本章目录

从例程目录到三项综合实践

第三章 三维环境建模与仿真

3.1 三维仿真基础与工具选型

3.2 UE场景运行渲染与交互

3.3 多模态传感器数据生成

3.4 分布式仿真架构与数据接口

3.5 场景与载具资源开发

3.6 例程体系与综合实践

3.7 习题案例分析与课程设计

本节 4 个教学单元

3.6.1 入口与选题

教材对应：第3.3.1节

3.6.2 接口目录

教材对应：第3.3.1节

3.6.3 基础与进阶目录

教材对应：第3.3.1节

3.6.4 代表案例与实践路线

教材对应：第3.3.2节

3.6.1 入口与选题：例程资料

用贴地放置例程定位入口与对照对象

- **入口：**0.ApiExps/e5_UEMapCtrl/3.TargetPlace/Readme.pdf。
- **脚本：**code/TargetPlaceDemo.py列出普通与贴地两次调用。
- **对照对象：**MountainTerrain、人物样式30、实例ID100和101。

记录：保存输入坐标、运行命令与两人物脚底位置；先确认各自x、y，不能把跨点高度差直接归因于接口。

入口、脚本、两种调用和落点记录组成一条复现链。

3.6.1 入口与选题：目录与学习任务

三层目录对应三种学习起点

- **0.ApiExps：**学习一个操作、一个命令或一种回传结构。
- **1.BasicExps：**用现成输入形成可创建、可保存的资源与场景。
- **2.AdvExps：**增加资源处理、蓝图或地理数据依赖；3.CustExps按条件选做。

选题示例：先在API层运行贴地例程，比较人物落点；再在基础层保存布局并重载。每一步都保留输入和对应结果。

按问题选择例程，依赖决定实践范围。

3.6.1 入口与选题：Readme读法

按3.TargetPlace的Readme走完输入与观察

- **输入：** MountainTerrain、人物样式30与Click Point读到的坐标。
- **准备：** ue4.bat开三维端；按M切山地，双击地面读z。
- **执行：** 按Readme编辑TargetPlace.py，再用Python38Run.bat运行。

产物： 分别核ID100的给定z和ID101的贴地落点；记录地图、坐标、命令与画面，不把不同x处高度差当作接口效果。

记录对象ID、坐标和脚底结果，再判断放置方式。

3.6.1 入口与选题：接口术语

用贴地调用区分命令、样式与实例

- **命令：** sendUE4Cmd切MountainTerrain，待地图加载后再放置。
- **API：** sendUE4Pos给定z；sendUE4Pos2Ground请求贴地。
- **样式/实例：** vehicleType=30选人物；ID100与101定位两个人物。

消息判据： 核调用中的ID、坐标与三维端对象一一对应；贴地标记由三维端处理，不能把普通调用误写成自动查地形。

命令切图，样式选人物，实例ID选对象，API定落点。

3.6.1 入口与选题：操作入口

操作、快捷键与命令入口

0.ApiExps

e0_DevToolsUsage	工具职责、输入格式与资源制作入口。
e0_RflySim3DUsage	场景程序的启动、显示与基本操作。
e1_KeyboardAPI	用快捷键改变观察、选择和场景对象。
e2_CommandAPI	用控制台或BAT发送场景命令。

人工操作便于探索，脚本调用使场景变化可重复验证。

3.6.1 入口与选题：初始化与对象

初始化配置与对象运动

0.ApiExps

e3_InitAPI	TXT在启动或切图时执行； 编辑后须重新触发对应加载。
e4_UAVCtrl	运行中向已有ID发输入， 更新部件动作或整对象依附。

验证：先重载地图确认初始状态，再向同一ID更新控制量，比较对象动作。

启动命令设置初始条件，控制接口更新对象。

3.6.1 入口与选题：环境与数据

地图、状态、地理和效果入口

0.ApiExps

e5_UEMapCtrl	对象放置、地形数据与贴地轨迹。
e6_RflySim3DCtrlAPI	模型状态控制、显示与结构化回传。
e7_RflySim3DGIS	地理原点与Cesium场景接入。
e8_RflySim3DEffect	环境、通信连线及动作效果。

场景控制与数据回传由不同接口承担。

3.6.1 入口与选题：碰撞与场景保存

碰撞事件和场景编辑入口

0.ApiExps

e9_RflySim3DCollision	碰撞事件与回传数据。
e10_BuildCusScene	在场景界面创建并保存自定义布局。

验证：保存布局后重载同一场景，再复现接触过程；用事件时间与对象ID对应碰撞记录。

事件记录与场景布局是两种不同产物。

3.6.1 入口与选题：课堂三项任务

用三个任务贯通本节目录

- **贴地放置：** e5_UEMapCtrl/3.TargetPlace → 人物位置与脚底落点。
- **场景保存：** e0_StarterContent/5.UE_CustomImport → 同次布局JSON及关联资源。
- **数据读取：** e6_RflySim3DCtrlAPI/10.GetCamObjDemo → 状态字段与两路RGB图像。

每个任务都对应入口、参数与可观察产物。

本章目录

从例程目录到三项综合实践

第三章 三维环境建模与仿真

3.1 三维仿真基础与工具选型

3.2 UE场景运行渲染与交互

3.3 多模态传感器数据生成

3.4 分布式仿真架构与数据接口

3.5 场景与载具资源开发

3.6 例程体系与综合实践

3.7 习题案例分析与课程设计

本节 4 个教学单元

3.6.1 入口与选题
教材对应：第3.3.1节

3.6.2 接口目录
教材对应：第3.3.1节

3.6.3 基础与进阶目录
教材对应：第3.3.1节

3.6.4 代表案例与实践路线
教材对应：第3.3.2节

3.6.2 接口目录：工具与资源

工具例程的最小复现产物

0.ApiExps/e0_DevToolsUsage

1.UEBlueprintUsage	运行盒体蓝图，记录旋转节点与结果。
2.3dsMaxUsage	创建盒体并导出box.fbx，核对尺寸。
3.CesiumForUnrealUsage	保存QuickStart关卡及地理加载配置。
4.SketchUpUsage	完成房屋模型并保存house.skp。

