三维场景交互接口RflySim3D碰撞检测模式对比实验（仅限完整版及以上版本）

1. 实验目的

了解RflySim3D不同碰撞模式的数据回传原理，对比不同碰撞检测模式的性能损耗。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链[1]。
- 硬件要求：笔记本/台式电脑1台[2]。

3.实验地址

例程目录：
[安装目录]\RflySimAPIs\3.RflySim3DUE\3.CustExps\e0_AdvApiExps\e3_CollisionMode

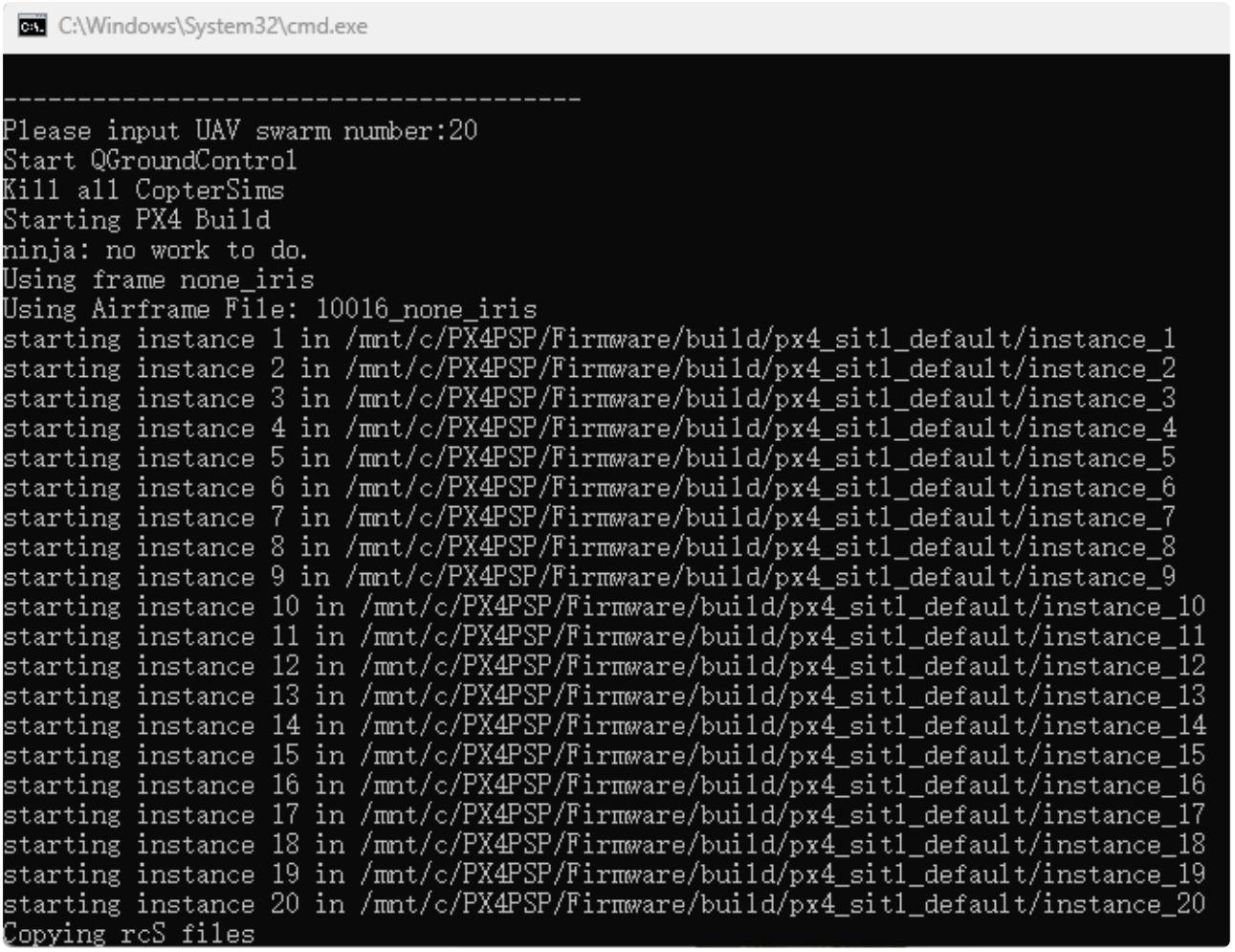
- Python38Run.bat：Python环境启动脚本。
- CollisionDemoSITL.bat：软件在环仿真启动脚本。
- CollisionDemo.py：例程功能主程序。

