

---

# 1. 实验名称及目的

## 1.1 实验名称

行为树结合大模型视觉控制实验

## 1.2 实验目的

通过Ollama本地推理服务，加载量化后的千问3-0.6b 模型，实现对自然语言指令的解析与行为树生成，用于无人机控制流程的决策支持，LLM 将用户意图解析为具体的 ROS 控制指令序列，形成“语义→动作”映射；ROS 话题驱动下，PX4 SITL 执行相应机动。

## 1.3 关键知识点

### 1. WSL2 虚拟化与 Linux 兼容性

- WSL2 通过轻量级虚拟机运行完整 Linux 内核，兼容 ROS、PX4、编译工具链，减少环境差异带来的调试开销。
- 采用 NAT 网络模式与 Hyper-V 虚拟交换机，实现 WSL2 与 Windows 主机的端口映射，保障 ROS 与 MAVLink 通信的稳定性。

### 2. 大模型（LLM）本地推理与行为树生成

- Ollama 部署量化后的 Qwen3 模型，实现自然语言指令（如“穿越方框”）到行为树或 ROS 控制指令的映射。
- 行为树封装高层语义决策，ROS 节点按序下发具体控制命令，构建“语义→动作”闭环。

## 2. 实验效果

## 3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\8.RflySimVision\3.CustExps\10.LLM-BehaviorTreeUAVCtrl

文件夹/文件名称	说明
大模型飞机指令.txt	预定义的无人机控制指令集合（起飞、穿框、打气球、降落等）
Build_src.sh	源码编译脚本，用于编译ROS包和依赖库
DroneSITL.bat	启动PX4 SITL仿真的Windows批处理脚本
EnvSet.py	UE4环境设置脚本，创建障碍物、起飞标志、红色气球等仿真元素
sensor.sh	传感器模块启动脚本，依次启动MAVROS、激光里程计、路径规划器、目标检测
WinWSL.bat	启动WSL2终端的Windows批处理脚本
src.zip	压缩的源代码包，包含所有ROS节点和行为树代码
catkin_match/	ROS工作空间根目录
├── src/	ROS工作空间源码目录
├── behaviortree_cpp/	BehaviorTree.cpp行为树框架库
├── common_msgs/	通用ROS消息定义包
├── ego-planner/	Ego-Planner路径规划算法包
├── mission_pkg/	核心任务执行包，包含行为树节点和控制逻辑
├── multi/	大模型接口包，负责自然语言指令解析
├── object_det/	YOLOv5目标检测包，用于识别气球等目标

文件夹/文件名称	说明
└── sensor_pkg/	传感器数据采集包，处理图像和IMU数据
└── test/	测试相关代码和配置文件

## 4.1 软件要求

Windows 10及以上版本（推荐Win11 22H2以上）；RflySim工具链；WinWSL2-GPU增量包。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4\_fmuv6x\_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://rflysim.com/doc/zh/1/Hardware.html>