每种工具留下明确产物，用实际文件连接制作与场景运行。

3.6.2 接口目录：外部资产获取

场景传递与外部资产的复现记录

0.ApiExps/e0_DevToolsUsage

5.TwinmotionUsage 保存house.tm；按教程版本分支导出场景。

6.EpicFabUsage 选一项资产，记录来源、许可及目标格式。

· **交付：**源文件、导出产物、引用纹理，以及对应的软件和插件版本。

原始资源、导出文件与版本记录，使外部资产能够重新导入。

3.6.2 接口目录：命令查阅与调用

命令、枚举和BAT调用

0.ApiExps/e2_CommandAPI

1.command	按参数顺序发送一条命令， 观察对应场景状态变化。
2.enumerate	先查该字段枚举，再代入命令；数字含义随字段变化。
3.BatAPI	把已验证参数写入BAT；重启后核对地图与视角。

枚举定义取值，命令和脚本负责执行。

3.6.2 接口目录：TXT触发时机

启动TXT与地图TXT

0.ApiExps/e3_InitAPI

1.TXTAllCrtlScript 启动时读取RflySim3D.txt。

2.TXTMapCrtlScript 切换指定地图时执行对应TXT。

对照：重启查看全局TXT的效果，再切到指定地图查看地图TXT；记录各次地图与显示状态。

触发时机决定预设命令何时生效。

3.6.2 接口目录：启动配置示例

RflySim3D.txt逐行设置启动条件



图：TXT预设启动场景

- **文件：**RflySim3D.txt，逐行命令。
- **地图：**RflyChangeMapbyName Changsha。
- **显示：**1280×720窗口；帧率上限25。

预设命令在启动时设置地图和显示参数。

3.6.2 接口目录：TXT实际参数

预设文件保存命令与显示参数

RflyChangeMapbyName Changsha
r.setRes 1280x720w
t.maxFPS 25
stat fps

地图：Changsha必须是当前程序可加载的资源。
显示：1280×720，w为窗口模式；25是帧率上限，stat fps显示统计。
文件：1.TXTAllCrtlScript/RflySim3D.txt，一条命令占一行。

地图名和显示参数可以作为实验初始条件。

3.6.2 接口目录：部件与对象依附

执行器控制与整对象依附

0.ApiExps/e4_UAVCtrl

1.ActuatorBinding	XML父项带动子部件，见3.5倾转例。
2.ActuatorCtrl	按模型约定分配扩展执行器输入。
3.VehicleAttachSim	用Simulink定义整对象依附。
4.VehicleAttachPy	用Python指定子对象、父对象与模式。

机内部件索引与场景对象ID分别定位目标。

3.6.2 接口目录：整对象依附

场景对象ID定义整体依附关系



图：旧双机依附演示；当前例程为人物与无人机

- **调用：** sendUE4Attatch(2,1,3)。
- **对象：** 2号依附1号；
  模式3同时依附位置和姿态。
- **入口：**
  e4_UAVCtrl/4.VehicleAttachPy。

对象依附使用实例ID，机内绑定使用执行器序号。

3.6.2 接口目录：放置与地形

对象放置与地形读取入口

0.ApiExps/e5_UEMapCtrl

1.TargetCreateKey	快捷键创建靶标，XML定义中心离地高度。
2.TargetCreatePy	按给定位置创建靶标并更新高度。
3.TargetPlace	普通位置与引擎贴地处理两种调用。
4.GetTerrainMAT	MATLAB读取高度PNG并查询地形。

贴地调用和读取高度矩阵是两条过程。

3.6.2 接口目录：贴地轨迹

单体、多体与异构轨迹

0.ApiExps/e5_UEMapCtrl

5.TrajGen	地形高程参与单模型贴地轨迹生成。
6.TrajGenMulti	同构多机共享轨迹生成机制。
7.HeterTrajGenMulti	异构模型按各自ID接收UDP状态。
8.TenCarCircleCtrl	十车圆环轨迹结合地形高度。

轨迹数据按实例ID发送，地形提供高度条件。

3.6.2 接口目录：创建与回传

模型创建与状态数据入口

0.ApiExps/e6_RflySim3DCtrlAPI

1.UECtrlPy	切图、创建模型并施加附加运动。
10.GetCamObjDemo	状态回传与图像采集分别读取。
2.LoadModelsOnBat	启动脚本指定模型与初始场景。
3.LoadModelsByTxt	RflyLoad3DFile加载模型控制命令。

创建发送状态，回传请求返回结构化数据。

3.6.2 接口目录：轨迹视口与反馈

轨迹、回传、视口和损伤显示

0.ApiExps/e6_RflySim3DCtrlAPI

4.TrajDemo	按轨迹更新模型的位置姿态。
5.UE4RecUE4APITest	请求并接收场景回传消息。
6.RflySim3DViewPortDemo	控制观察视口和显示区域。
7.DamageModel	模型损伤状态与对应可见效果。

对象状态与观察窗口可以分别控制。

3.6.2 接口目录：消息与位置

消息显示和位置读取

0.ApiExps/e6_RflySim3DCtrlAPI

8.RflySim3DMsgDispDemo	显示指定消息内容与样式。
9.RflySim3DPosGet	读取场景中的位置数据。

验证：移动同一ID的对象，核对位置回传随之更新；屏幕标签便于观察，数值记录用于计算。

屏幕文字与位置结构体分别承载信息。

3.6.2 接口目录：地理原点

Cesium与倾斜模型入口

0.ApiExps/e7_RflySim3DGIS

1.CesiumPlugin	加载地理瓦片并设置地图参考原点。
2.ObliModelMap	倾斜模型转换与地理位置配置。

应用：把源数据基准、地图原点与加载地址一起记录。

地理资源加载需要同一坐标基准。

3.6.2 接口目录：连接与车辆效果

通信连线、管道与车辆效果

0.ApiExps/e8_RflySim3DEffect

1.Comm	连线的样式和持续时间表达连接关系。
2.Tunnel	坐标序列生成分段管道几何。
3.EastBomberSU24	模型扩展输入触发可见效果。
4.RoadRollerDemo	创建车辆并发送贴地移动状态。

效果几何由坐标、样式或事件参数驱动。

3.6.2 接口目录：环境与模型动作

天气、模型动作与事件效果

0.ApiExps/e8_RflySim3DEffect

5.StandardVtol	按状态更新垂直起降翼模型。
6.Weather	天气参数改变天空和降水外观。
7.3DlasersBomb	定制资源将命中事件映射为特效。
8.RobotDog	按模型协议用扩展输入驱动关节。