4. 实验内容或步骤

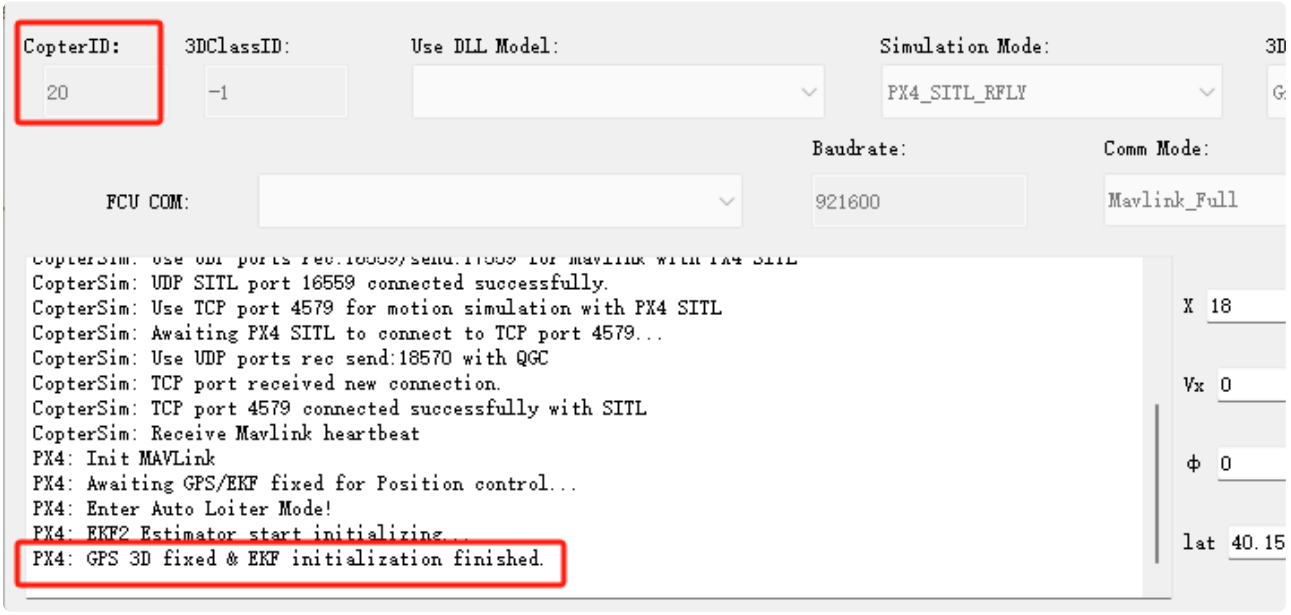
4.1 碰撞模式性能消耗统计实验（必做）

步骤1：启动软件在环仿真

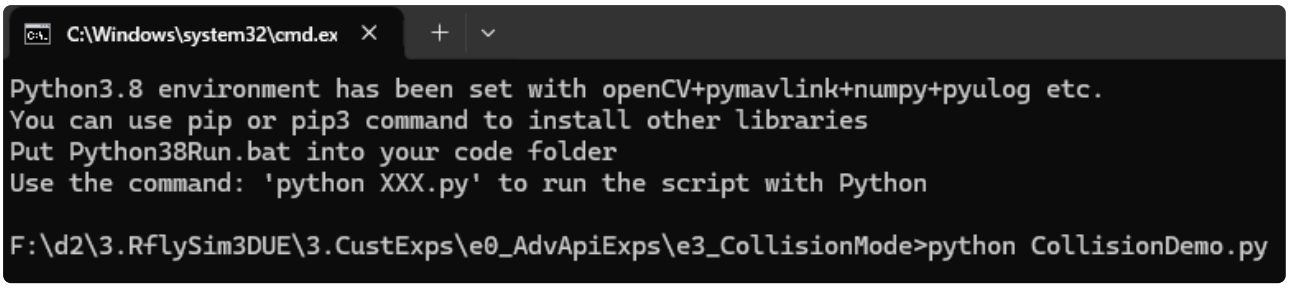
双击运行CollisionDemoSITL.bat，在终端输入所需飞机数量启动软件在环仿真。



