sub = nh.subscribe("/hit_flag", 10, flag_cb);
11
12 // 发布位置需求信息、降落标志位、速度需求信息
13
14 ros::Publisher local_pos_pub =nh.advertise<geometry_msgs::PoseStamped>("mavros1/setp
15 10);
16
17 ros::Publisher land_pub = nh.advertise<std_msgs::Bool>("/land_flag", 10);
18
19 ros::Publisher velocity_pub =nh.advertise<geometry_msgs::TwistStamped>("/mavros1/setp
20 10);

```

2.实验效果

运行程序后，可以看到2号无人机在接收到指令后，独自前往指定区域进行侦察任务，在飞行过程中主动寻找作为目标物的红色气球。当2号无人机成功抵达指定位置并发现红色气球后，会立即将目标的相关信息发送给1号无人机。1号无人机在接收到这些信息后，迅速起飞并准确地飞向目标物所在位置，对红色气球实施打击。





3. 文件目录

| 例程目录:

[\[安装目录\]\RflySimAPIs\8.RflySimVision\3.CustExps\9.DualDroneReconStrike](#)

| 文件夹/文件名称 | 说明 |

| [Build_src.sh](#) | 解压、创建工作空间并编译的Bash脚本文件 |

| [src.zip](#) | 实验运行文件夹压缩文件 |

| [WinWSL.bat](#) | 进入 WinWSL 的 Ubuntu 环境的程序运行脚本 |

| [RoomScene_UE4_Cooked.zip](#) | RflySim3D软件仿真环境压缩包 |

| [WslGUI.bat](#) | WSL1/Ubuntu 20.04 可视化界面脚本 |