## 4.2 硬件要求

笔记本/台式电脑 1台。

①：推荐配置请见：<https://rflysim.com/>

# 5. 实验步骤

## 5.1 环境准备

### 1. 安装与配置WSL2

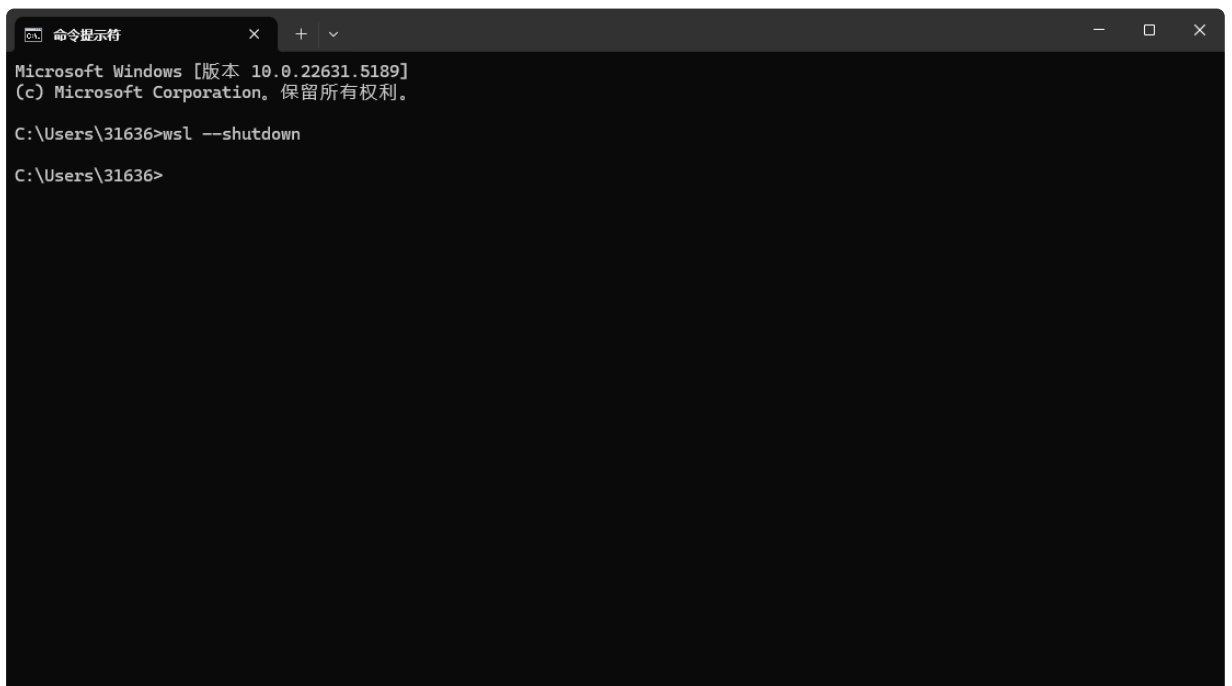
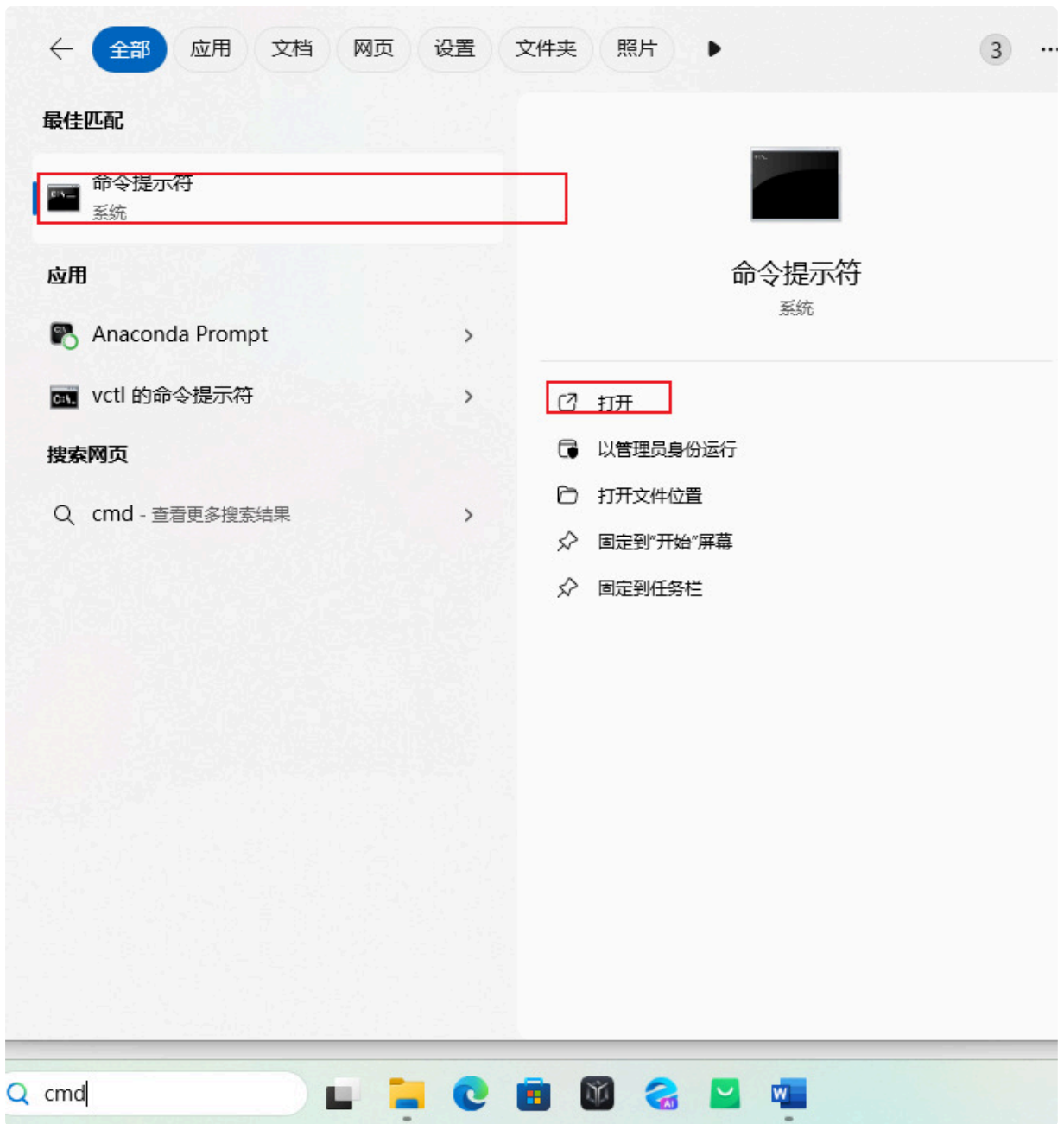
首先参考例程 `1.RflySimIntro\2.AdvExps\e13.WSL2_GPUAccConfig` 完成WSL2的安装，其中已经内置ollama等必要环境,对于Win11 22H2以下的系统，请按照以下步骤进行配置：

