本章目录

场景与载具资源开发

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

3.5.1 三维资产集成路线

教材对应：第3.2.2.1节

3.5.2 UE资产管理

教材对应：第3.2.2.2节

3.5.3 载具XML与执行器

教材第3.2.1.2、3.2.2.3–3.2.2.4节

3.5.4 资源验证与复用

教材第3.2.1.2、3.2.3节；验证流程补充

3.5.1 三维资产集成路线：三路交付与两类资源

按资源来源选择场景与载具的交付路线

路线：UE资产整理依赖后Cook；GLB由运行时插件解析；地形影像按瓦片与服务加载。
选择：先看输入格式，再核目标程序；以下展开UE分支的场景与载具。

- 场景：在UE中制作地图、材质与蓝图，Cook后部署。
- **载具：**静态网格体或蓝图模型配可扩展标记语言（XML）配置，登记模型与部件。
- 运行：三维端加载资源；CopterSim、Python或Simulink发送编号、类型与位姿。

场景资产与载具模型都要连同依赖交付，显示状态由仿真数据驱动。

3.5 场景与载具资源开发

3.5.1 三维资产集成路线：来源与产物

资源进入仿真器的整体关系

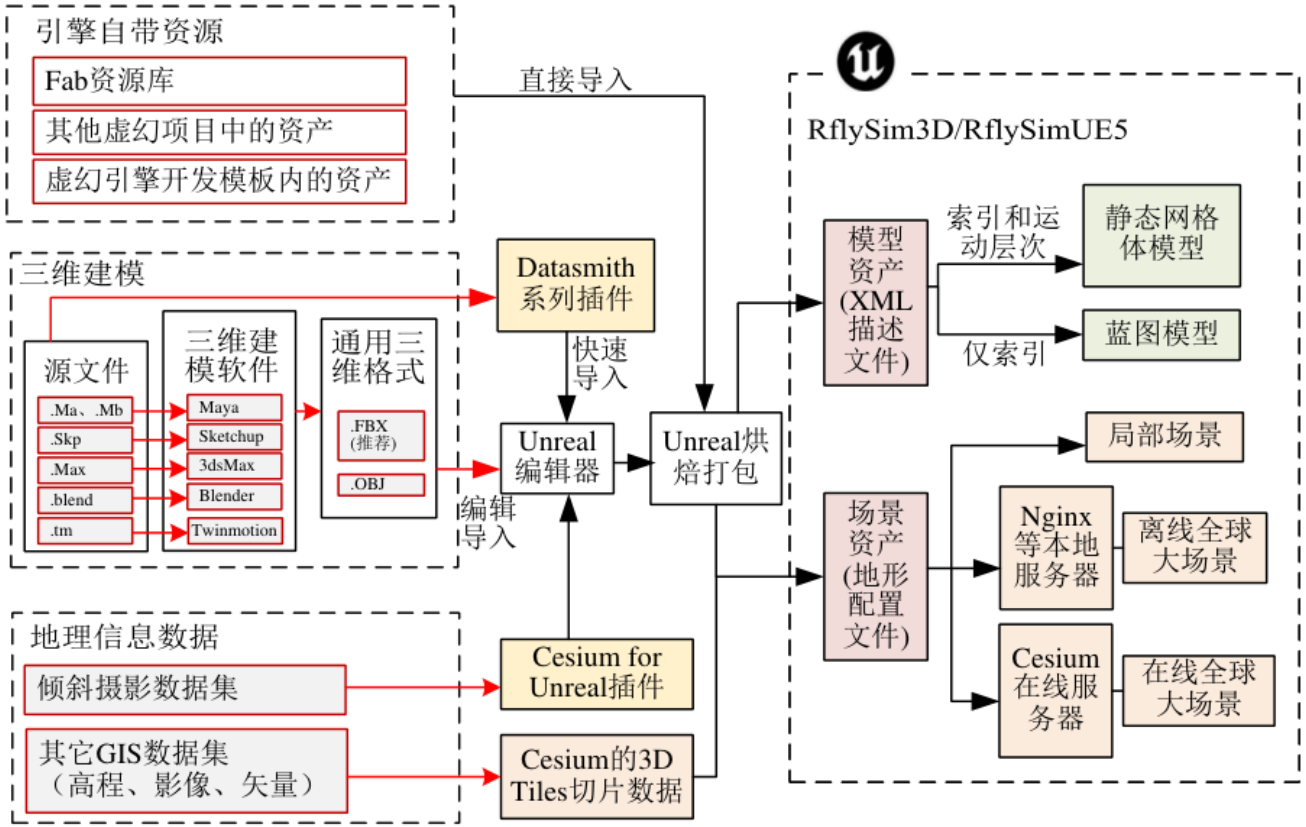


图3.7 RflySim3D三维资源导入方法

读图：先找输入，再沿箭头核对转换工具与产物。

本节示例：FBX经编辑器；GLB运行时解析；地理数据转瓦片。

3.5.1 三维资产集成路线：引擎资源

已有UE资源可直接进入编辑器

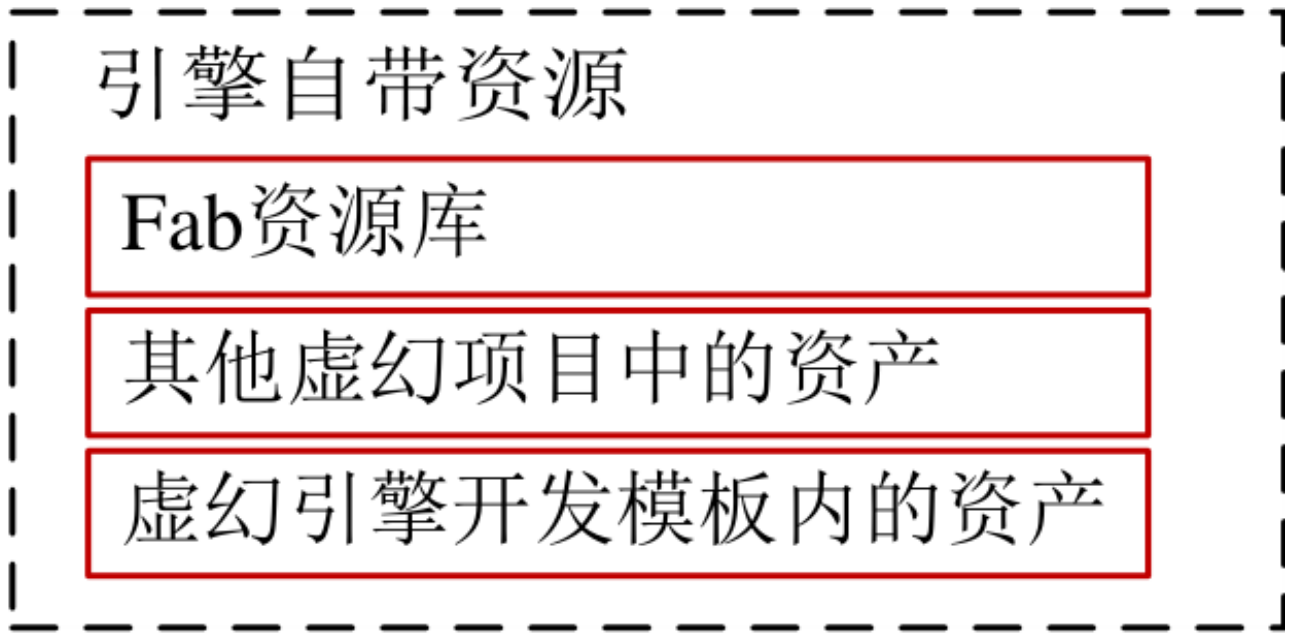


图3.7（引擎资源局部）三类可复用资源

- 资源库：核资产许可。
- 仅导入获准的资源。
- 项目迁移：连同材质、
- 纹理与引用一起迁移。
- 开发模板：从现成场景
- 出发，再逐项修改。

同为UE资产，先核许可和依赖，再进入工程与烘焙。

3.5.1 三维资产集成路线：模型资产

XML对两种载具资产承担不同职责

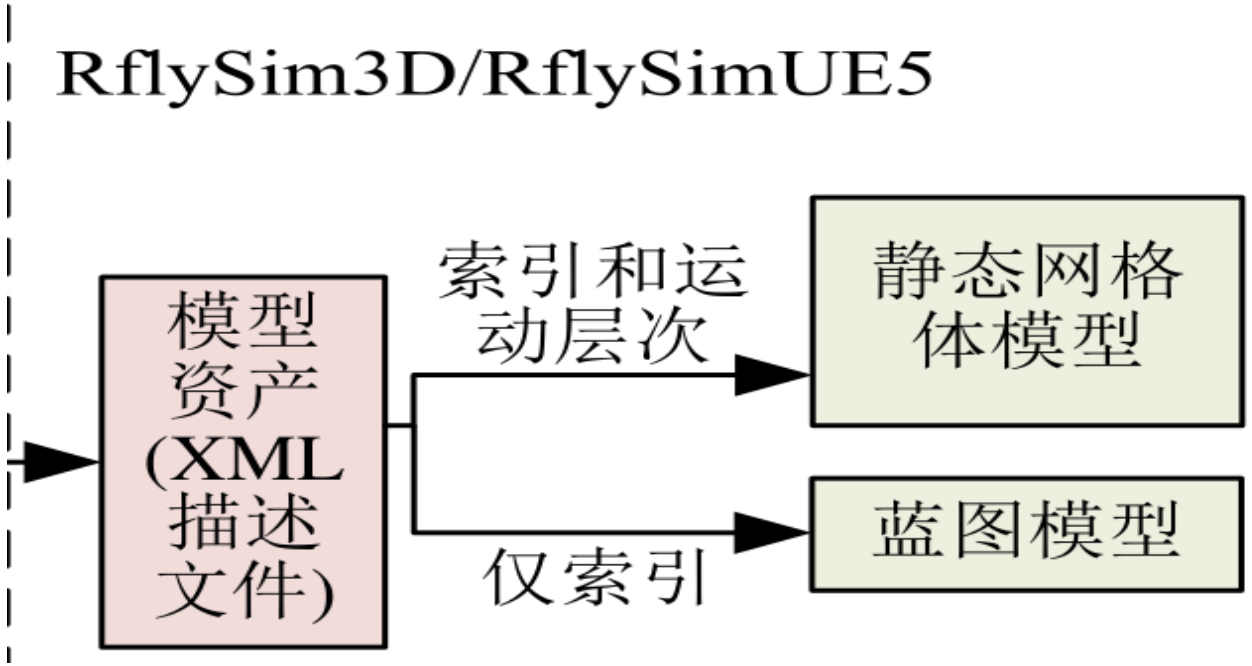


图3.7（模型资产局部）XML索引与运动层次

- 静态网格体：在XML中写部件运动层次。
- 蓝图模型：把层级和动作逻辑封装在资产中。
- 共同入口：XML登记三维端要加载的资产。

静态网格体在XML写运动层次，蓝图把动作封装在资产内。

3.5.1 三维资产集成路线：建模文件

按要保留的内容选择FBX、OBJ或Datasmith

- FBX：携带网格、层级与部分材质引用，导入后核部件轴心。
- **OBJ：**主要交接静态几何与UV，外观依赖MTL和纹理文件。
- Datasmith：面向场景层级、相机与灯光的交接；按版本核插件。
- 例：多部件旋翼选FBX，导入后检查轴心与部件层级。
- 验收：核网格数量、贴图通道和蓝图动作是否符合任务。

逐项核层级、材质贴图与动作，再决定导入格式。

3.5.1 三维资产集成路线：三条路线

按资源和目标环境选择导入路线

- **编辑器导入：**FBX或UE资产 → 依赖整理与Cook → Content资源；XML登记模型。
- **运行时解析：**GLB等本地网格 → UMG配置与解析 → 仓库条目 → 场景实例。
- **地理数据加载：**OSGB、高程或影像 → 对应瓦片产品 → Cesium组件与数据服务。