| [SITLPosStrOnekey.bat](#) | 自动化启动和配置多旋翼无人机仿真环境程序运行脚本 |

| [Src/detect/sh/GO.sh](#) | Shell脚本运行文件 |

4. 运行环境

4.1 软件要求

Windows 10及以上版本; RflySim工具链; Linux (Ubuntu 20.04)。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：<https://rflsim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

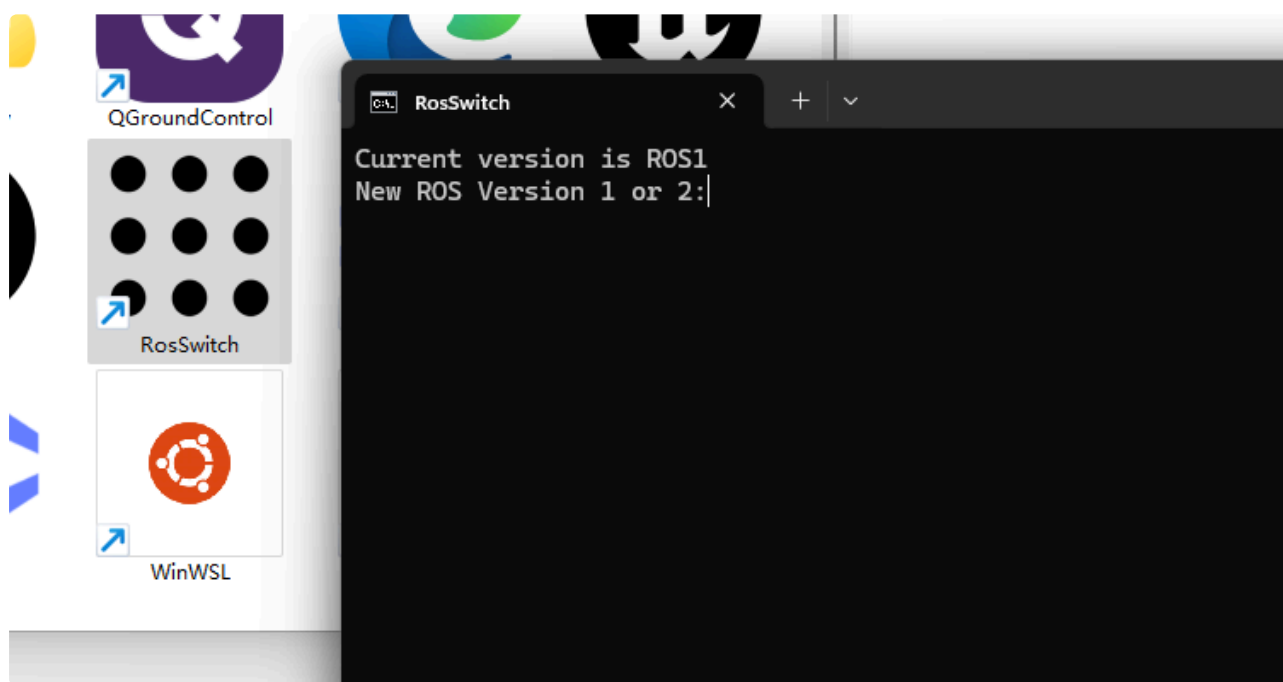
笔记本/台式电脑① 1台；WinWSL 1台。

①：推荐配置请见：<https://rflsim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

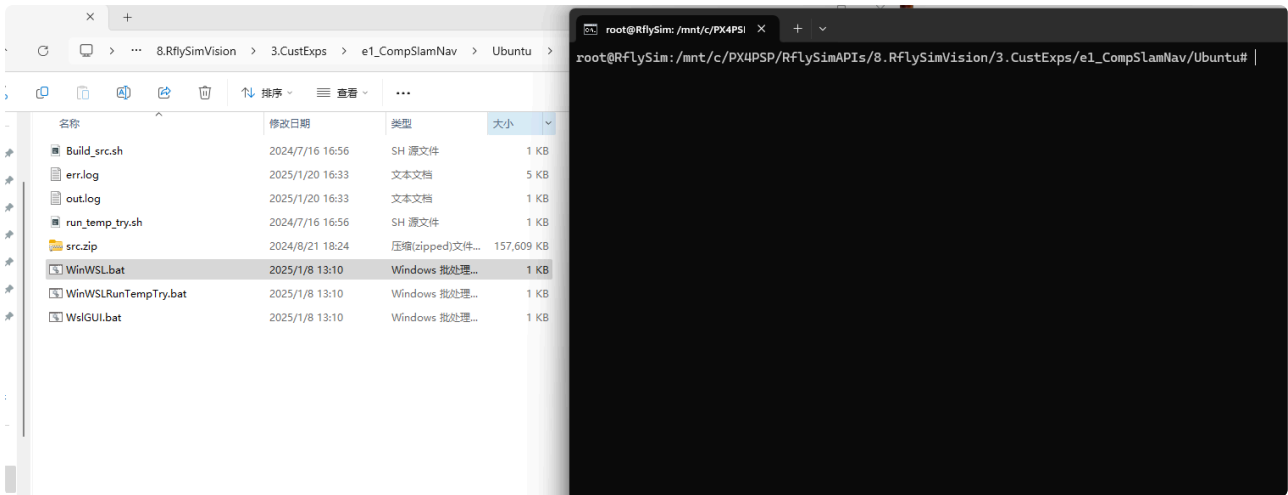
1. 切换ROS版本

打开RflyTools文件夹中的RosSwitch，双击运行，将版本切换到ROS1。



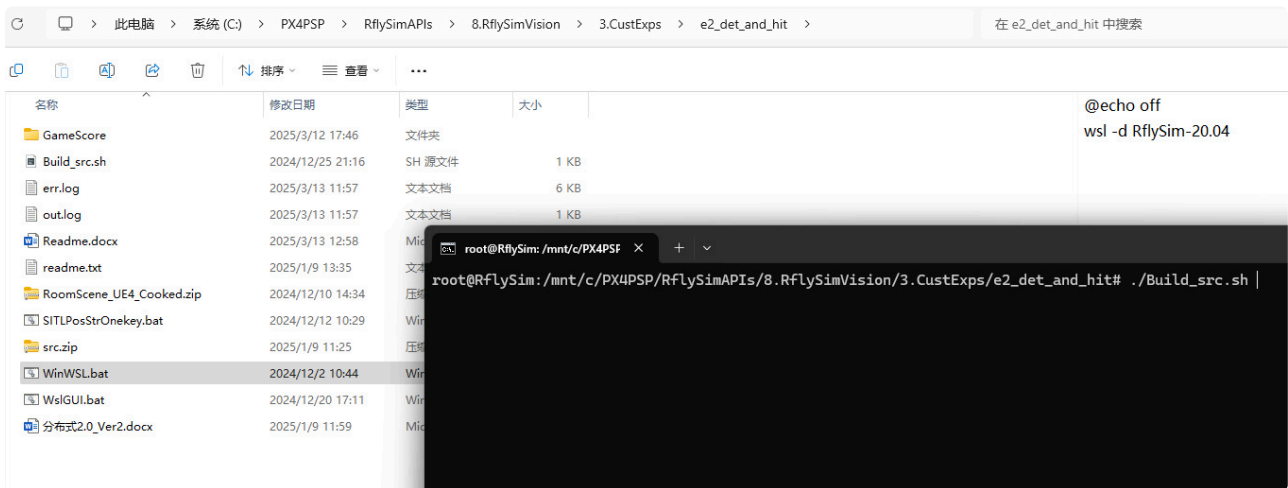
2. 启动Ubuntu环境

双击 [WinWSL.bat](#) 进入WinWSL的Ubuntu环境。输入 `./Build_src.sh`，等待运行完成



3.编译源代码

在Ubuntu环境中运行 `./Build_src.sh`，运行此文件可以将src.zip拷贝到catkin_two_ws并编译。



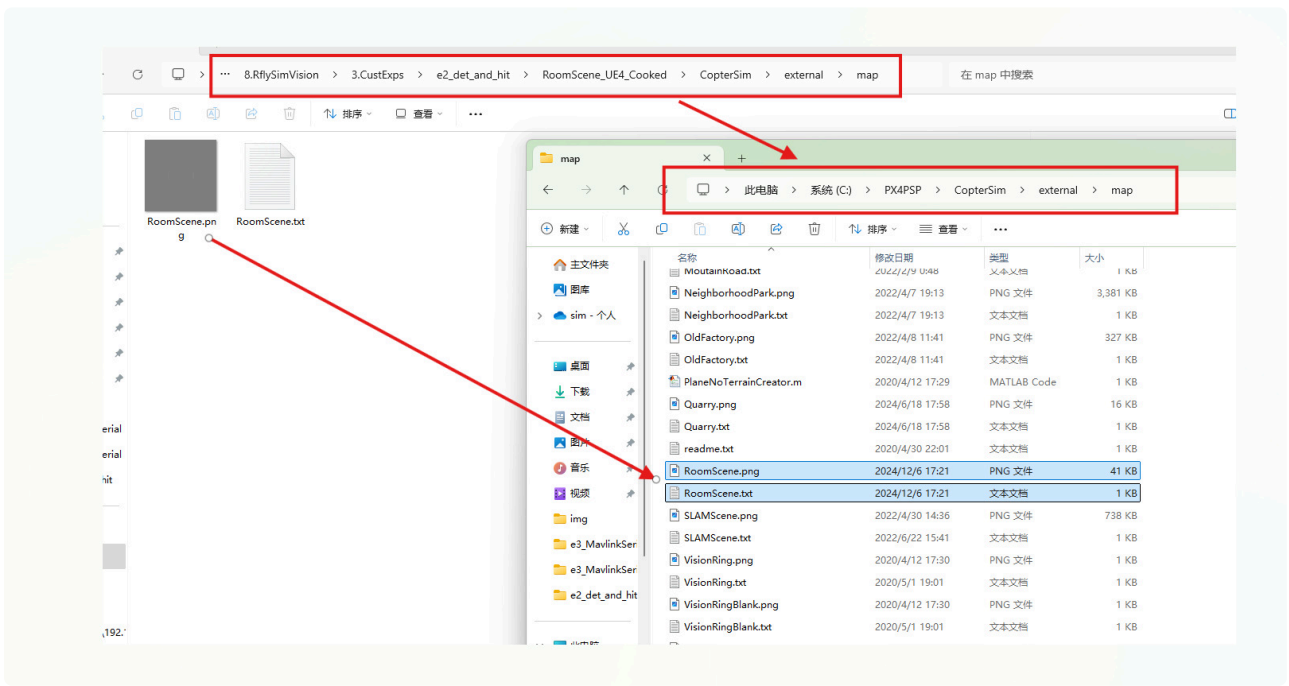
如果编译成功，说明没有问题，编译成功界面如下图所示：

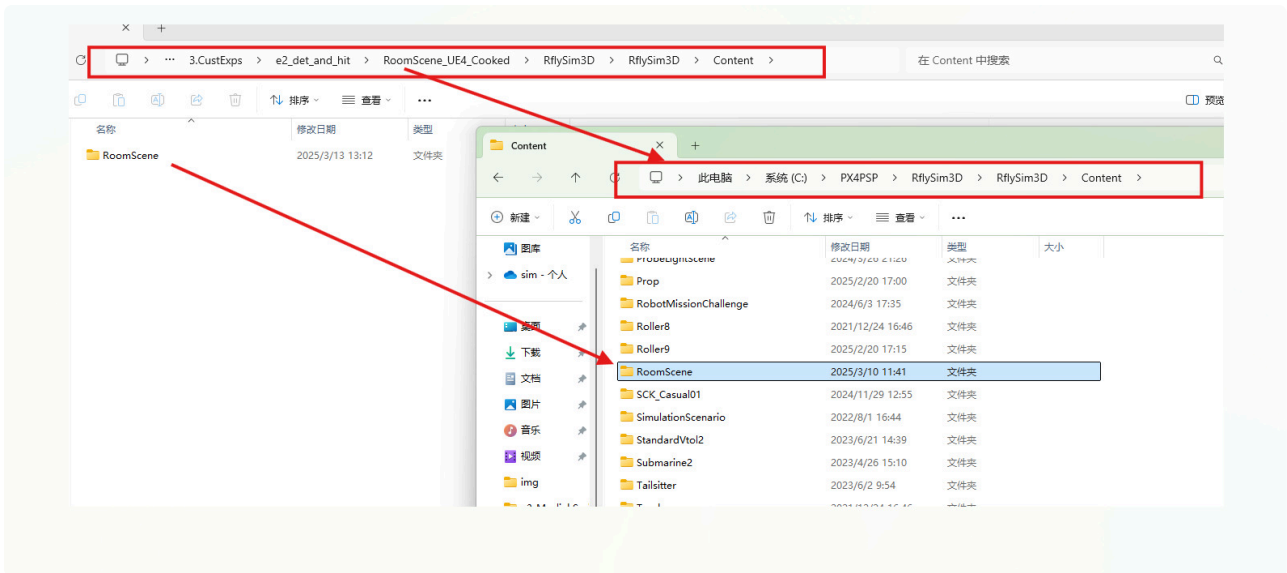
```