- 配置 WSL 网络为 NAT 模式

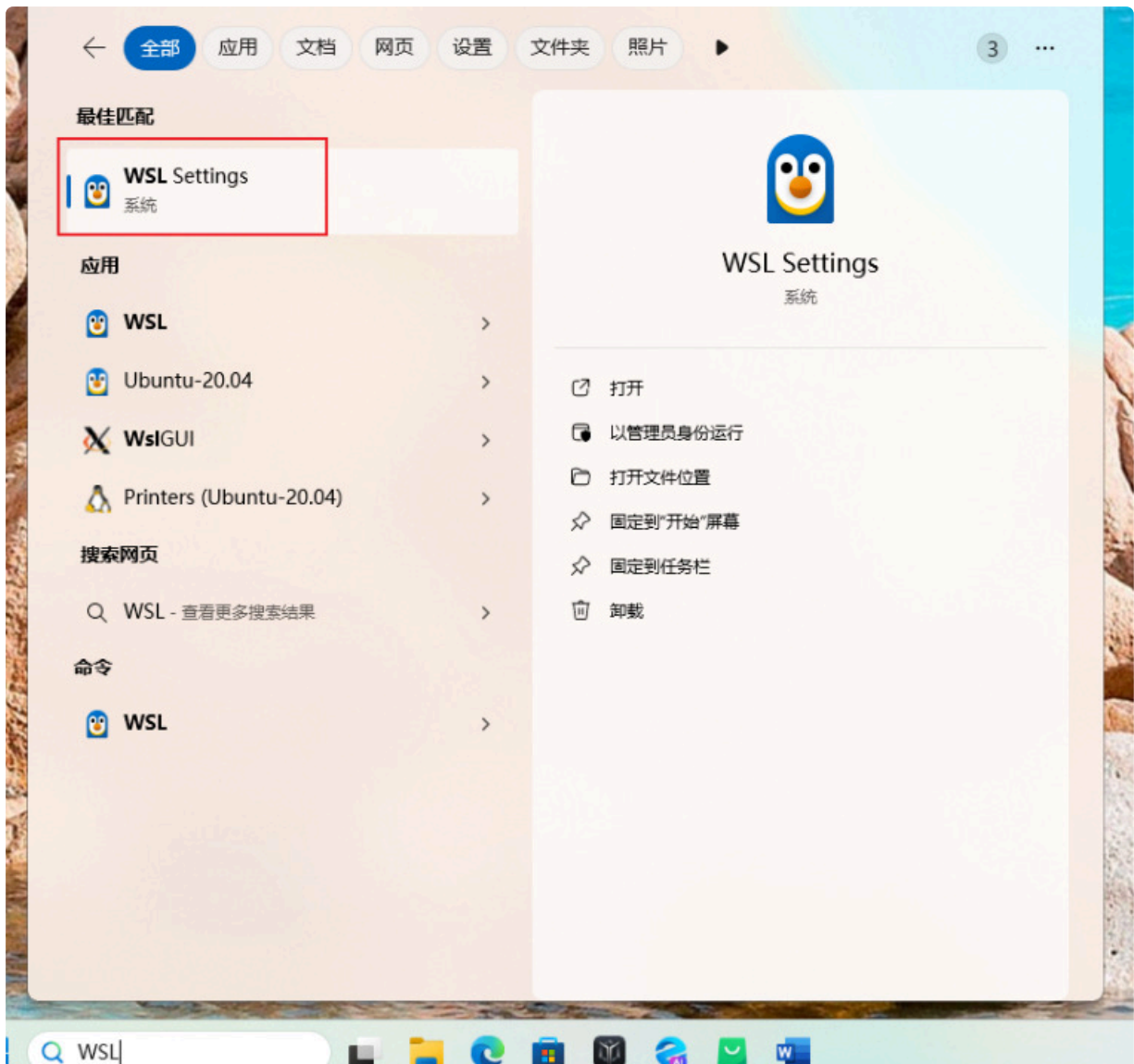
1. 打开cmd终端，首先在终端输入 `wsl --update`，确保已更新到支持 GUI 设置的最新版本（部分电脑不支持此GUI，对应修改 `.wslconfig` 即可，见下文）