模型输入含义随对应资源协议确定。

3.6.2 接口目录：RflySim3D碰撞例程

RflySim3D用射线查询检测碰撞，再把结果交给CopterSim

例程：[安装目录]\RflySimAPIs\3.RflySim3DUE\0.ApiExps\e9_RflySim3DCollision

RflySim3D 射线查询	→	距离小于阈值 碰撞信息	→	CopterSim 接收结果

检测原理：按设定频率向各方向查询距离；飞行器与物体的距离小于阈值即判为碰撞。

步骤一：双击ue4.bat打开RflySim3D，再双击Python38Run.bat，输入python collsion.py，观察地面、静态物体和飞行器的碰撞消息。

步骤二：运行CopterSimColl.bat启动两机软件在环仿真；
再运行python CopterSimColl.py，按Ctrl+鼠标滚轮放大飞机直到相碰，飞机被弹开。

版本限定：按P键开启的物理碰撞引擎仅完整版及以上可用。

3.6.2 接口目录：当前接口查阅

从例程调用追到SDK定义

- **任务入口：**安装包中的Readme.pdf说明步骤；同目录脚本和配置给出实际输入。
- **文档回指：**3.4“分布式仿真架构与数据接口”中的UE4CtrlAPI文档。
- **源码入口：**RflySimAPIs/RflySimSDK/ue/UE4CtrlAPI.py。
- **调用追踪：**TargetPlaceDemo.py → sendUE4Pos2Ground → 数据包与UDP发送。

版本区别：sendUE4PosFull明确用NED米；sendUE4PosNew的轴向与时间说明仍有冲突，配套例程先保留原值。

例程给任务过程，SDK给当前参数与返回定义。

本章目录

从例程目录到三项综合实践

第三章 三维环境建模与仿真

3.1 三维仿真基础与工具选型

3.2 UE场景运行渲染与交互

3.3 多模态传感器数据生成

3.4 分布式仿真架构与数据接口

3.5 场景与载具资源开发

3.6 例程体系与综合实践

3.7 习题案例分析与课程设计

本节 4 个教学单元

3.6.1 入口与选题
教材对应：第3.3.1节

3.6.2 接口目录
教材对应：第3.3.1节

3.6.3 基础与进阶目录
教材对应：第3.3.1节

3.6.4 代表案例与实践路线
教材对应：第3.3.2节

3.6.3 基础与进阶目录：X680资源链

从X680的FBX到可创建的三维端模型

- **输入：** VehicleModel/DroneyeeX680Body.FBX与桨叶FBX。
- **处理：** 在UE核层级、轴心和材质，再Cook为运行资产。
- **产物：** DroneyeeX680Body.uasset/.uexp及DroneyeeX680.xml。

使用： Copy.bat部署至RflySim3D/Content；MeshPath指向机身资产，创建后核对旋翼。

核FBX、Cook资产、XML路径和部署目录一一对应。

3.6.3 基础与进阶目录：基础资源总览

基础层的四组资源入口

1.BasicExps

e0_StarterContent	场景创建、Cook、原点与保存入口。
e1_CusLoadDroneyeeX680	现成FBX、Cooked资源与XML模板。
e3_RflySim3DTerrainPcd	地形栅格采样、查询与PLY输出。
e4_CusLoadObj	图片、运行时网格和XML对象导入。

先得到一份可加载资源，再改变其参数。

3.6.3 基础与进阶目录：场景版本分支

UE4与当前UE5场景流程

1.BasicExps/e0_StarterContent

1.UE4StarterContent	UE4.27场景Cook及地形文件交付。
2.UE5StarterContent	当前UE5.8使用RflyDemo基础场景。
3.MaterialContent	材质引用、Cook与显示问题定位。
4.SetSceneOrigin	区分资产枢轴、世界原点和地理参考。

相同目录主题可以对应不同的版本化流程。

3.6.3 基础与进阶目录：场景生成与保存

Python创建与界面保存衔接

1.BasicExps/e0_StarterContent

5.UE_CustomImport TargetCreate.py创建，界面保存JSON。

- **输入：** ChallengeMap与TargetCreate.py中的pos、ang列表。
- **处理：** 循环发送对象状态；在界面保存本次生成结果。
- **产物：** 布局JSON及同名关联文件；随机方框保留本次坐标。

创建脚本发对象状态，保存动作写布局文件。

3.6.3 基础与进阶目录：对象格式分支

图片、运行时网格与XML对象

1.BasicExps/e4_CusLoadObj

1.loadPng	将标定图案作为带尺寸的场景资源。
2.loadSimpleMesh	通过UMG导入本地网格并保存仓库配置。
3.loadMeshXML	FBX经UE4 Cook后用XML登记资源。

三类输入产生不同的加载配置。

3.6.3 基础与进阶目录：进阶资源工作流

工作流、场景、蓝图和地理资源

2.AdvExps

e0_DevWorkflow	资产、XML与运行加载的开发流程。
e1_TwinmotionContent	Datasmith传递场景和引用资产。
e2_BlueprintModel	Rfly事件、组件动作与特效蓝图。
e3_CusGIS	OSGB转换与地理场景配置。

每个进阶入口增加一种明确的资源依赖。

3.6.3 基础与进阶目录：点云与扫描

点云资源与实物扫描重建

2.AdvExps

e4_LidarPointClound	LAS点云导入、碰撞构建与Cook。
e5_LiDAR-UE	手机扫描物体或室内，修复后导入UE。

地形高程采样见3.5；实物扫描需要相应采集设备。

扫描采集与既有点云导入分开选择。

3.6.3 基础与进阶目录：资源产物衔接

从Cook到XML再到运行加载

2.AdvExps/e0_DevWorkflow

1.3DMap_Model_Import_Dev Cook资源、XML索引与场景加载衔接。

- **输入：**目标引擎的UE资产及材质、纹理等引用依赖。
- **处理：**Cook形成运行资源；XML登记样式与MeshPath。
- **产物：**Content资源、XML和实例创建调用，三者路径对应。

运行资源、索引配置与实例状态逐层连接。

3.6.3 基础与进阶目录：蓝图实现路线

旋翼、固定翼、动画与碰撞事件

2.AdvExps/e2_BlueprintModel

1.BlueprintBuild	输入存速度变量，Tick驱动X680旋翼。
2.BPModelLoad	按固定翼蓝图的定义配置扩展输入。
3.CusLoadBP	Tello骨骼与动画蓝图制作。
4.BombEfffect	碰撞事件触发粒子与声音效果。

从具体事件追踪输入到模型动作。

3.6.3 基础与进阶目录：扩展输入特效

扩展输入与Niagara响应

2.AdvExps/e2_BlueprintModel

5.LasersEffect

Niagara是UE的视觉特效系统，本例用于显示命中效果。

- **输入：** ActuatorInputsExt携带目标Copter ID。
- **处理：** 匹配Actor标签，LineTraceByChannel返回命中位置。
- **产物：** 该位置的Niagara效果；延时后销毁目标与自身。