root@RflySim: /mnt/c/PX4PSP
[ 92%] Linking CXX shared library /root/catkin_two_ws/devel/lib/libbspline_opt.so
[ 92%] Built target random_forest
[ 92%] Built target pcl_render_node
[ 92%] Linking CXX shared library /root/catkin_two_ws/devel/lib/libpose_utils.so
[ 92%] Built target pose_utils
[ 93%] Building CXX object ego-planner/src/uav_simulator/Utils/odom_visualization/CMakeFiles/odom_visualization.dir/src/odom_visualization.cpp.o
[ 94%] Building CXX object ego-planner/src/uav_simulator/Utils/multi_map_server/CMakeFiles/multi_map_visualization.dir/src/multi_map_visualization.cc.o
[ 94%] Linking CXX executable /root/catkin_two_ws/devel/lib/multi_map_server/multi_map_visualization
[ 94%] Linking CXX shared library /root/catkin_two_ws/devel/lib/python3/dist-packages/cv_bridge/boost/cv_bridge_boost.so
[ 94%] Linking CXX executable /root/catkin_two_ws/devel/lib/odom_visualization/odom_visualization
[ 94%] Built target bspline_opt
[ 95%] Building CXX object ego-planner/src/planner/traj_utils/CMakeFiles/traj_utils.dir/src/planning_visualization.cpp.o
[ 95%] Building CXX object ego-planner/src/planner/traj_utils/CMakeFiles/traj_utils.dir/src/polynomial_traj.cpp.o
[ 95%] Built target cv_bridge_boost
[ 96%] Linking CXX shared library /root/catkin_two_ws/devel/lib/libtraj_utils.so
[ 96%] Built target multi_map_visualization
[ 96%] Built target odom_visualization
[ 96%] Built target traj_utils
[ 96%] Building CXX object ego-planner/src/planner/plan_manage/CMakeFiles/ego_planner_node.dir/src/ego_planner_node.cpp.o
[ 97%] Building CXX object ego-planner/src/planner/plan_manage/CMakeFiles/ego_planner_node.dir/src/ego_replan_fsm.cpp.o
[ 97%] Building CXX object ego-planner/src/planner/plan_manage/CMakeFiles/ego_planner_node.dir/src/planner_manager.cpp.o