[learn.microsoft.com](https://learn.microsoft.com)。

更新完成后，在终端中执行 `wsl --shutdown`` 关闭wsl。



按如下方式打开 **WSL Settings GUI** 页面，把wsl网络模式设置为mirrored，同时关闭防火墙（否则数据传输受限）



## 应用流程

- i. 在 GUI 中选择网络模式 →
- ii. Settings 自动在 `.wslconfig`（全局，应用到所有WSL发行版）中写入相应 `networkingMode` 值 →
- iii. 重启 WSL（`wsl --shutdown`）使新设置生效 [github.com](https://github.com)。

## networkingMode

- `Nat`：WSL2 使用 Hyper-V NAT 桥接（Windows 与 WSL2 网络隔离）
- `None`：禁用 WSL2 网络
- `Mirrored`：镜像模式，将 Windows 主机的所有网络接口映射进 WSL2  
[github.com](https://github.com) [learn.microsoft.com](https://learn.microsoft.com)
- `VirtioProxy`：通过 virtio-proxy 加速跨系统通信

## 2. 修改网络配置

ROS 网络依赖 IP 地址映射，正确配置后各节点（MAVROS、激光里程计、规划器等）才能互联。

在cmd中获取wsl2在windows主机端的IP（注意不要记成防火墙的ip）

```
(base) PS C:\Users\31636> ipconfig

Windows IP 配置

以太网适配器 vEthernet (Default Switch):

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::255f:b6a6:e008:434a%23
    IPv4 地址 . . . . . : 172.21.64.1
    子网掩码 . . . . . : 255.255.240.0
    默认网关 . . . . . :

以太网适配器 vEthernet (WSL):

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::c22:9443:a234:d535%51
    IPv4 地址 . . . . . : 172.28.112.1
    子网掩码 . . . . . : 255.255.240.0
    默认网关 . . . . . :

以太网适配器 以太网:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::5a91:785a:f6fa:a472%26
    IPv4 地址 . . . . . : 192.168.31.251
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.31.1
```

在WSL终端中获取虚拟机IP

```

root@h:/mnt/e/hfm/multimodel/bt/match# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
ether fe:b2:9d:30:91:59 txqueuelen 0 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.28.112.232 netmask 255.255.240.0 broadcast 172.28.127.255
inet6 fe80::215:5dff:fec6:f440 prefixlen 64 scopeid 0x20<link>
ether 00:15:5d:c6:f4:40 txqueuelen 1000 (Ethernet)
RX packets 378628 bytes 548582672 (548.5 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 118782 bytes 9639085 (9.6 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

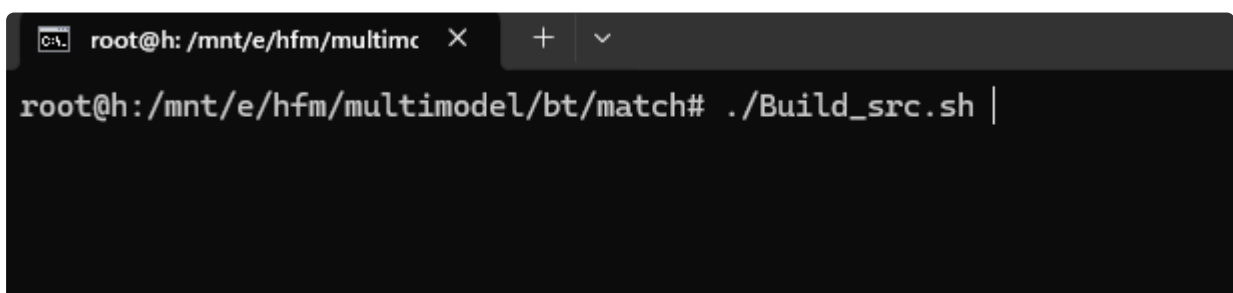
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 2427 bytes 367275 (367.2 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2427 bytes 367275 (367.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- 编辑 `match/sensor.sh` ，将 MAVROS 的 IP 修改为 Windows 主机端 IP；
- 在 `catkin_match/src/sensor_pkg/main.py` 中同步修改为 Windows 主机端 IP；
- 编辑 `match/DroneSITL.bat` ，将 IP 修改为 WSL2 端 IP。

## 2. 编译仿真源码

双击 `WinWSL.bat` 打开 WSL2 终端，执行 `./Build_src.sh`