Cook（烘焙）：将编辑器资产及其依赖转换为目标程序可加载的运行文件。

TM旧版例程：e1_TwinmotionContent；TM2022.1.2 → .tm → UE4.27。

TM小屋例程：5.TwinmotionUsage；TM2023.1.2 → .udatasmith及资源 → 配套UE5。

路线由资源格式、目标程序和数据依赖共同决定。

3.5.1 三维资产集成路线：最短网格入口

运行时导入：配置模型仓库



图：warcraft模型的仓库配置界面

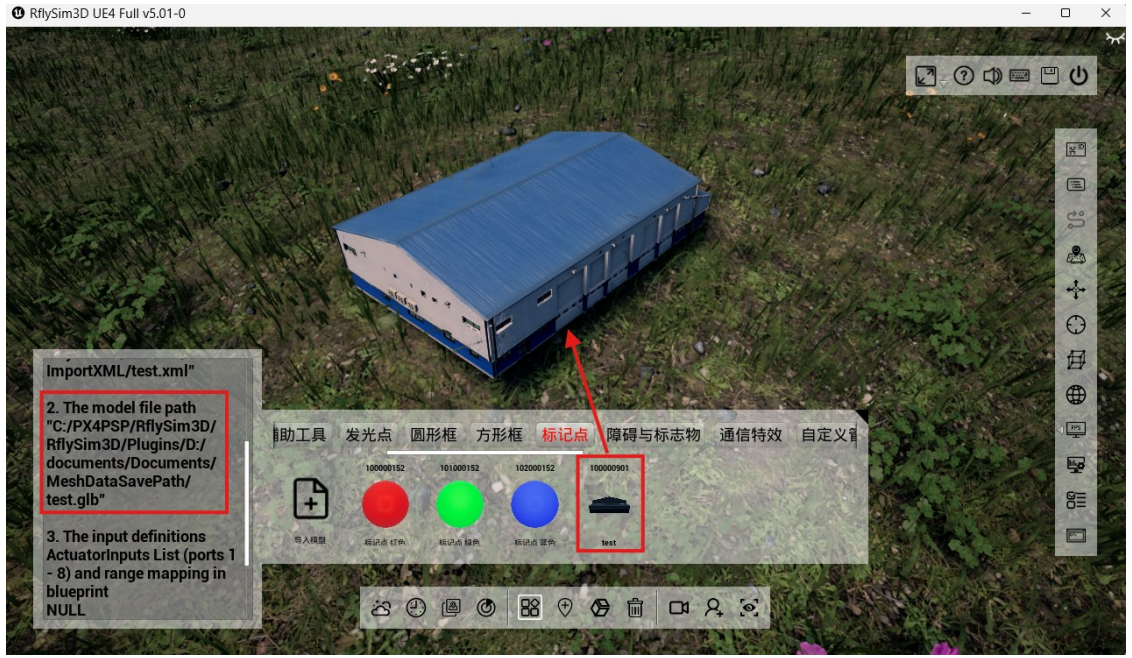
- **图示：**warcraft飞机的配置界面；练习文件选test.glb。
- **配置：**填写名称与未占用的资源编号，调整姿态和保存路径。
- **条件：**本例GLB采用完整版运行时插件。