等待CopterSim初始化完成



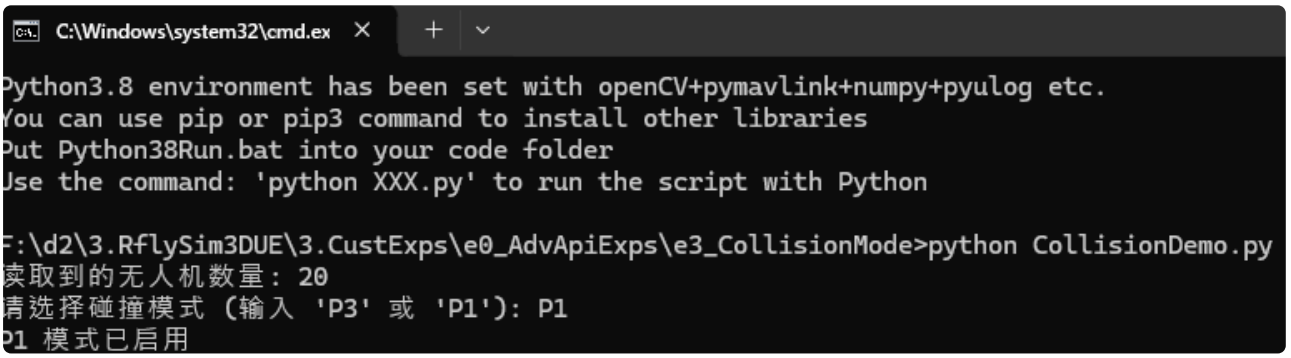
步骤2：运行python文件



在文件夹下，双击Python38Run.bat，打开集成好的python环境，输入 python CollisionDemo.py，回车运行。

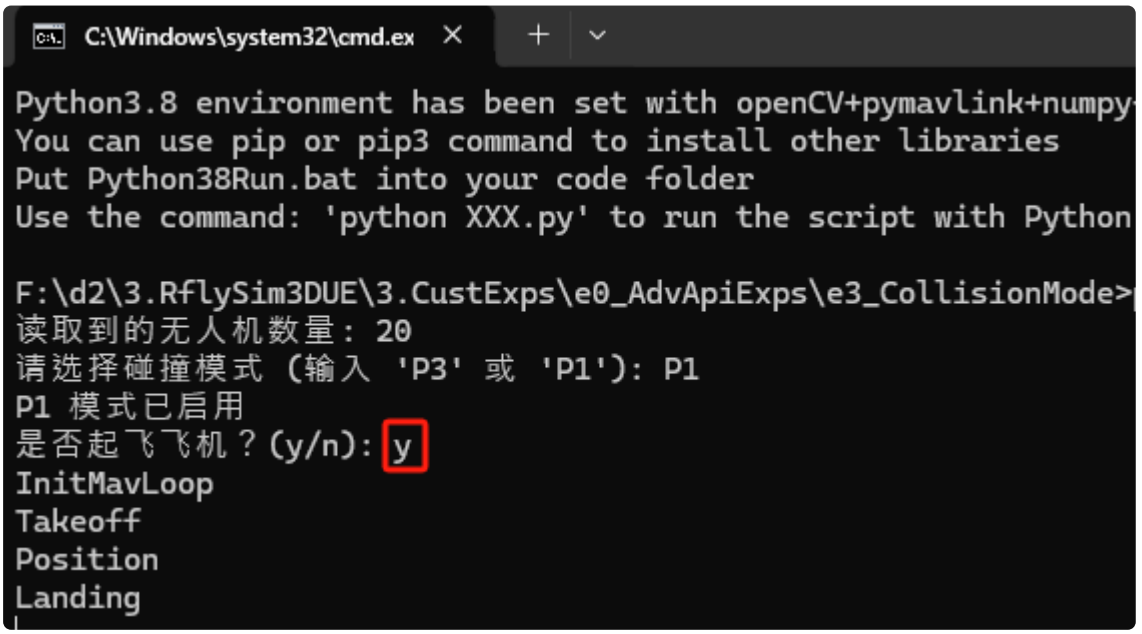
步骤3：选择启用碰撞模式

根据步骤2启用的终端提示选择对应的碰撞模式



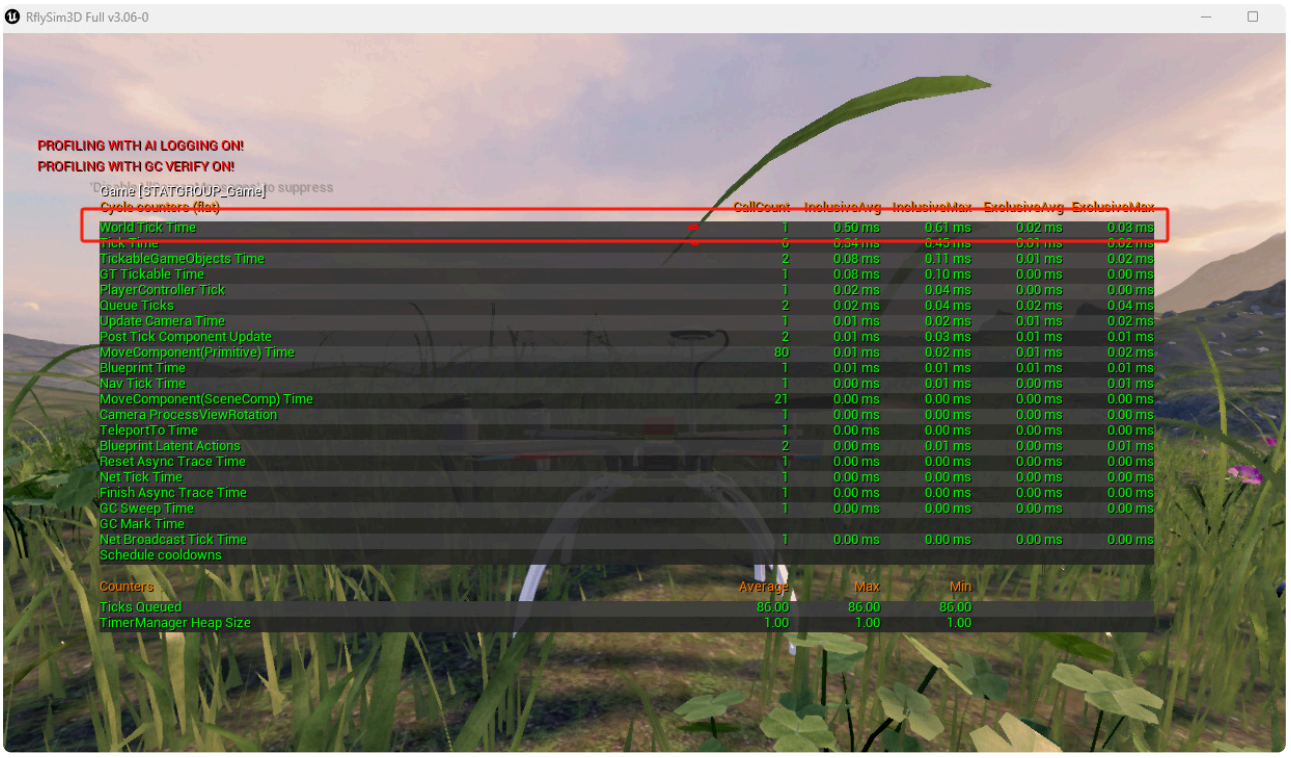