```

root@h:/mnt/e/hfm/multimodel/bt/match# ./Build_src.sh |

```

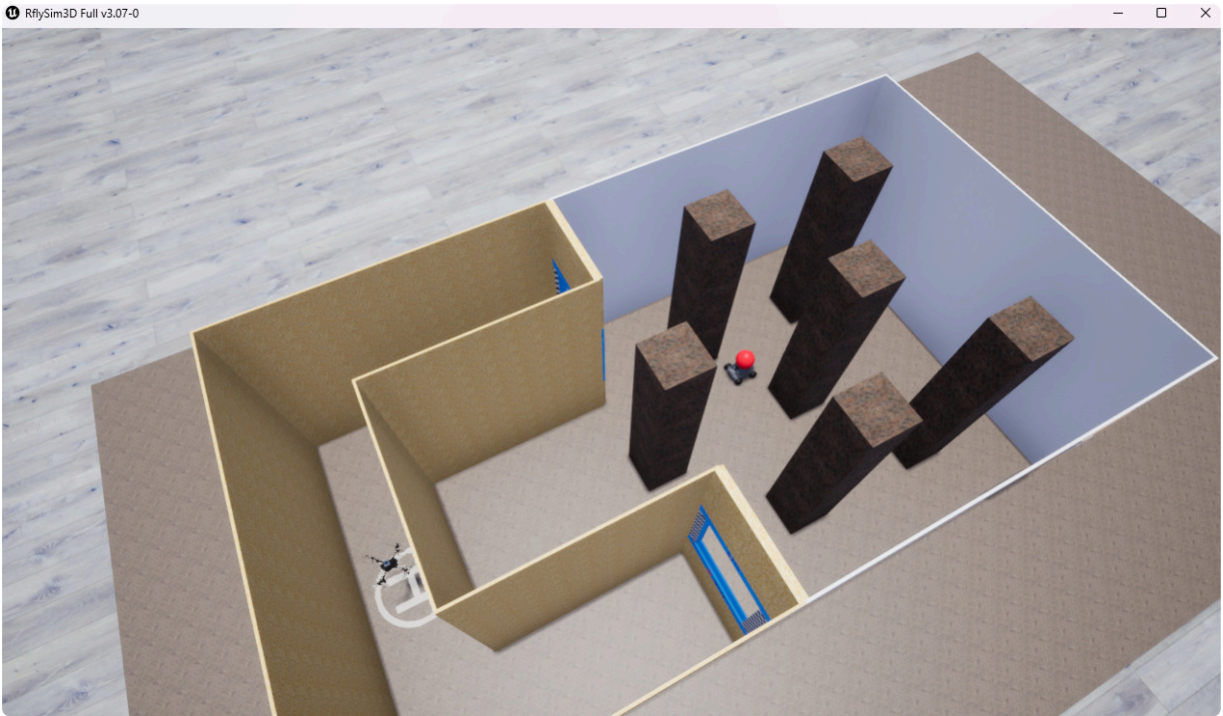
将后续用到的 ROS 包、PX4 插件及自定义节点编译成可执行文件及库。

## 5.2 仿真运行与测试

### 1. 启动仿真底层

- 在比赛目录下双击 `DroneSITL.bat` ，等待界面提示 `Enter Altitude Mode` (图1) 并看到无人机模型同时动态加载障碍物、起飞标志、红色气球等元素 (图2)。

