

| RflySim AI Skill 安装与使用实验

| 1. 实验目的

本实验围绕 RflySim 配套建设的 5 个 AI Skill 展开，目标是完成 Skill 的安装部署、目录检查、环境验证和典型使用展示，并理解这些 Skill 在 RflySim 开发与使用过程中的辅助作用。

通过本实验，读者应能够：

- 理解 `rfly-api-guide`、`rfly-exp-nav`、`rfly-fc-dev`、`rfly-sim-usage`、`rfly-vis-scene` 五个 Skill 的用途与边界。
- 掌握 Skill 的标准目录结构，包括 `SKILL.md`、`scripts/` 和 `references/`。
- 掌握基于 `scripts/runner.py` 的本地执行与环境检测方式。
- 了解 Skill 在实验检索、SDK 接口定位、飞控源码分析、仿真组件检查和三维场景识别中的应用效果。

| 2. 实验要求

- 软件要求：Windows 10 及以上版本；已安装 RflySim 工具链^[1]；已安装 Python；已具备可使用 Skill 的 AI 宿主环境。
- 环境要求：已正确安装 RflySim 工具链。

| 3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e15.RflySim_Skills](#)

本实验包含以下 5 个 Skill：

- 1 | rfly-api-guide：用于定位和解释 RflySim SDK 接口文件及方法定义
- 2 | rfly-exp-nav：用于检索 RflySim 各类实验目录并生成学习路径建议
- 3 | rfly-fc-dev：用于识别 PX4 与 ArduPilot 飞控源码环境及相关资料
- 4 | rfly-sim-usage：用于检查仿真链路组件、BAT 模板和 DLL 模型等内容
- 5 | rfly-vis-scene：用于识别 UE4、UE5、VisCreate 及三维显示相关环境

每个 Skill 目录下的主要文件含义如下：

- `SKILL.md`：Skill 的触发条件、工作流和使用规则说明。
- `scripts/runner.py`：Skill 的本地执行脚本，用于环境检测、搜索、索引刷新等确定性任务。
- `references/`：Skill 的参考文档或索引缓存文件，用于按需加载。

4. 实验内容或步骤

4.1 Skill 安装与目录部署

<https://lingma.aliyun.com/download>

本次实验将以阿里的AI IDE——通义灵码来进行演示，可通过'

<https://lingma.aliyun.com/download>'来下载安装。安装完成后直接登录自己的账号。打开后导航到当前实验文件夹下，在右侧侧边栏直接输入`帮我把当前文件夹中的5个SKILL全局安装一下。`。即可实现全局安装。对于Trae安装完成的Skill存放在`C:\Users\`

<用户名>.lingma\skills`中。部署完成后，检查每个 Skill 是否具备如下结构：

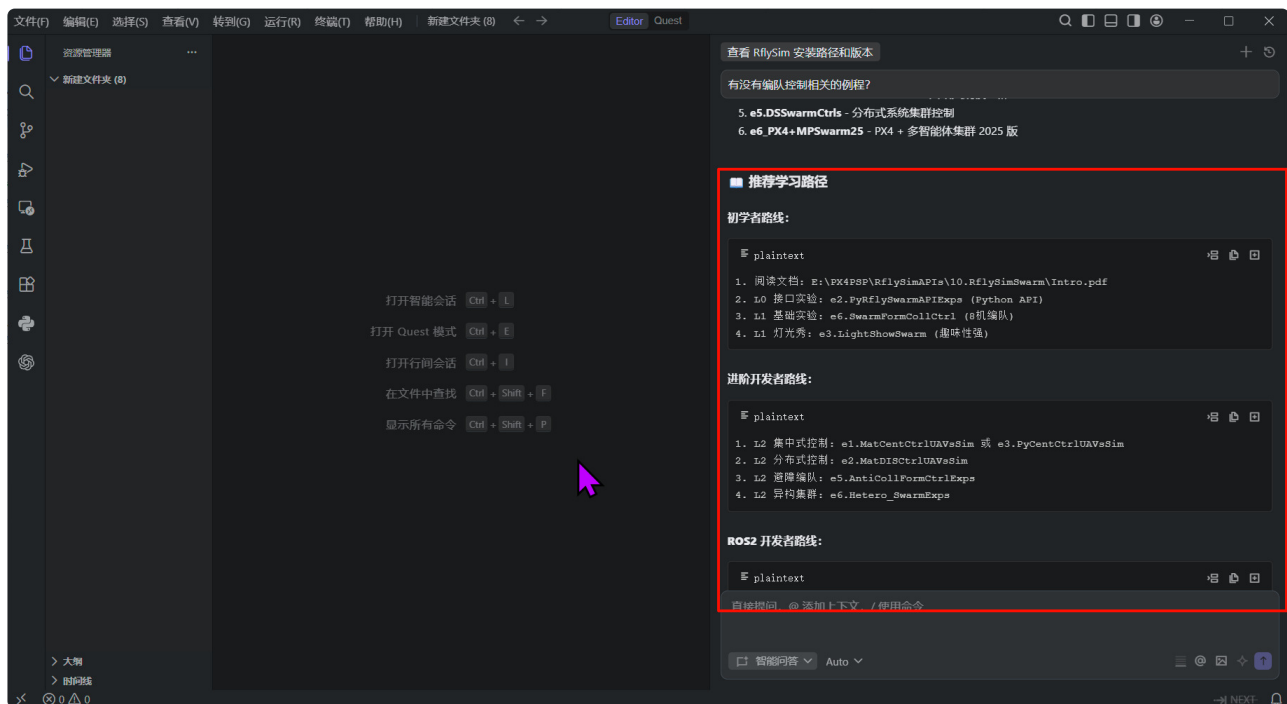
```
1 | <skill-name>/
2 | └─ SKILL.md
3 | └─ scripts/
4 |   └─ runner.py
5 | └─ references/
```