步骤4：选择是否起飞飞机

在上一步选择碰撞模式后，在同样的终端选择是否起飞飞机

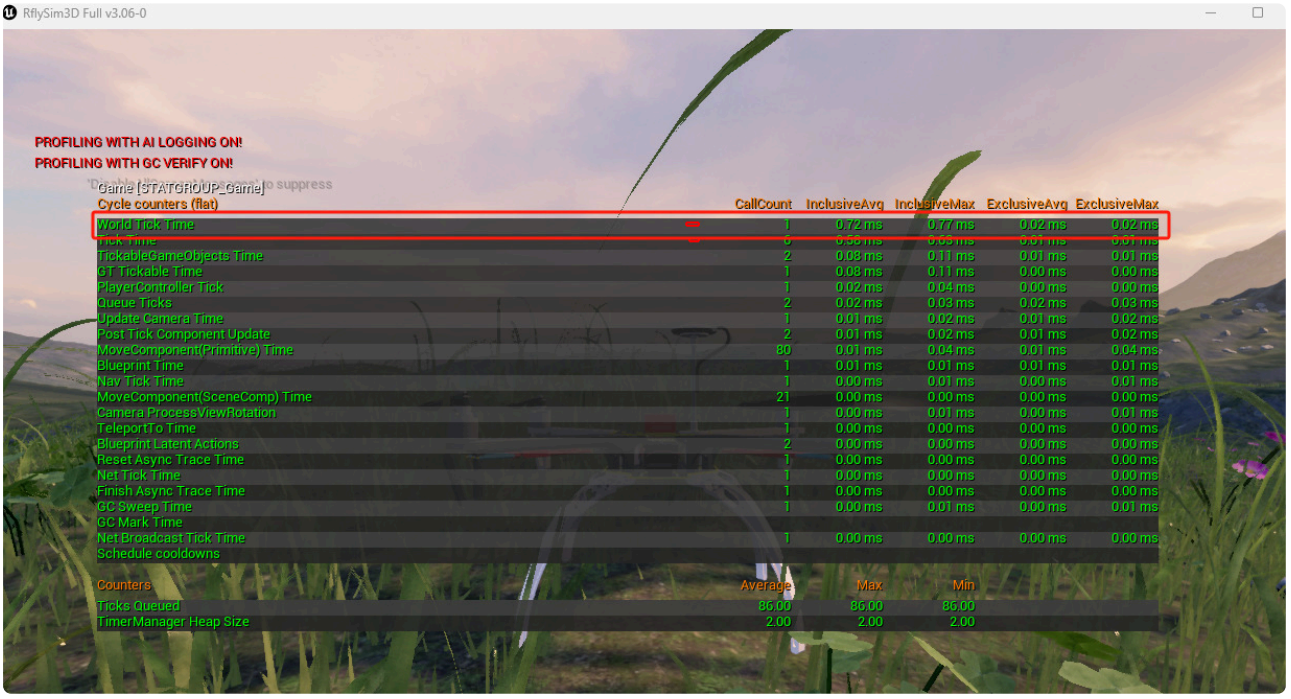


步骤5：对比不同碰撞模式性能消耗

启用碰撞模式前的性能消耗



启用P1碰撞模式起飞前的性能消耗



启用P3碰撞模式起飞前的性能消耗

4.2 Vscode调试运行实验（选做）

准备工作：

- 先确保已经按RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig\Readme.pdf步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，在运行python文件时，可使用VS Code（或Pycharm等工具）来打开python文件文件，并阅读代码，修改代码，调试执行等。

扩展实验：

- 请自行使用VS Code阅读例程中的python源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。
- 请尝试修改代码，创建更多的碰撞和输出其他碰撞信息。