```
Copy to Clipboard: Accepted MAVLink from PX4 SITL
PX4: Init MAVLink
PX4: Found firmware version: 1.12.3dev
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 512 ACCEPTED
PX4: Command ID: 512 DENIED
PX4: Command ID: 512 ACCEPTED
PX4: [logger] ./log/2025-04-17/02_57_45.ulg
PX4: Failsafe mode activated
PX4: Enter Altitude Mode!
PX4: Command ID: 512 DENIED
PX4: Command ID: 512 DENIED
PX4: Vision PARAM Send.
PX4: Command ID: 512 DENIED
```



## 2. 启动感知与规划模块

- 双击 `WinWSL.bat` ，执行：

```
1 | ./sensor\.sh
```

依次启动：

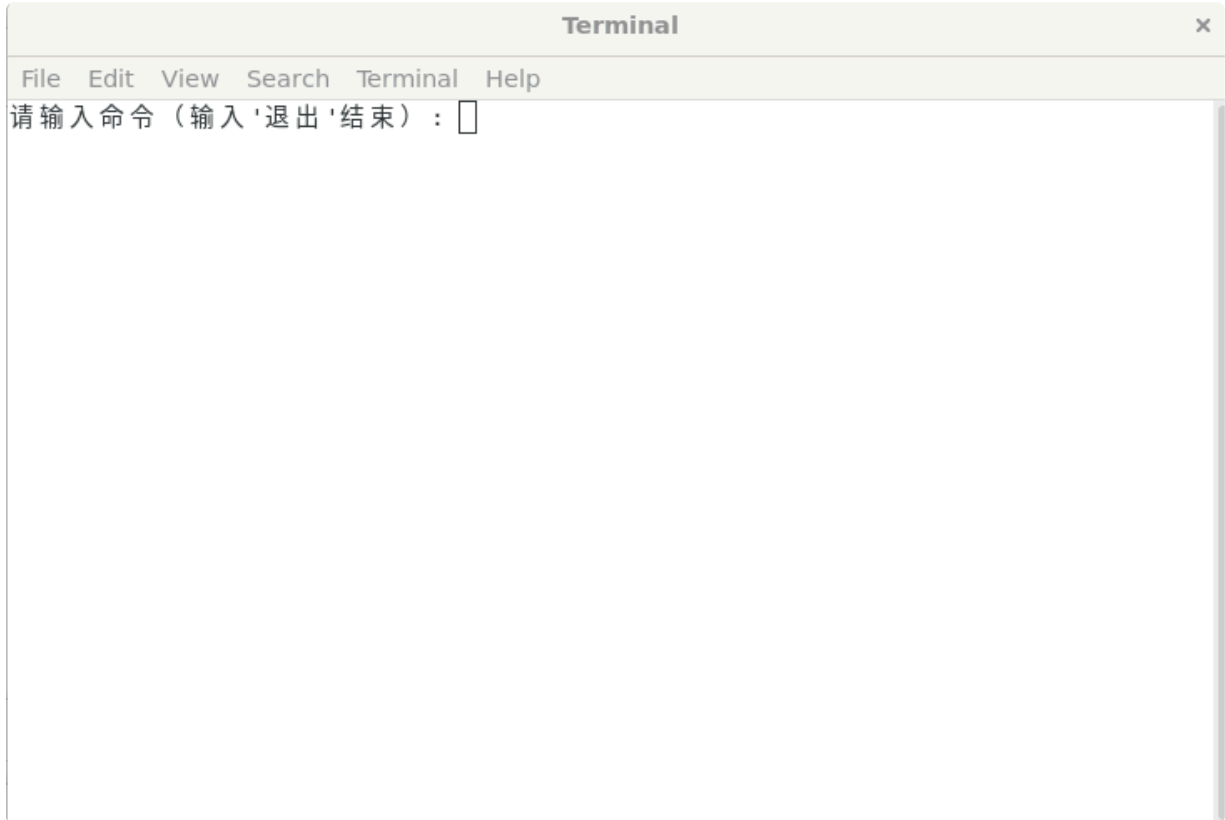
- MAVROS (ROS↔MAVLink 桥接)
- 激光里程计模块 (Faster-LIO)
- 路径规划器 (Ego-Planner)
- 目标检测 (YOLOv5)

各功能模块订阅/发布 ROS 话题，共同构成无人机的感知—决策—控制闭环：

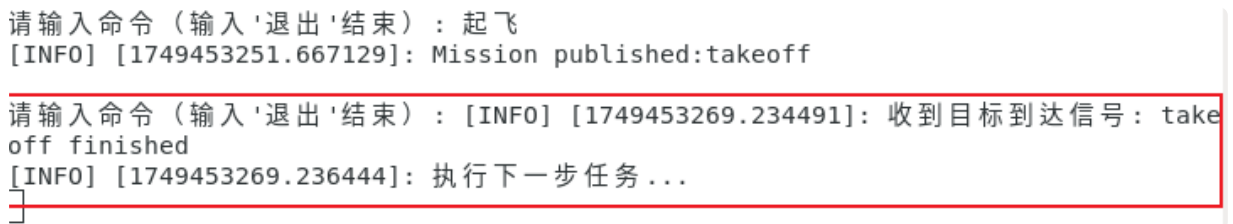
- **Faster-LIO** 提供高频里程计和地图更新
- **Ego-Planner** 基于当前环境和目标动态生成避障路径
- **YOLOv5** 检测红色气球等物体位置，指导飞行任务
- **MAVROS** 将 ROS 的控制命令转换为 MAVLink 并发送给 PX4 SITL

### 3. 执行竞赛流程

- 在 Ollama 的大模型交互窗口输入自然语言指令（如“穿越红色气球”），模型返回对应的行为树或控制指令；复制到终端后，MAVROS 将其下发给 PX4。
- 待所有程序加载完毕后，找到大模型窗口，如下：



可复制命令进去控制无人机运动，完成整个飞行流程。



当看到这个表示已完成相应任务，即可复制指令输入进行下一步任务。对于穿框和到达最终目标点两个任务，可能会出现第一次输入没反应的情况，再次输入相应命令，在命令发布后看到rviz里面显示运动轨迹即可。