扩展事件将输入映射到指定视觉效果。

3.6.3 基础与进阶目录：OSGB数据链

倾斜摄影模型的转换产物

2.AdvExps/e3_CusGIS

1.ObliModelMap OSGB转3D Tiles并配置地理原点。

- **输入：** Data/Tile_*与metadata.xml的坐标信息。
- **处理：** OSGB转换为3D Tiles，配置源基准与地图原点。
- **产物：** tileset.json及子瓦片；模型高度偏移单独记录。

tileset.json及子瓦片连同坐标基准交付。

3.6.3 基础与进阶目录：扫描对象与空间

单个物体与室内扫描

2.AdvExps/e5_LiDAR-UE

1.3DScannerModel	手机扫描物体，MeshLab修复网格。
2.3DScannerScene	手机扫描室内，整理网格后导入UE。

验证：用已知长度核对网格尺度，检查孔洞与朝向；导入后查看几何和材质，检测任务另查碰撞。

采集范围不同，后续都需整理可用网格。

3.6.3 基础与进阶目录：条件深化

已有模板与外部数据分别准备

- **先读现成模板：**X680的XML与Cooked资源；BlueprintBuild的源资产和蓝图。
- **先备输入数据：**loadMeshXML所用FBX；OSGB的metadata.xml与模型；离线地形/影像TIF。
- **条件深化：**扫描设备、GIS服务或定制版本；课程入口和目标程序保持一致。

模板路线复用已有资源；扫描与GIS路线增加采集、转换和服务依赖。

3.6.3 基础与进阶目录：扩展条件

三类定制例程各有输入与验收证据

- **碰撞：** e3_CollisionMode选P1/P3，用stat game记录WorldTick。
- **地理：** e5_CesiumOffline将TIF切片，由Nginx:8080提供瓦片。
- **联网：** e1_CtrlAllUEinLAN需两机互通，接收端按I1。

范围： 三者位于3.CustExps、需完整版及以上；Nanite/Lumen/Chaos另按UE工程版本核可用性。

分别交帧耗时、瓦片加载与异机命令响应证据。

本章目录

从例程目录到三项综合实践

第三章 三维环境建模与仿真

3.1 三维仿真基础与工具选型

3.2 UE场景运行渲染与交互

3.3 多模态传感器数据生成

3.4 分布式仿真架构与数据接口

3.5 场景与载具资源开发

3.6 例程体系与综合实践

3.7 习题案例分析与课程设计

本节 4 个教学单元

3.6.1 入口与选题
教材对应：第3.3.1节

3.6.2 接口目录
教材对应：第3.3.1节

3.6.3 基础与进阶目录
教材对应：第3.3.1节

3.6.4 代表案例与实践路线
教材对应：第3.3.2节

3.6.4 代表案例与实践路线：综合应用总览

四类综合应用连接不同数据链



(a) 360度激光雷达与红外传感器　(b) 实时效能评估系统

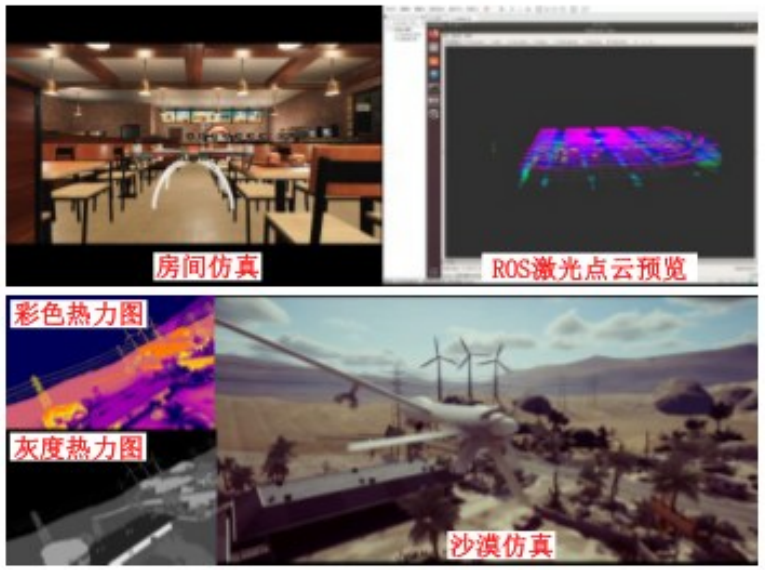
(c) 沉浸式VR实训应用　(d) 基于CityEngine规则快速生成城市场景