4.3 实验效果

通过性能消耗对比得出，P3碰撞模式对比P1碰撞模式有较明显的性能改善，更能适应大规模集群仿真。

5. 关键知识点

关键知识点1：碰撞模式原理及性能消耗验证思路

RlfySim3D在开启碰撞检测后，会以一定频率向各个方向发送射线检测物体和飞行器的距离，当距离小于一定阈值

时，认为飞行器与该物体发生碰撞。碰撞模式CurMode

3，减少射线的检测频率（只有在大规模集群的时候才能看出显著的效果），如果无人机5m范围内没有检测到其他对象会将检测频率降低为1hz，平台默认是25hz（锁帧状态）

对比不同碰撞模式性能消耗的主要验证方式是创建大量无人机初始化在地面的时候开启P3模式对比开启P1模式下性能的消耗（无人机在静止状态的性能消耗对比）。主要通过观测发包时的一个
WorldTick
耗时对比，在起飞之前飞机P3模式默认是不会发包的所以会有一个明显的比较。起飞之后因为发包条件是min
< 0.3,
也就是射线检测到对象且目标距离很近的时候才会触发，P1模式和P3模式在空中时性能消耗对比并不明显。所以另一种验证方式是通过启用大规模集群仿真验证，但这时飞机控制相对困难。

关键知识点2：CollisionDemo.py关键代码解析

初始化 UE控制并显示统计信息：

```
ue = UE4CtrlAPI.UE4CtrlAPI()

ue.sendUE4Cmd(r'stat fps')

ue.sendUE4Cmd(r'stat game')

time.sleep(5)
```

stat　stat

通过 UE4CtrlAPI 实例向 UE4 发送命令 fps 和 game，显示帧率和游戏性能统计信息，用于监控仿真性能。

切换启用碰撞模式（P1 或 P3 模式）：

```
bP3 = False

if bP3:

ue.sendUE4Cmd(r'RflyChangeViewKeyCmd P 3')

else:
```

ue.sendUE4Cmd(r'RflyChangeViewKeyCmd P 1')

time.sleep(10)

根据 bP3 的值选择 P3 或 P1 模式，控制碰撞检测的频率。P3 模式用于降低频率以优化性能。

初始化无人机控制实例：

mav = []

number = 20

for i in range(0, number):

mav.append(PX4MavCtrl.PX4MavCtrler(i + 1))

创建并存储多个 PX4MavCtrler 实例，用于控制每个无人机。

启动 MAVLink 通信循环：

for i in range(0, number):

mav[i].InitMavLoop()

启动每个无人机的 MAVLink 通信循环，使其能够接收和发送指令。

起飞和悬停控制：

for i in range(0, number):

mav[i].initOffboard()

time.sleep(0.5)

mav[i].SendPosNED(0, 0, -4)

启用 Offboard 模式，指令所有无人机起飞至 4 米高度。

定点位置控制：

for i in range(0, number):

mav[i].SendPosNED(10, 10, -12)