[ 98%] Building CXX object ego-planner/src/planner/plan_manage/CMakeFiles/traj_server.dir/src/traj_server.cpp.o
[ 98%] Linking CXX executable /root/catkin_two_ws/devel/lib/ego_planner/traj_server
[100%] Linking CXX executable /root/catkin_two_ws/devel/lib/ego_planner/ego_planner_node
[100%] Built target traj_server
[100%] Built target ego_planner_node
root@RflySim: /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/3.CustExps/e2_det_and_hit#
```

注意：如果在执行此步就有报错，显示没有xxx目录，说明生成的catkin_two_ws文件夹不是名字为catkin_two_ws的文件夹，可能包含有乱码后缀如catkin_two_ws#。把catkin_two_ws文件夹删除，去回收站也删除，确保系统中不含有catkin_two_ws文件夹。

4.导入UE4场景

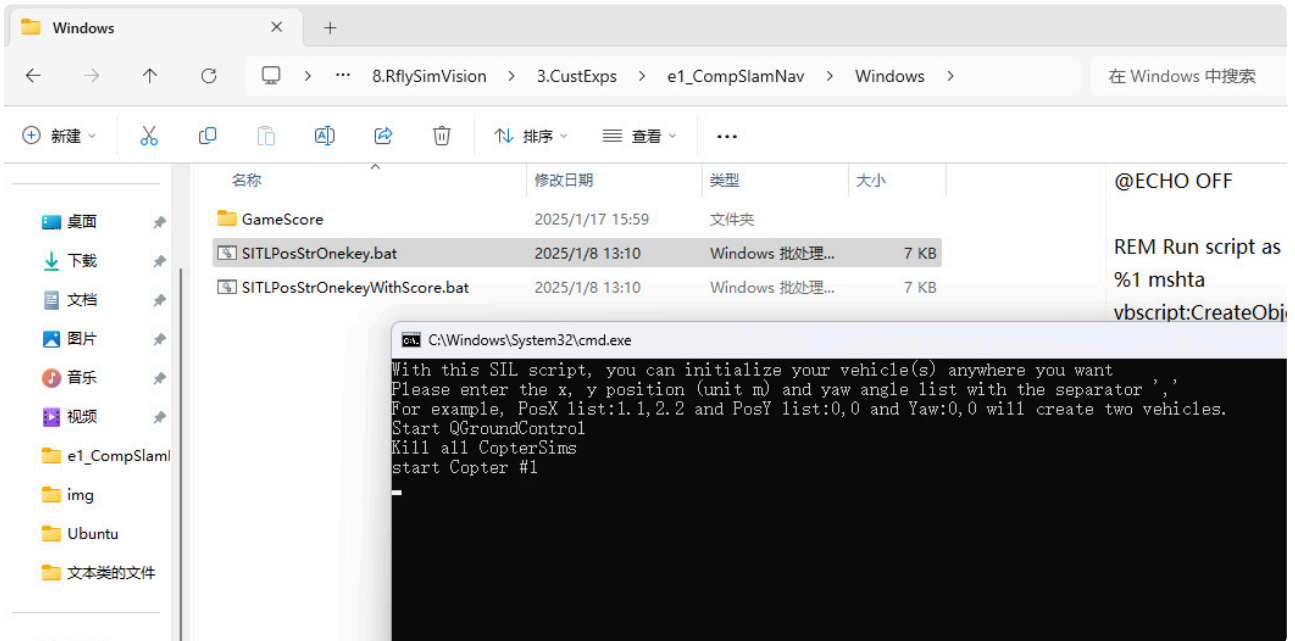
解压RoomScene_UE4_Cooked.zip，内容文件复制到C:\PX4PSP对应目录下，导入UE4场景。





5.启动SITL程序

双击 [SITLPosStrOnekey.bat](#) 启动SITL



此时会启动RflySim3D以及CopterSim:


```
Terminal - root@RflySim: ~/catkin_two_ws