仓库配置保存资源定义，场景创建生成独立实例。

3.5 场景与载具资源开发

3.5.1 三维资产集成路线：运行时操作与产物

GLB模型从仓库进入场景



图：test.glb入库与场景实例（UE4 Full v5.01）

入口： 1.BasicExps/ e4_CusLoadObj/ 2.loadSimpleMesh。

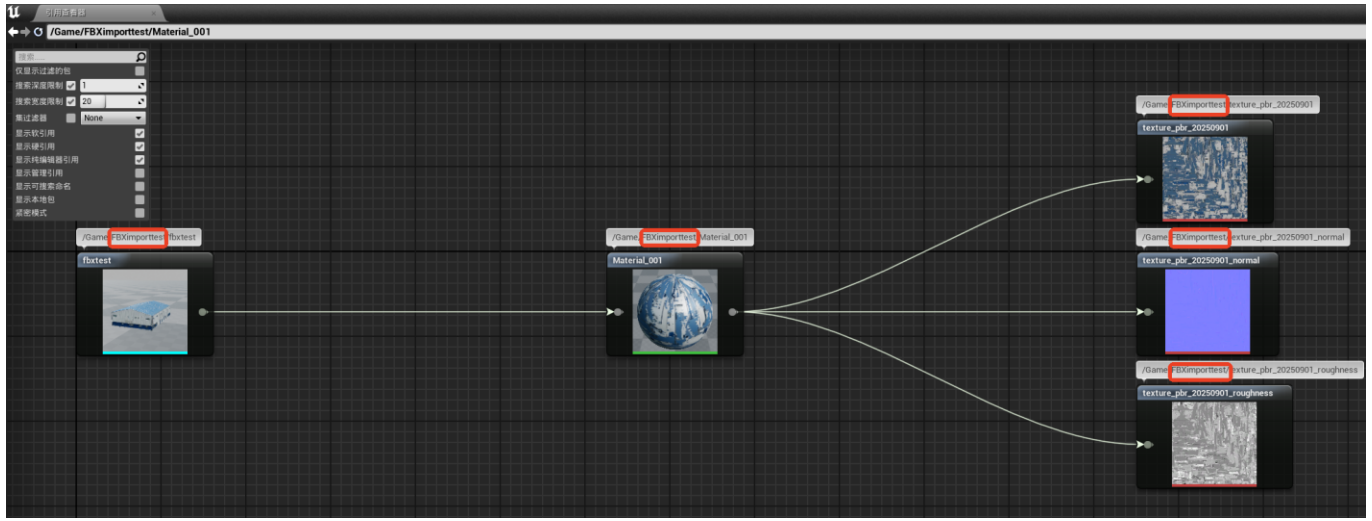
资源： 列表新增test条目，红框中的索引用于选择模型。

实例： 拖入后出现建筑，屋顶、墙面与材质可见；对象ID另行分配。

模型文件定义外观，仓库索引供创建，场景实例承载位置与姿态。

3.5.1 三维资产集成路线：FBX依赖

沿引用关系交付完整的模型外观



入口： 1.BasicExps/e4_CusLoadObj/3.loadMeshXML。

图： fbxtest的材质与纹理引用

- **引用链：** 网格经材质连接三张纹理。
- **缺失后果：** 形状可见，颜色或受光细节可能丢失。
- **完整部署：** Cook模型与依赖，保留引用路径。

模型、材质与纹理形成引用链，部署时一同交付对应运行资源。

3.5.1 三维资产集成路线：Cook与路径

UE4.27的Cook目录对上XML路径

- **版本分支：**本例采用UE4.27；UE5须由匹配工程重新Cook。
- **Cook输出：**Saved/Cooked/WindowsNoEditor/<项目名>/Content。
- **运行部署：**RflySim3D/RflySim3D/Content，保留模型、材质和纹理的目录层级。
- **路径对应：**/Game/FBXimporttest/fbxtest定位该子目录中的资产，不写.uasset扩展名。

操作：保存所有资产，烘焙Windows内容，确认日志成功。

MeshPath使用UE资产路径，省略文件扩展名。

3.5.1 三维资产集成路线：静态模型XML

fbxtest.xml把资源登记成可创建样式

<ClassID>910</ClassID>
<DisplayOrder>100</DisplayOrder>
<ModelType>0</ModelType>
<MeshPath>/Game/FBXimporttest/fbxtest</MeshPath>