图3.9 高阶案例展示

综合任务要让环境变化、状态更新与传感器观测保持对应。

3.6.4 代表案例与实践路线：传感器应用链

激光雷达与红外来自各自观测链



(a) 360度激光雷达与红外传感器

图3.9(a) 360度激光雷达与红外传感器

- **点云：**房间环境与ROS显示。
- **红外：**沙漠环境与两类热力图。
- **对接：**记录配置、采样数据和显示方式。

传感器配置决定观测形式，采集接口交付数据。

3.6.4 代表案例与实践路线：状态与评价链

效能评价使用轨迹和任务规则



(b) 实时效能评估系统

图3.9(b) 实时效能评估系统

- **输入：** 载具状态与轨迹。
- **规则：** 任务区域、路径或评价条件。
- **输出：** 监测画面与计算指标。

状态数据进入评价程序，画面提供同步观察。

3.6.4 代表案例与实践路线：VR交互条件

VR训练连接头显、手柄与视景



(c) 沉浸式VR实训应用

图3.9(c) 沉浸式VR实训应用

- **设备：** VR头显与手柄。
- **显示：** 沉浸视野及实时画面共享。
- **条件：** 对应硬件、驱动和支持版本。

交互设备输入与视景输出共同构成训练界面。

3.6.4 代表案例与实践路线：规则化城市生成

规则化城市的输入、对照与交付



(d) 基于CityEngine规则快速生成城市场景

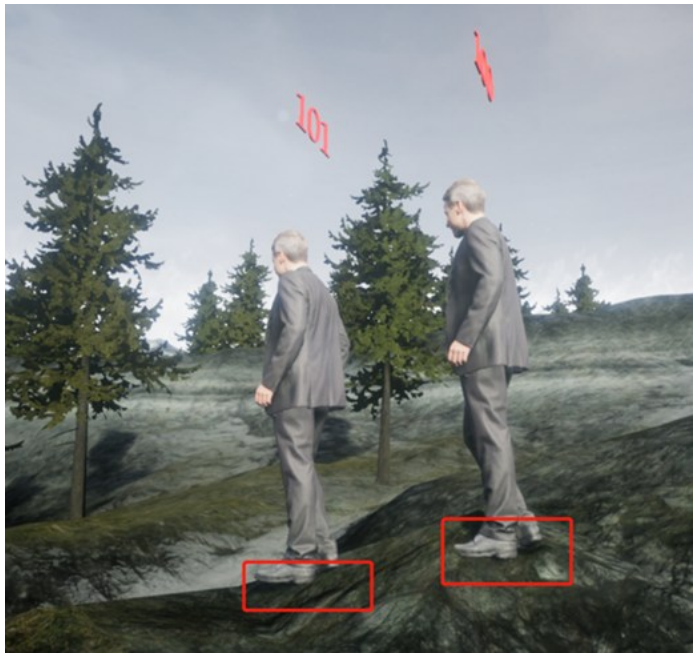
图3.9(d) 基于CityEngine规则快速生成城市场景

- **输入对应：**矢量组织道路，高程给地形，影像给背景。
- **规则对照：**固定地理输入，改变建筑规则并保存结果。
- **场景交付：**模型、纹理、地理原点与版本一同保存。

固定地理输入并保存生成规则，才能复现同一城市资源。

3.6.4 代表案例与实践路线：贴地任务入口

同一张地图有两种放置处理



图： MountainTerrain中的人物放置

- **启动：** 在code内运行ue4.bat，选MountainTerrain。
- **执行：** python TargetPlaceDemo.py
- **预期：** 出现100/101号人物，比较脚底与地表。

环境： Python38Run.bat

普通调用使用给定高度，贴地调用由引擎处理落地。

3.6.4 代表案例与实践路线：两次实际调用

原模板使用相邻位置进行展示

```
ue.sendUE4Pos(
  100,30,0,[-16.479,1.45,-177.181],[0,0,math.pi])

ue.sendUE4Pos2Ground(
  101,30,0,[-17.479,1.45,0],[0,0,math.pi])
```

对象：ID为100/101，样式30；x相差1米，y相同。
单位：位置NED米；姿态为roll/pitch/yaw弧度，偏航为π。

两个人物的x相差1米，位置与方法同时发生变化。

3.6.4 代表案例与实践路线：贴地消息机制

sendUE4Pos2Ground发送贴地标志

- **消息字段：**贴地标志1234567891、对象ID、模型样式、转速、位置和姿态。
- **作用过程：**UDP状态 → RflySim3D识别贴地标志 → 当前地形与模型偏移决定落点。
- **数据方向：**本调用发送状态；Python高度查询另读地图PNG/TXT及其标定。

输出解释：普通调用把模型原点放到给定z；贴地调用由引擎结合地表和模型中心离地偏移确定落点。

Python发出贴地请求，场景引擎处理模型落点。

3.6.4 代表案例与实践路线：同点对照条件

固定位置后比较放置方法

- **固定：** MountainTerrain、模型样式30、x/y与姿态；分次创建避免对象重叠。
- **改变：** 普通位置调用与贴地调用；记录输入z、对象ID和可见落点。
- **解释：** 地形位置、碰撞/地表几何与模型中心离地偏移共同影响脚底位置。

预期现象： 地表与偏移配置正确时，贴地人物脚底贴合地面；悬空或穿地时，分别核对地表碰撞和模型原点。

固定地形与横向位置，才能解释高度处理差异。

3.6.4 代表案例与实践路线：脚底间隙算例

按NED高度计算人物脚底与地面的间隙

- **题设：**同一NED系，地面Down坐标177.0 m，人物参考点176.0 m。
- **模型：**脚底相对参考点向Down方向偏移+0.8 m。
- **代入：**$g = 177.0 - (176.0 + 0.8) = 0.2\,\mathrm{m}$

判读：g>0表示脚底悬空；g=0才是几何贴合。此数值仅为教学题设，贴地落点须以三维端实测核对。

脚底高于地面0.2 m；贴地调用仍须核模型原点。

3.6.4 代表案例与实践路线：贴地对照

同一地形上对照参考高度与贴地放置

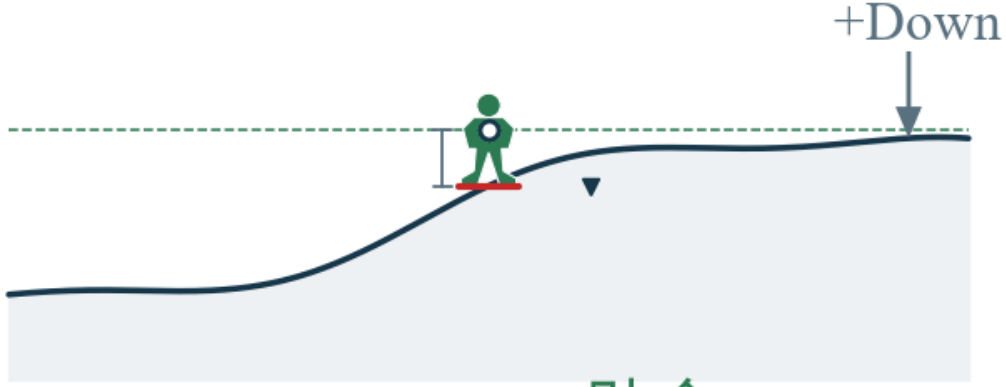
○ 原点　– 脚底　– 地面　脚底偏移 +0.30　示意单位

普通：原点高度固定



g = +0.198 悬空

贴地：调整原点高度



g = +0.000 贴合

$$g = z_{ground} - (z_{ref} + 0.30)$$

固定地形与模型原点后，两种放置方式的脚底差异才可解释。

3.6.4 代表案例与实践路线：场景生成代码

TargetCreate.py按列表创建九个元素

```
for i in range(len(pos)):
    ue.sendUE4PosNew(
        copterID=i+10002, vehicleType=pos[i][0],
        PosE=pos[i][1:4], AngEuler=ang[i],
        runnedTime=-1)
```

编号：九项对应10002–10010，vehicleType取每行类型。
状态：PosE为[x,y,z]米；AngEuler为[roll,pitch,yaw]弧度。
时间：runnedTime=-1保留本例默认值，客户端原样发送。