若某个 Skill 缺少 `scripts/runner.py`，则该 Skill 无法完成本地执行验证；若缺少 `references/`，则 Skill 将无法加载缓存索引或细化参考资料。

另外，也可以进入通义灵码的设置中查看已经加载的Skill。

4.3.2 实验二：例程检索与学习路径推荐

当提问“有没有编队控制相关的例程？”，Trae中的AI助手回答如下：



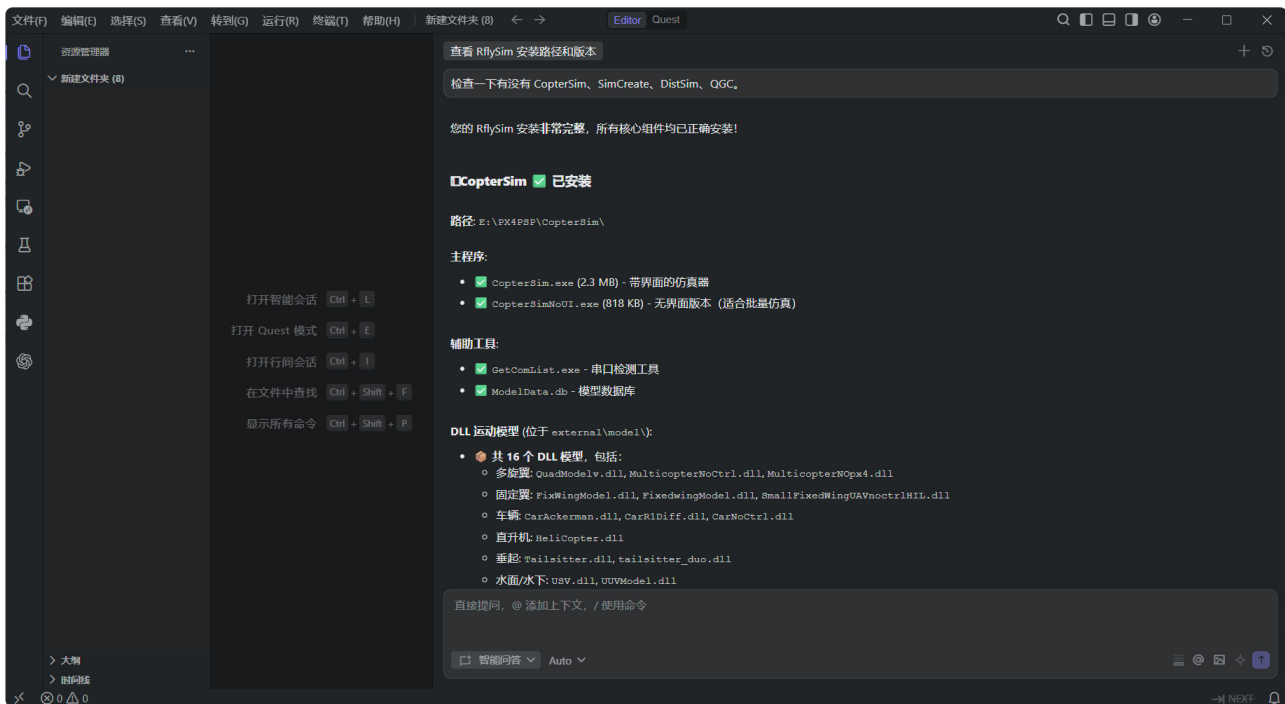
4.3.3 实验三：SDK 接口定位与能力说明