File Edit View Terminal Tabs Help
root@RflySim:~/catkin_two_ws# source ./devel/setup.bash
root@RflySim:~/catkin_two_ws#
```

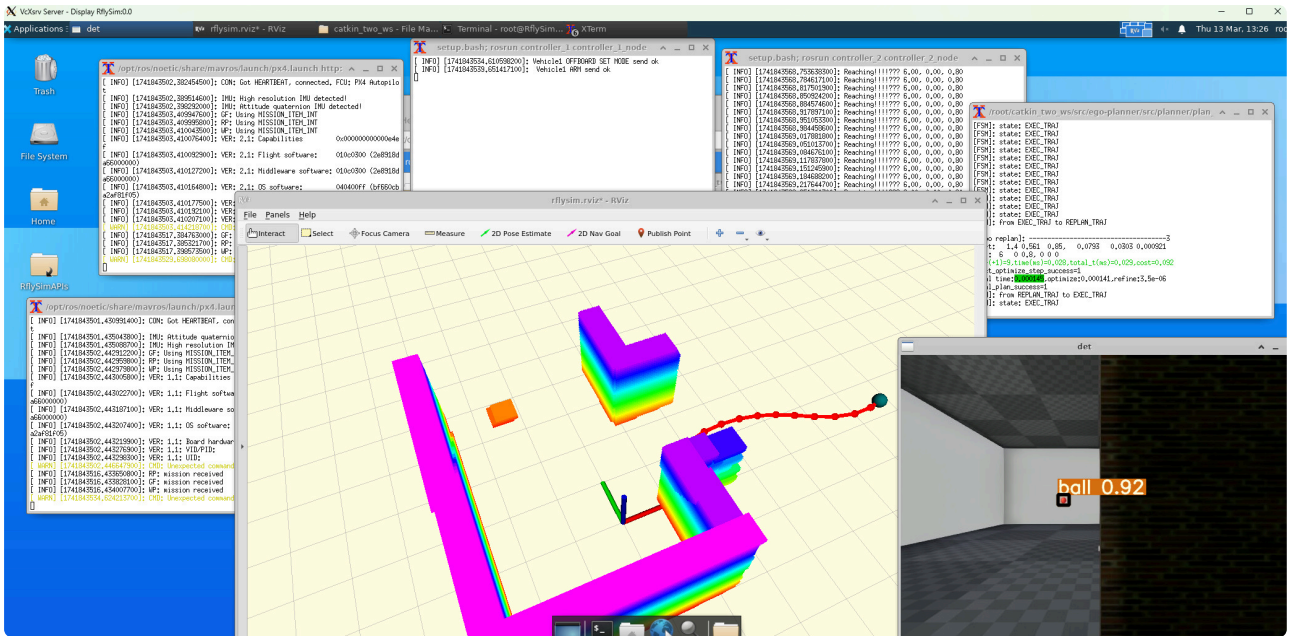
继续在该窗口中进入 `/src/detect/sh/` 目录下，运行 `cd src/detect/sh/`

输入 `./G0.sh`，开始仿真

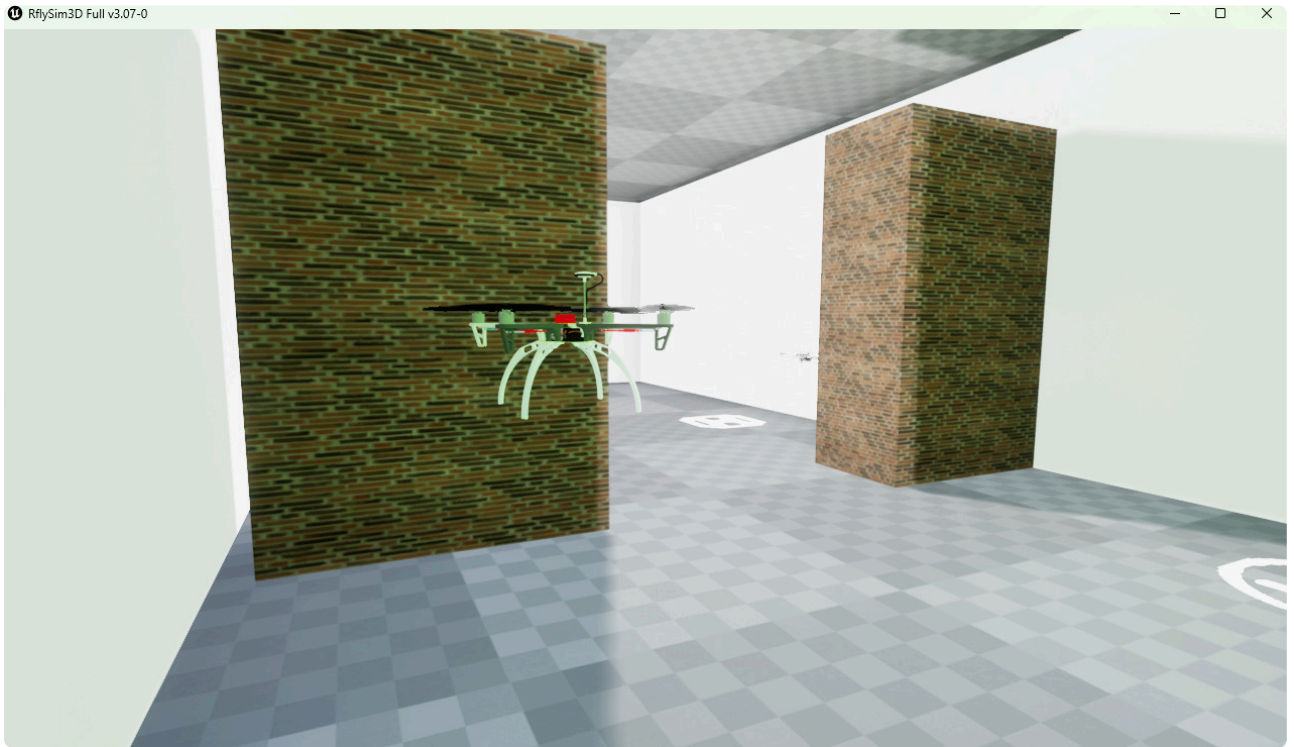
```
Terminal - root@RflySim: ~/catkin_two_ws/src/detect/sh
File Edit View Terminal Tabs Help
root@RflySim:~/catkin_two_ws# source ./devel/setup.bash
root@RflySim:~/catkin_two_ws# cd src/detect/sh/
root@RflySim:~/catkin_two_ws/src/detect/sh# ./G0.sh
Simulation Begin!

```

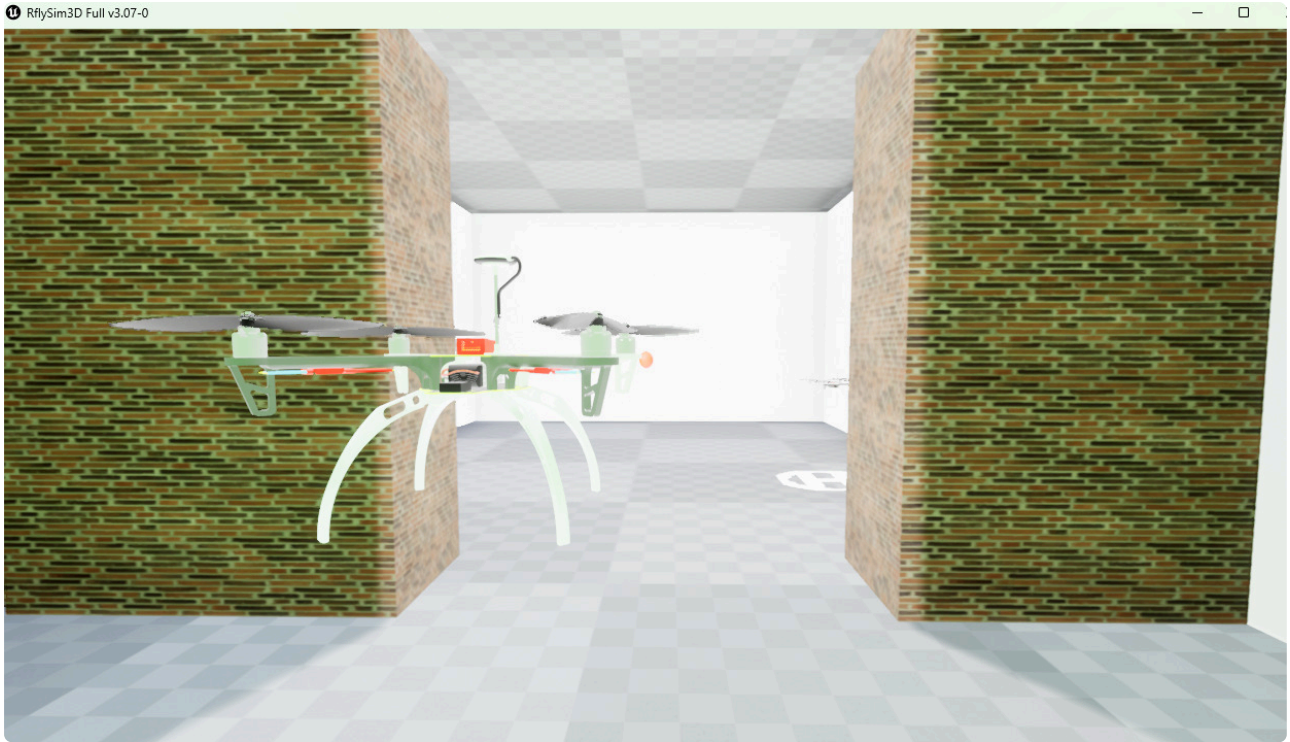
程序运行后，在WSL下可以观察到如下窗口，并在Rviz下可视化了规划的路径，并在det窗口可视化了目标检测的结果。



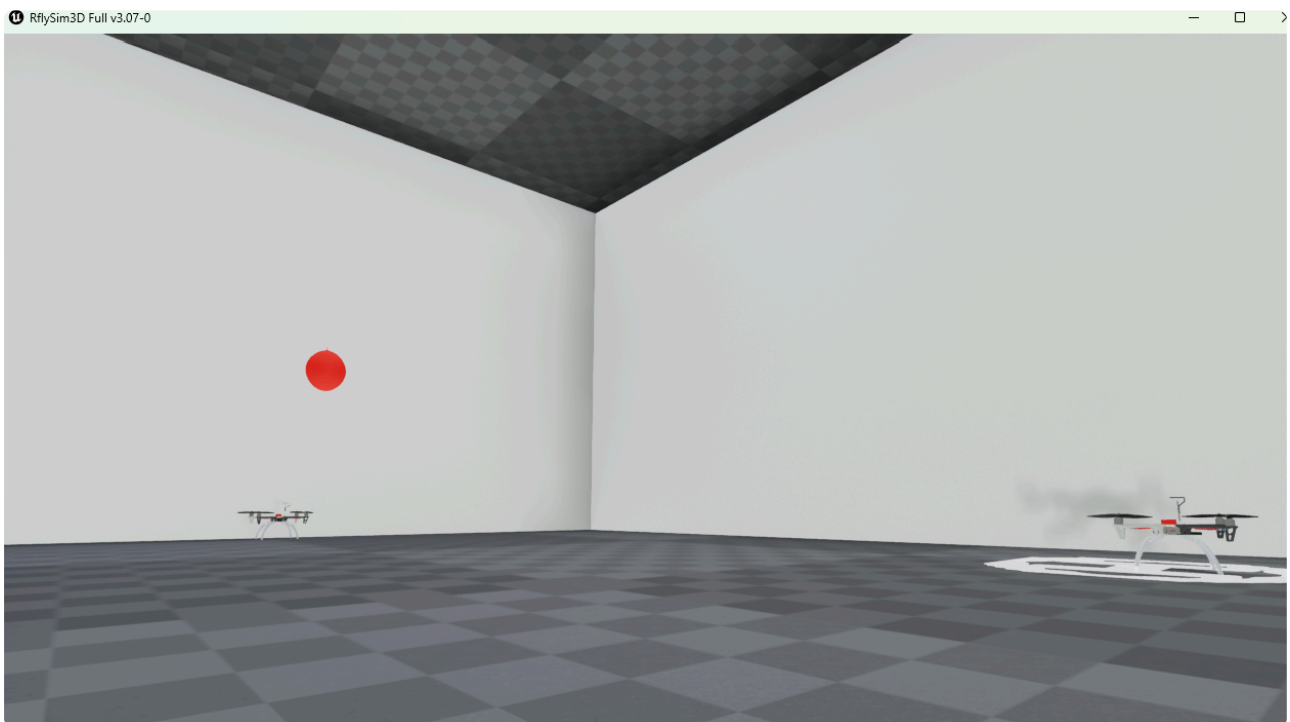
在RflySim3D窗口中，可以看到两架飞机起飞，2号飞机首先规划路径前进并寻找目标物。



在二号飞机运动到指定位置并且已经发现目标物时，1号飞机接收到信息，并开始运动到指定位置，对目标点进行快速打击。



1号飞机对目标点进行打击完成后，1号、2号任务完成，分别降落到地面。



8. 结束仿真

在脚本开启的命令提示符 CMD