实例ID来自循环，类型与位置来自对应列表项。

3.6.4 代表案例与实践路线：随机偏移与输出

随机生成后的布局是本次输入结果

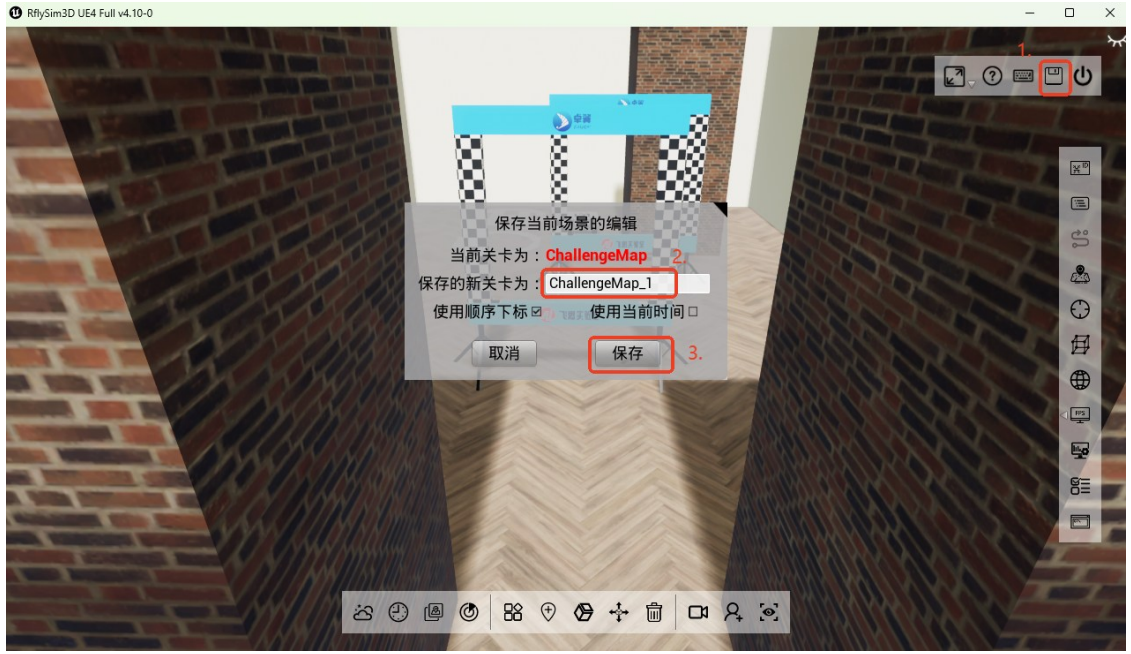
```
pos[i][1] += random.uniform(-0.1,0.1)
pos[i][2] += random.uniform(-0.2,0.2)

print(x[2],y[2])
```

偏移： 当前列表的两个815方框，x增量为±0.1米、y为±0.2米。
记录： 创建前打印完整pos列表，或创建后保存本次布局JSON。
打印： 1 1属于未扰动的824立柱，不能代表两个方框的位置。

重跑生成脚本会改变部分方框的位置。

3.6.4 代表案例与实践路线：界面保存布局

保存场景将本次Actor信息写入JSON



图： 命名并保存自定义场景

- **创建：** 在ChallengeMap运行TargetCreate.py一次。
- **操作：** 保存场景 → 命名 → 确认。
- **产物：** 本次布局的JSON及关联文件。

保存当前布局后，重载才有同一份描述可读。

3.6.4 代表案例与实践路线：两类保存目录

布局文件与地形文件保存在不同位置

布局文件：

RflySim3D/RflySim3D/Saved/
AllActorSave/<场景名>.json及.png。
JSON保存对象布局，PNG用于预览。

地形文件：

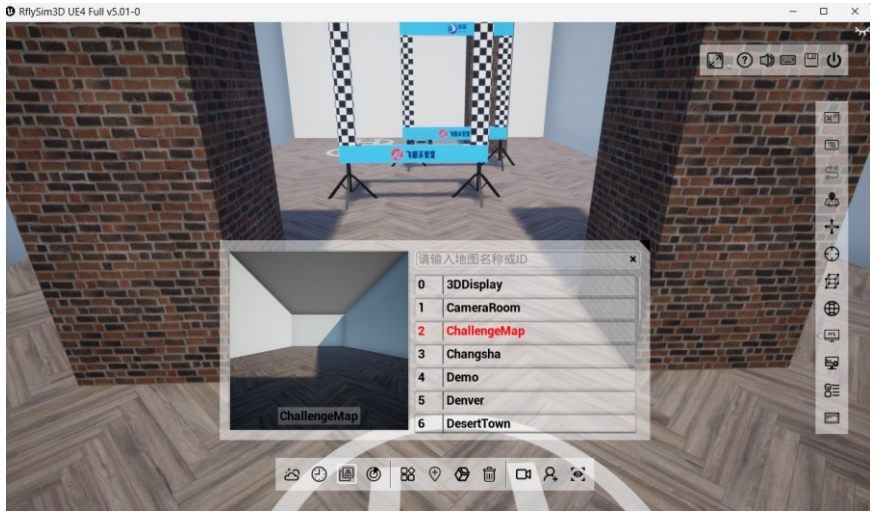
CopterSim/external/map/
<场景名>.txt及.png。
TXT给标定，PNG存高程。