当提问“帮我查一下 RflySim 中 `send_pos_ned` 是在哪定义的，它的作用是什么？”，Trae中的AI助手回答如下：



4.3.4 实验四：仿真组件检查与流程理解

当提问“检查一下有没有 CopterSim、SimCreate、DistSim、QGC。”，Trae中的AI助手回答如下：



4.3.5 实验五：飞控源码环境识别

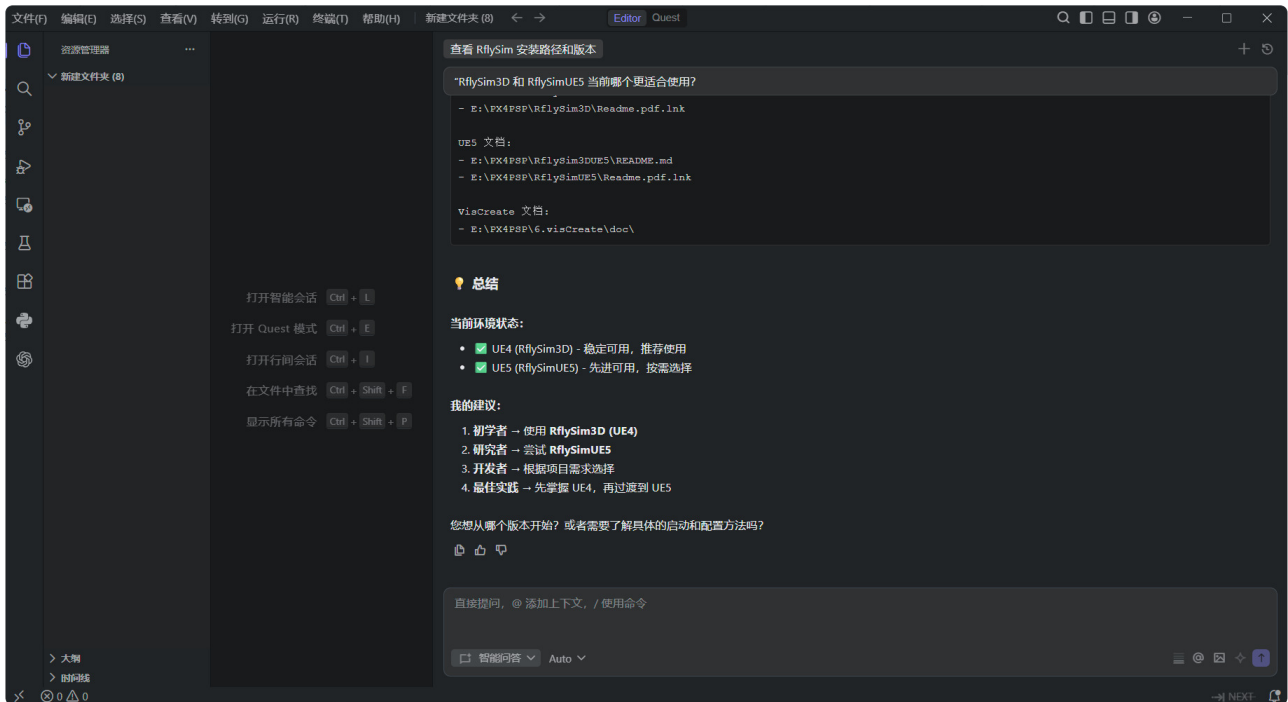
当提问“当前环境里 PX4 和 ArduPilot 都装了吗？”，Trae中的AI助手回答如下：