窗口中，按下回车键（任意键）就能快速关闭CopterSim、QGC、RflySim3D等所有程序。

6.参考资料

无。

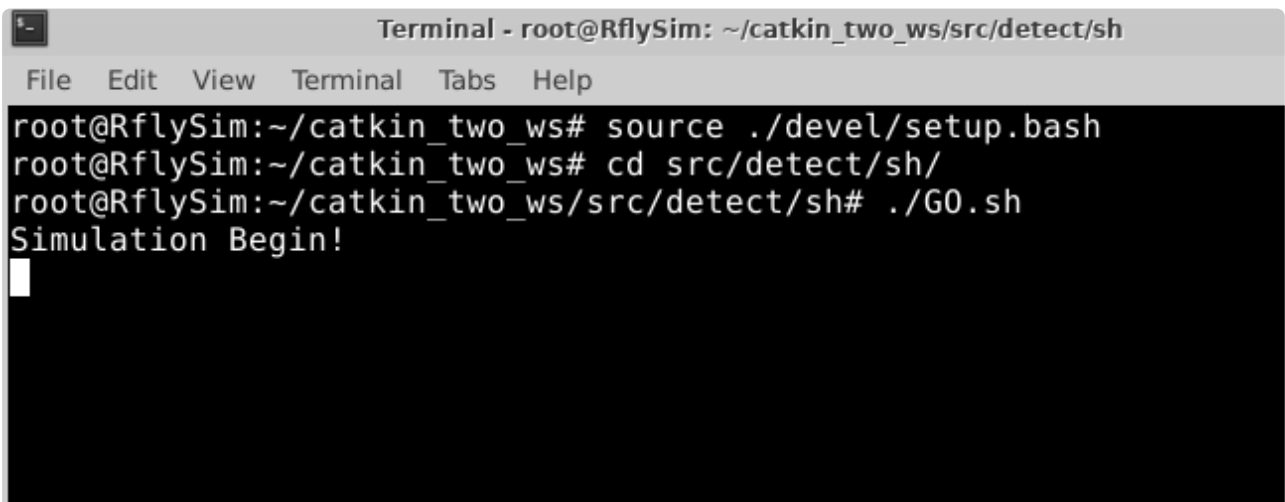
7.常见问题

Q1: 如果运行时出现了如下报错:



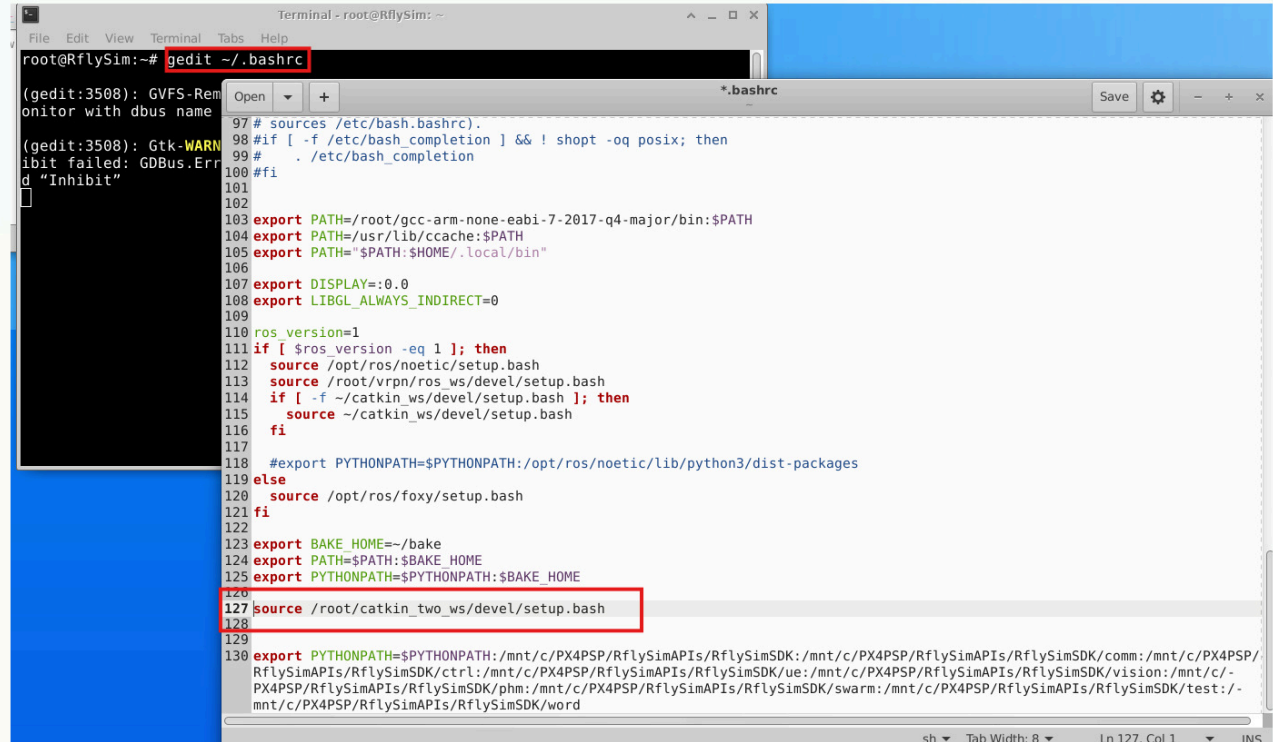
```
scripts && python3 det.py
Traceback (most recent call last):
  File "det.py", line 12, in <module>
    from common_msgs.msg import Objects
ModuleNotFoundError: No module named 'common_msgs'
```

A1: 这是因为没有在工作空间下运行 `source ./devel/setup.bash` 命令, 该命令可以让工作空间的环境变量设置可以生效。(注意, 运行 `source ./devel/setup.bash` 命令后需要在同一终端下运行文件, 重新打开终端环境变量仍然没有刷新)



```
Terminal - root@RflySim: ~/catkin_two_ws/src/detect/sh
File Edit View Terminal Tabs Help
root@RflySim:~/catkin_two_ws# source ./devel/setup.bash
root@RflySim:~/catkin_two_ws# cd src/detect/sh/
root@RflySim:~/catkin_two_ws/src/detect/sh# ./G0.sh
Simulation Begin!
```

如果每次执行程序不想每次都调用 `./devel/setup.bash` 命令，可以在根目录下的`.bashrc`下添加 `source /root/catkin_two_ws/devel/setup.bash`，这样之后，每次仅在 `/root/catkin_two_ws/src/detect/sh/`，输入 `./GO.sh`，就可开始仿真。



```
Terminal - root@RflySim: ~
root@RflySim:~# gedit ~/.bashrc
(gedit:3508): GVFS-RemoteMonitor with dbus name
(gedit:3508): Gtk-WARNING: GDBus.Error: org.freedesktop.DBus.Error.Inhibit
127 source /root/catkin_two_ws/devel/setup.bash
128
129
130 export PYTHONPATH=$PYTHONPATH:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/comm:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/ctrl:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/ue:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/vision:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/phm:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/swarm:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/test:/mnt/c/PX4PSP/RflySimAPIs/RflySimSDK/word
sh Tab Width: 8 Ln 127, Col 1 INS
```