- **重载：**选择已保存场景，按同一ID核对类型、位置与姿态；不重跑随机生成。

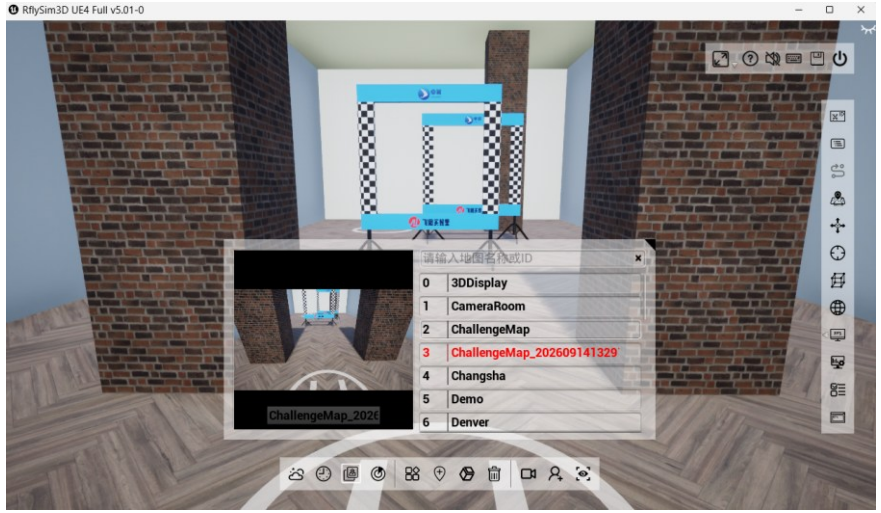
步骤：停止创建脚本 → 切到其他地图 → 选择保存场景 → 逐项核对。

3.6.4 代表案例与实践路线：重载对照产物

场景保存与重载的布局对照



图：生成布局与地图选择界面



图：重载保存场景后的布局

对照：以砖柱为参照，观察方框间的前后与左右关系。

记录：同一方框的ID、位置与姿态写入JSON；重载对照时不重跑随机生成。

保存把本次布局变为可重载资源，方框与立柱提供直观的空间参照。

3.6.4 代表案例与实践路线：保存与重跑

重跑生成脚本与重载保存场景回答不同问题

重跑生成脚本：
随机项重新取值，得到新布局。

保存本次布局：
JSON写出当前一次生成结果。

重载旧JSON：
读取已保存的JSON，恢复保存时的位置。

判据：按实例ID逐个比对位置和姿态。

3.6.4 代表案例与实践路线：状态与像素两条链

查询结构体与采集图像分别启动



图：1号飞行器与人物目标（UE4 Full v5.01）

场景：前景是1号飞行器，人物目标标为100；远处人物被放大。

状态链：查询相机与对象结构，获取位置、包围盒等字段。

像素链：WithCam版本另行请求图像，从vis.Img[i]读取画面。

状态回传给字段，图像采集给像素数组。

3.6.4 代表案例与实践路线：请求类型与标识

同一查询接口用类型区分目标

```
ue.reqCamCoptObj(0,0)
ue.reqCamCoptObj(1,[1,100])
ue.reqCamCoptObj(2,'Landscape_1')
ue.initUE4MsgRec()
```

启动： GetCamObjDemo.bat；等待GPS与EKF初始化完成。
类型： 0按相机SeqID；1按实例ID；2按场景对象名。
条件： 首行相机请求用于WithCam版；实例1/100与Landscape_1须先存在。
接收： initUE4MsgRec启动监听，getCamCoptObj读取缓存。

类型和标识共同决定请求哪一个结构体。

3.6.4 代表案例与实践路线：读取状态字段

hasUpdate表示缓存收到新状态

```
Copt = ue.getCamCoptObj(1,1)
if isinstance(Copt,UE4CtrlAPI.CoptReqData):
    if Copt.hasUpdate:
        print(Copt.CopterID,Copt.PosUE,Copt.BoxExtent)
        Copt.hasUpdate = False
```

运行：在Python38Run.bat终端执行python GetCamObjDemo.py。

位置：PosUE为模型原点，boxOrigin为几何包围盒中心。

尺寸：BoxExtent为半尺寸，完整长宽高为其2倍。

新状态：hasUpdate为真时处理一次，随后置False，等待下一次回传。

3.6.4 代表案例与实践路线：状态更新

hasUpdate只在缓存收到新状态后报告一次

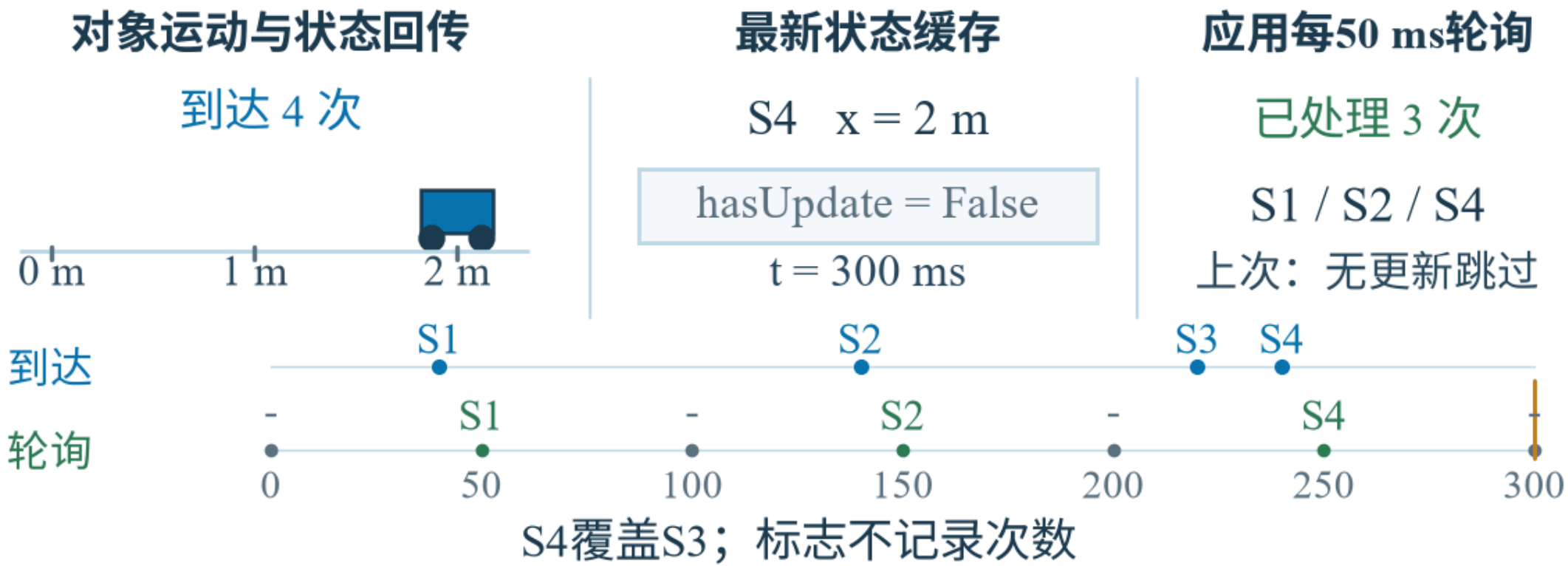
- **请求：**按类型和目标标识订阅对象状态。
- **到达：**SDK缓存新结构体并设置更新标志。
- 读取：发现hasUpdate后处理一次并清除标志。