```
Terminal
File Edit View Search Terminal Help
请输入命令 (输入 '退出' 结束) : 起飞
[INFO] [1749453251.667129]: Mission published:takeoff

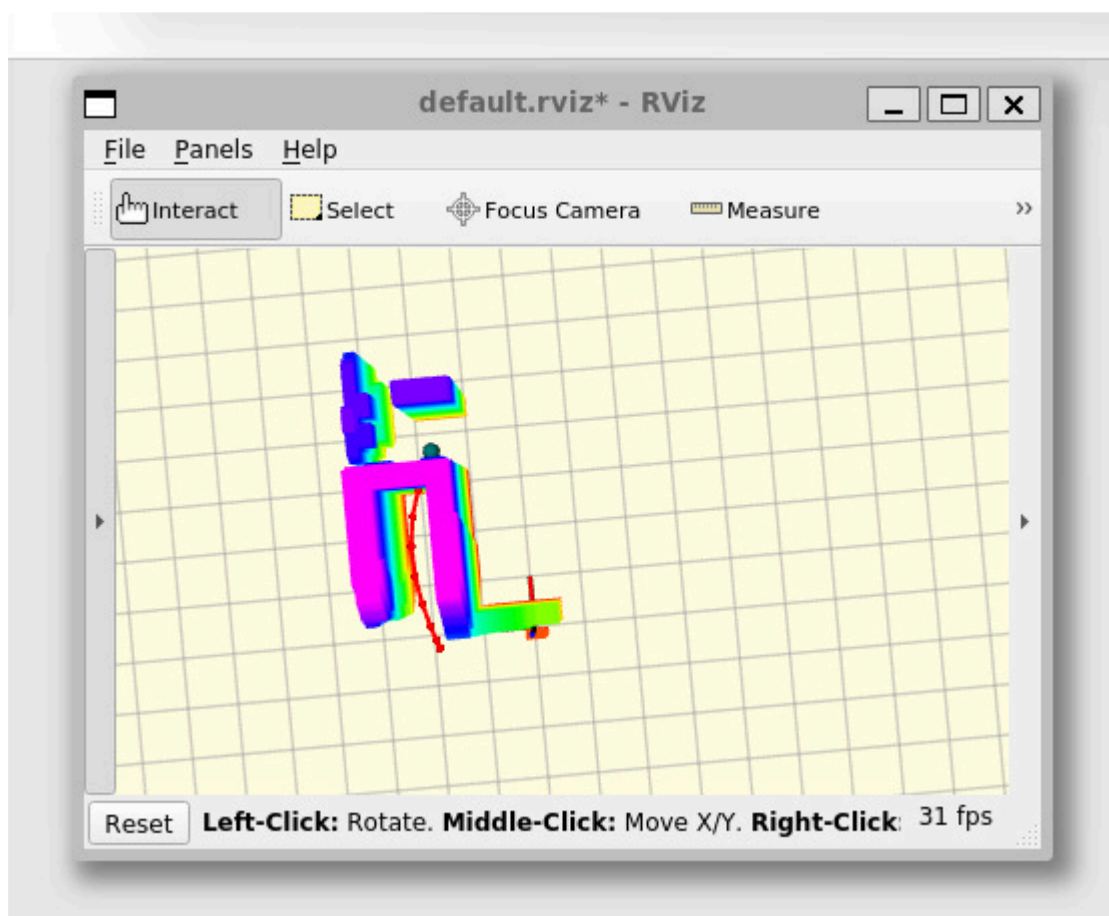
请输入命令 (输入 '退出' 结束) : [INFO] [1749453269.234491]: 收到目标到达信号: take
off finished
[INFO] [1749453269.236444]: 执行下一步任务...
到达穿框目标点
[INFO] [1749453443.838160]: Mission published:plan