将每个无人机移动到坐标 (10, 10, -12)，模拟无人机定点悬停。

降落：

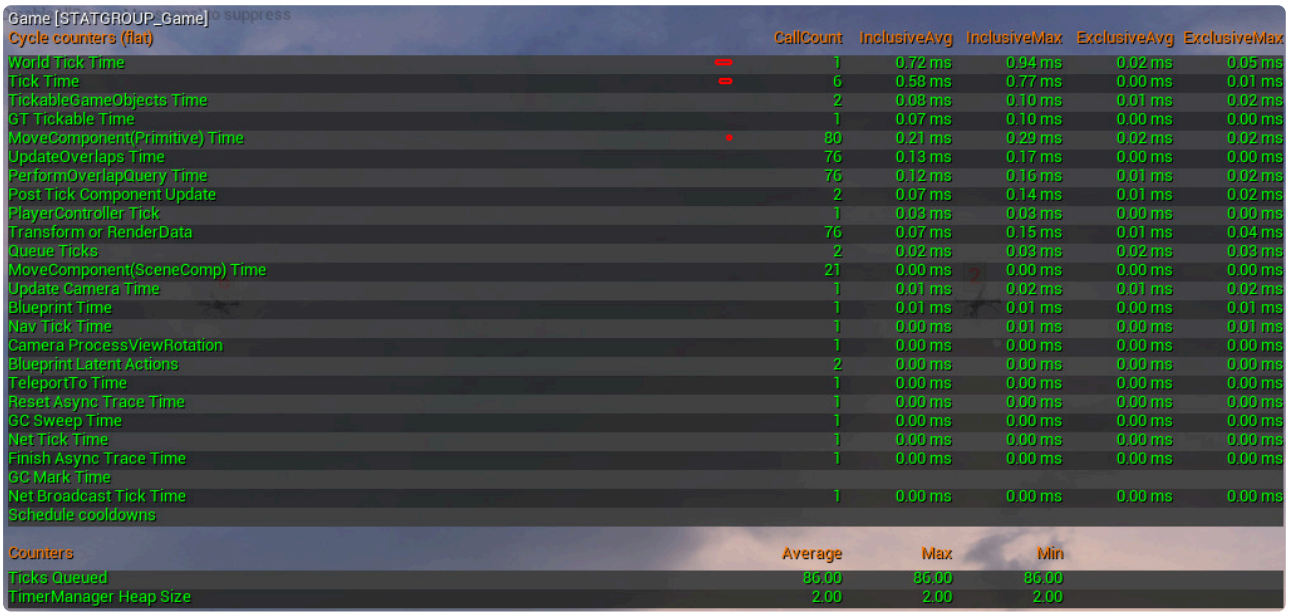
for i in range(0, number):

mav[i].sendMavLand(0, 0, 0)

指令所有无人机降落，完成仿真任务。

关键知识点3：game性能统计表解析

通过UE内置的控制台命令查看性能消耗统计表，这些参数提供了不同层次的性能信息，帮助识别性能瓶颈，确定哪些操作最消耗资源。



Game [STATGROUP_Game]

Cycle counters (flat)	CallCount	InclusiveAvg	InclusiveMax	ExclusiveAvg	ExclusiveMax
World Tick Time	1	0.72 ms	0.94 ms	0.02 ms	0.05 ms
Tick Time	6	0.58 ms	0.77 ms	0.00 ms	0.01 ms
TickableGameObjects Time	2	0.08 ms	0.10 ms	0.01 ms	0.02 ms
GT Tickable Time	1	0.07 ms	0.10 ms	0.00 ms	0.00 ms
MoveComponent(Primitive) Time	80	0.21 ms	0.29 ms	0.02 ms	0.02 ms
UpdateOverlaps Time	76	0.13 ms	0.17 ms	0.00 ms	0.00 ms
PerformOverlapQuery Time	76	0.12 ms	0.16 ms	0.01 ms	0.02 ms
Post Tick Component Update	2	0.07 ms	0.14 ms	0.01 ms	0.02 ms
PlayerController Tick	1	0.03 ms	0.03 ms	0.00 ms	0.00 ms
Transform or RenderData	76	0.07 ms	0.15 ms	0.01 ms	0.04 ms
Queue Ticks	2	0.02 ms	0.03 ms	0.02 ms	0.03 ms
MoveComponent(SceneComp) Time	21	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Update Camera Time	1	0.01 ms	0.02 ms	0.01 ms	0.02 ms
Blueprint Time	1	0.01 ms	0.01 ms	0.00 ms	0.01 ms
Nav Tick Time	1	0.00 ms	0.01 ms	0.00 ms	0.01 ms
Camera ProcessViewRotation	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Blueprint Latent Actions	2	0.00 ms	0.00 ms	0.00 ms	0.00 ms
TeleportTo Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Reset Async Trace Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
GC Sweep Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Net Tick Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Finish Async Trace Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
GC Mark Time					
Net Broadcast Tick Time	1	0.00 ms	0.00 ms	0.00 ms	0.00 ms
Schedule cooldowns					

Counters	Average	Max	Min
Ticks Queued	86.00	86.00	86.00
TimerManager Heap Size	2.00	2.00	2.00