判据：静止时仍可有周期回传；按观察到的新缓存更新计数。

轮询次数不是到达次数，重复读取不算新数据。

3.6.4 代表案例与实践路线：状态更新

对象停住以后，状态消息仍可继续更新



图：横轴单位为ms；S为本图消息编号，布尔标志不是消息计数器

3.6.4 代表案例与实践路线：图像配置

Config.json配置两路640×480 RGB

"SeqID": 0, "TypeID": 1, "TargetCopter": 1
"DataWidth": 640, "DataHeight": 480
"DataCheckFreq": 30
"SensorPosXYZ": [0.3,-0.15,0]
"SeqID": 1, "SensorPosXYZ": [0.3,0.15,0]

类型：两路RGB，640×480，横向安装间隔0.3米。
频率：DataCheckFreq=30请求30 Hz更新，实际接收频率另测。
传输：SendProtocol首项0选择本机共享内存。

相机序号、挂载对象和图像尺寸共同定义采集请求。

3.6.4 代表案例与实践路线：双相机时序

先核Config，再请求传感器和接收两路像素

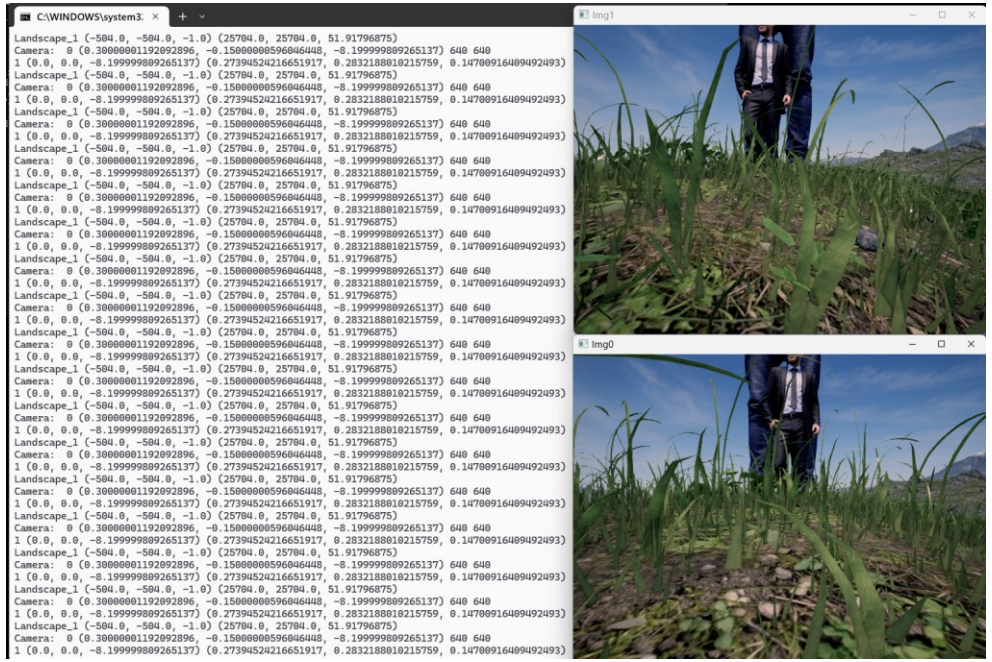
- **配置：**两路相机各写序号、挂载目标和分辨率。
- **启动：**SITL初始化后运行配套采集脚本。
- **接收：**逐路检查新帧、数组形状与时间戳。

结束：停止脚本并关闭窗口，保留配置和记录。

两路窗口只是观察手段，数组形状才是尺寸判据。

3.6.4 代表案例与实践路线：启动像素接收

相机像素与状态回显同时可见



图：终端状态与两路相机图像

运行： python GetCamObjDemoWithCam.py

接收： hasData[i]为真时读取vis.Img[i]。

读图： Camera 0与对象1给位置；两路图像显示人物和草地。

尺寸： 两次640的含义见下页。

状态与像素来自不同通路，须分别核对对象身份与新帧。

3.6.4 代表案例与实践路线：尺寸与打印错误

打印两次宽度不会得到图像高度

Cam.DataWidth, Cam.DataWidth # 原脚本末行

原打印：两个值都是宽度，不能代表宽和高。

Cam.DataWidth, Cam.DataHeight # 课堂修正

字段：DataWidth为640，DataHeight为480。

---

vis.Img[i].shape # 像素数组形状

数组：按高、宽、通道排列；本配置预期(480,640,3)。

3.6.4 代表案例与实践路线：实践记录与解释

三项实践的结果与排错依据

- **贴地：**普通调用使用给定z；贴地调用由引擎依据地表与中心离地偏移确定落点。
- **保存：**JSON保留同次布局；用同一ID核对位置姿态，重跑随机脚本会改变输入。
- **采集：**状态与像素分别接收；无图像先查配置、请求与接收，尺寸按数组的高、宽核对。

输出与输入一一对应，才便于解释差异。

3.6.4 代表案例与实践路线：配置与采集资料

从配置文件继续到状态与像素源码

0.ApiExps/e6_RflySim3DCtrlAPI/10.GetCamObjDemo

- **配置：**Config.json；修改一台相机的位置，保留另一台作为参照。
- **状态：**GetCamObjDemo.py；追踪请求类型、目标标识和hasUpdate。
- **像素：**GetCamObjDemoWithCam.py；追踪请求、接收和vis.Img。
- **文档回指：**3.4.4的视觉配置与采集；3.4的UE4CtrlAPI接口定义。

对照同一例程的配置、状态脚本和图像脚本。

课堂练习

每次只改一个输入，再预测可观察差异

- 贴地：固定地图与x、y，比较给定z和贴地落点。
- 保存：保存一次布局，重跑脚本，再重载旧JSON。
- **采集：**只改1号相机挂载偏移，0号保留参照。

记录：各项写输入、预测、结果和未通过原因。

预测、运行和记录应使用同一组条件。

练习参考要点

用高度差、布局文件和新帧标识作判断

- **贴地：**北东地z向下为正，同点比较地表与输入参考点。
- 保存：重跑随机项可变；旧JSON重载应复原旧布局。
- **采集：**0号保持参照，1号用新帧时间戳和形状判定。

$$\Delta z = z_{\text{ground}} - z_{\text{input}}$$

排除输入变化和旧数据，才知道观察到的差异来自哪一步。

本节小结

从例程选题到结果解释形成闭环

- **贴地：**固定地图与横向位置，比较普通放置与贴地调用的脚底落点。
- **保存：**记录本次随机布局及JSON；重载保存场景与重跑生成脚本分开比较。
- **采集：**按状态更新标志与图像时间戳识别新数据，核对数组形状与处理耗时。

每项实践都应保留入口、输入、产物与判据。