请输入命令 (输入 '退出' 结束) : [INFO] [1749453451.642524]: 收到目标到达信号: targ
et point (0.00, 1.80, 1.20) has arrived
[INFO] [1749453451.645239]: 执行下一步任务...
穿框
[INFO] [1749453510.966424]: Mission published:cross_frame

请输入命令 (输入 '退出' 结束) : 穿框
[INFO] [1749453527.826032]: Mission published:cross_frame

请输入命令 (输入 '退出' 结束) : [INFO] [1749453543.360407]: 收到目标到达信号: targ
et point (3.50, 1.50, 1.20) has arrived
[INFO] [1749453543.364149]: 执行下一步任务...

```



# 7. 常见问题

## Q1. 激活虚拟机时卡住

```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

部署映像服务和管理工具
版本: 10.0.26100.1150

映像版本: 10.0.26100.4349
```

A1. 通常是因为后台有挂起的更新或待重启操作。

打开任务管理器，查看 `dism.exe` 或 `Tiworker.exe` (Windows 模块安装工作进程) 是否在占用 CPU/磁盘。

## Q2. 安装Ollama时进度缓慢

A2. 这是因为WSL2网络模式设为NAT与Windows隔离后网络资源受限，此时可以将网络模式更改为Mirrored，等待下载完成后再改回NAT

## Q3. 执行程序时提示DISPLAY相关错误

```
root@RflySim99:/mnt/e/git/8.RflySimVision/3.CustExps/e10.LLM-BehaviorTreeUAVCtrl# ./sensor.sh
Option "-x" is deprecated and might be removed in a later version of gnome-terminal.
Use "--" to terminate the options and put the command line to execute after it.
Unable to init server: Could not connect: Connection refused
Failed to parse arguments: Cannot open display:
Option "-x" is deprecated and might be removed in a later version of gnome-terminal.
Use "--" to terminate the options and put the command line to execute after it.
```

A3. 这是因为WSL2中的GUI程序无法找到X11显示服务器导致的。WSL2默认不支持GUI应用程序显示，需要配置X服务器来解决此问题。

### 解决方案：

#### 1. 启动VcXsrv X服务器

在Windows主机上启动VcXsrv（已包含在RflySim工具链中）：

```
1 | cd /d %PSP_PATH%\VcXsrv
2 | xlaunch.exe -run config1.xlaunch
```

#### 2. 设置DISPLAY环境变量

方法一（推荐）：使用Windows主机IP

```
1 | export DISPLAY=winip:0.0
```

其中 `winip` 是Windows主机的IP地址（如192.168.31.241）

## 方法二：动态获取IP

```
1 | export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):0.0
```

为了永久生效，可以将命令添加到 `~/.bashrc` 中：

```
1 | echo 'export DISPLAY=winip:0.0' >> ~/.bashrc
2 | source ~/.bashrc
```

## 3. 验证X11转发

测试GUI应用是否能正常显示：

```
1 | sudo apt-get install x11-apps
2 | xeyes
```

如果看到眼睛图标程序，说明X11转发配置成功。

## 4. 防火墙设置

确保Windows防火墙允许VcXsrv通过，或者临时关闭防火墙进行测试。

### 注意事项：

- 每次重启WSL2后需要重新设置DISPLAY变量（除非写入.bashrc）
- 确保VcXsrv在启动ROS节点之前已经运行
- 如果仍有问题，可以尝试使用固定IP地址替代动态获取的方式