表头参数

CallCount：各操作调用次数。表示在当前帧中每个操作或函数被调用的总次数。

InclusiveAvg：该操作或函数的平均耗时，包括所有子操作的时间。代表该操作在每次调用时的平均总耗时。

InclusiveMax：该操作或函数的最大耗时，包括所有子操作的时间。表示该操作在某一帧中经历的最高总耗时。

ExclusiveAvg：该操作的平均耗时，但不包括任何子操作的时间。用于评估该操作本身的开销。

ExclusiveMax：该操作的最大耗时，不包括子操作时间。显示该操作在某一帧中的最大自身耗时。

底部计数器（Counters）

Ticks Queued：在当前帧中排队等待执行的"Tick"任务数量。Ticks 代表游戏引擎中的更新任务，比如物体位置更新、碰撞检测等。

TimerManager Heap Size：计时器管理器的堆大小，表示当前被管理的任务或事件的数量。

各种操作含义

World Tick Time：整个世界更新所需的时间。表示所有场景元素（如物体、角色、环境等）的处理耗时，是整体性能的关键指标。

Tick Time：每帧中基本更新任务的总耗时。较低的值表示系统可以流畅地更新场景。

TickableGameObjects Time：处理所有可更新（Tickable）游戏对象的时间。它影响场景中动态对象的行为，比如无人机的移动和交互。

GT Tickable Time：渲染线程中可更新对象的时间。GT（Game Thread）表示游戏主线程，这部分用于处理与渲染和更新相关的任务。

Post Tick Component Update：在主要更新之后，对场景组件进行更新的耗时。常用于延迟任务或需要同步的任务。

PlayerController Tick：玩家控制器更新的时间。这部分用于处理用户输入和玩家控制对象的响应。

MoveComponent（SceneComp）Time 和 MoveComponent（Primitive）Time：用于更新场景和基本组件的移动，表示对场景中物体移动和碰撞检测的时间。

Update Overlaps Time 和 PerformOverlapQuery Time：用于更新物体间的重叠检测时间，帮助实现碰撞检测和物理效果。

Update Camera Time：更新摄像机位置和视图的时间。对于场景视角的调整非常重要。

Queue Ticks：用于队列中的更新任务耗时。可以是多个组件需要处理的任务时间。

Blueprint Time：执行蓝图（Blueprint）脚本的时间。蓝图是虚幻引擎中的一种可视化脚本语言，通常用于游戏逻辑。

Camera
ProcessViewRotation：处理摄像机视角旋转的时间。与玩家或摄像机的视角更新相关。

GC（Garbage Collection） 相关的时间：

GC Mark Time：标记内存中的未使用对象。

GC Sweep Time：清除未使用的对象，释放内存空间。

Net Tick Time：处理网络数据包的时间。此项在多人游戏或网络同步中尤为重要。

TimerManager Heap Size：管理定时器的堆大小，表示当前任务或事件的管理需求。

6.参考资料

[1]. [安装目录]\RflySimAPIs\3.RflySim3DUE\API.pdf

[2]. 知乎 UE4性能调试分析常用方法 - ： https://zhuanlan.zhihu.com/p/273608458

[3]. RflySim官方文档

7.常见问题

Q1：为什么P3模式比P1模式性能更好?

A1：P3模式在无人机5米范围内没有检测到其他对象时，会将检测频率降低为1Hz，而P1模式默认是25Hz（锁帧状态）。这减少了不必要的射线检测，从而降低了性能消耗，更适合大规模集群仿真。

Q2：如何判断仿真性能是否受到影响?

A2：可以通过UE4内置的性能统计命令（如stat fps和stat game）来监测性能变化。观察WorldTick Time、GT Tickable Time等指标的变化，可以直观地看到不同碰撞模式下的性能差异。

Q3：在什么情况下P1和P3模式的性能差异最明显？

A3：在大量无人机静止在地面上时，P3模式的性能优势最为明显，因为它会降低检测频率。起飞后由于碰撞检测机制的触发条件，两者的性能差异会减小。

---

1. https://rflysim.com/ ↩
2. 推荐配置请见： https://rflysim.com/doc/zh/HowToInstall.pdf ↩