ArduPilot源码为 v4.13 版本才具备的。

4.3.6 实验六：三维场景与传感器环境识别

当提问“RflySim3D 和 RflySimUE5 当前哪个更适合使用？”，Trae中的AI助手回答如下：

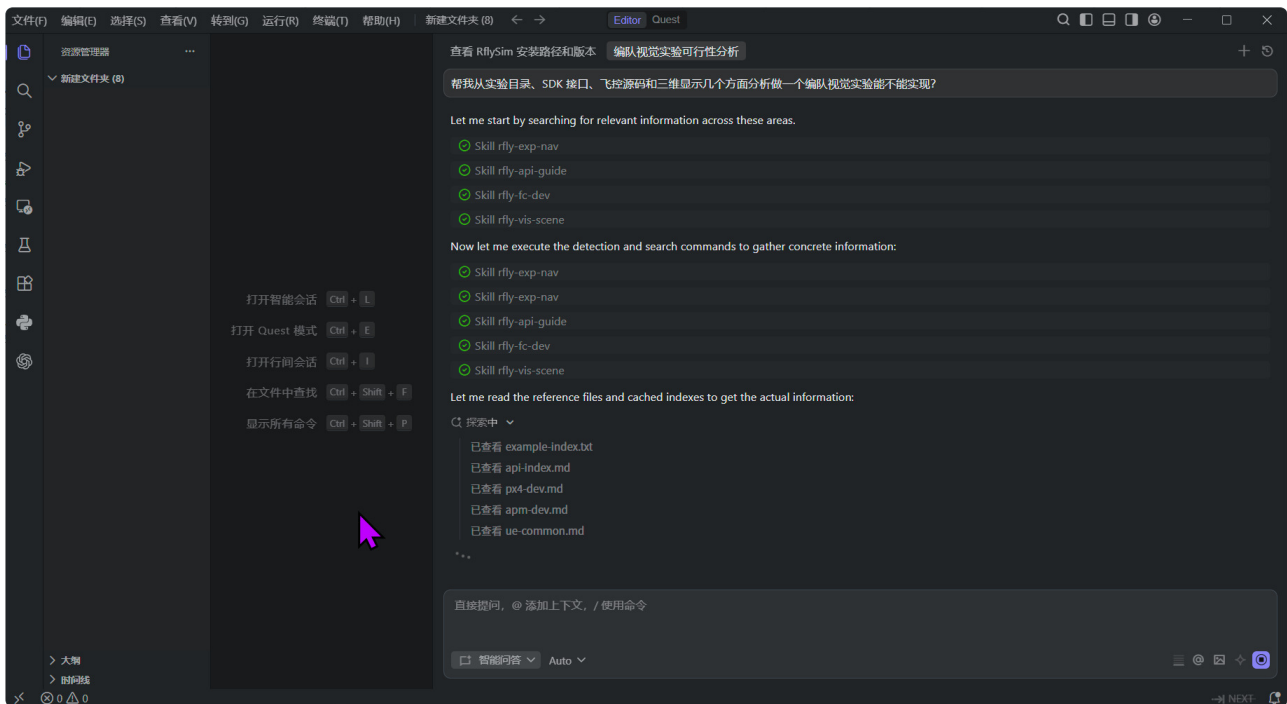


4.3.7 实验七：多 Skill 协同分析

可提问：“帮我从实验目录、SDK 接口、飞控源码和三维显示几个方面分析做一个编队视觉实验能不能实现？”。可能涉及的 Skill：

- rfly-exp-nav
- rfly-api-guide
- rfly-sim-usage
- rfly-fc-dev
- rfly-vis-scene

预期效果：





5. 关键知识点

本实验涉及的关键知识点如下：

- Skill 的标准目录结构：`SKILL.md`、`scripts/`、`references/`。
- `SKILL.md` 中 `description` 对 Skill 触发的重要作用。
- `scripts/runner.py` 作为本地确定性执行层的意义。
- `references/` 作为缓存索引和参考资料目录的作用。
- 不同 Skill 的职责边界：
 - `rfly-exp-nav` 负责实验导航。
 - `rfly-api-guide` 负责 SDK 接口定位。
 - `rfly-fc-dev` 负责飞控源码环境识别。
 - `rfly-sim-usage` 负责仿真链路组件检查。
 - `rfly-vis-scene` 负责三维场景与可视化环境识别。
- 多 Skill 协同可以形成“找实验、查接口、查环境、查源码、查显示”的完整辅助链路。

6. 参考资料

1. [RflySim 官方文档](#)

7. 常见问题

Q1: 多 Skill 协同使用的意义是什么?

A1: 单个 Skill 往往只解决某一类问题，例如实验检索、接口定位或环境检测。多 Skill 协同可以把实验导航、接口阅读、环境检查和源码分析串联起来，形成更完整的辅助能力链。

1. <https://rflysim.com/> ↩