索引： 100×100000+910=10000910。

加载： ModelType=0选择静态网格；MeshPath定位已部署的资产。

注册： XML随资产放入Content/FBXimporttest；重启后查同名TXT。

前提： 先准备FBX并Cook；例程有配置与脚本，没有配套模型。

3.5.1 三维资产集成路线：样式与实例

Python用样式索引创建一个对象

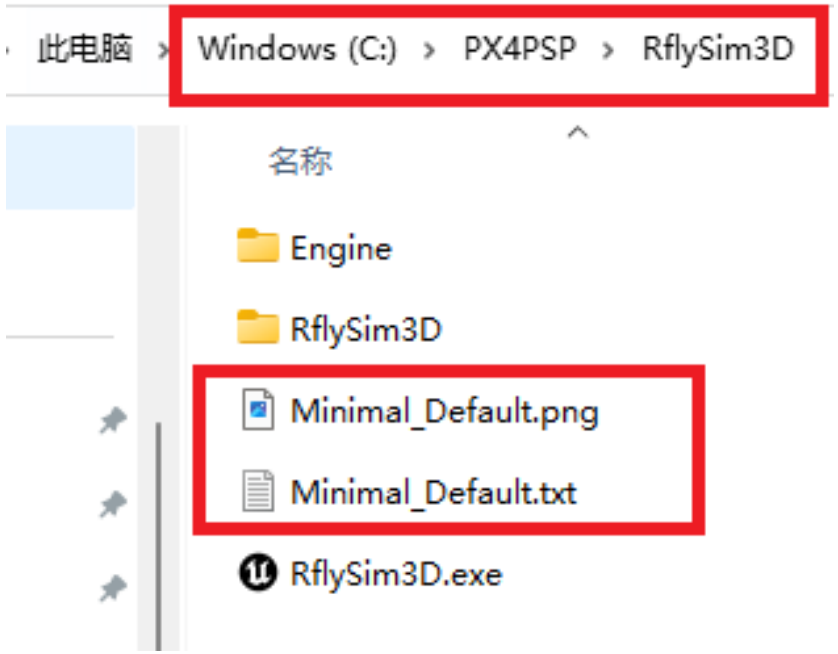
```
ue.sendUE4Cmd('RflyChangeMapbyName VisionRing')
ue.waitForMapLoaded('VisionRing')
time.sleep(3)
ue.sendUE4Pos(100,10000910,0,[2.5,0,0],[0,0,0])
```

参数：实例ID100；样式10000910；位置[2.5,0,0]米，姿态为弧度。
运行：从本例目录启动loadmodel.bat，等待VisionRing加载后创建对象。
预期：资源齐全时出现模型；再次给ID100发位姿只更新该实例。

同一样式可创建多实例，位置更新按实例ID定位。

3.5.1 三维资产集成路线：场景与地形文件

三维场景和地形高度分别交付



- **视景端：** RflySim3D加载关卡.umap及依赖。
- **扫描输出：** 在RflySim3D目录生成同名PNG高度矩阵与TXT标定。
- **部署：** 将PNG与TXT一起复制到CopterSim/external/map。

图： Minimal_Default地形文件

视景资源负责显示，高度文件服务地形查询。

3.5.1 三维资产集成路线：地理数据产品

地理分支先区分模型、高程和影像

输入	处理或元数据	入口与产品
OSGB模型	Data/Tile_*、metadata.xml	tileset.json及子瓦片
高程TIF	地形高度切片	layer.json及分层地形
影像TIF	表面影像切片	tilemapresource.xml及TMS

SRS：坐标参考系；SRSOrigin：源模型原点。
地图原点连接仿真坐标，模型高度偏移调整数据间的相对位置。

三种地理数据保留各自的产品与加载入口。

3.5.1 三维资产集成路线：地理分支前提

GIS深化按数据和目标版本选择

- **OSGB路线：**2.AdvExps/e3_CusGIS/1.ObliModelMap；先备Data/Tile_*和metadata.xml。

- **离线地形/影像：**3.CustExps/e3_CusGIS/e5_CesiumOffline；完整版UE5.8及本地数据服务。

- **准备：**OSGB、TIF原始数据与服务软件需另备；交付瓦片根目录、服务地址和地图原点，高度偏移单独记录。

OSGB路线交付模型瓦片；离线地形路线还依赖高程、影像与数据服务。

3.5.1 三维资产集成路线：地理数据

倾斜摄影与GIS数据需转换成分块资源

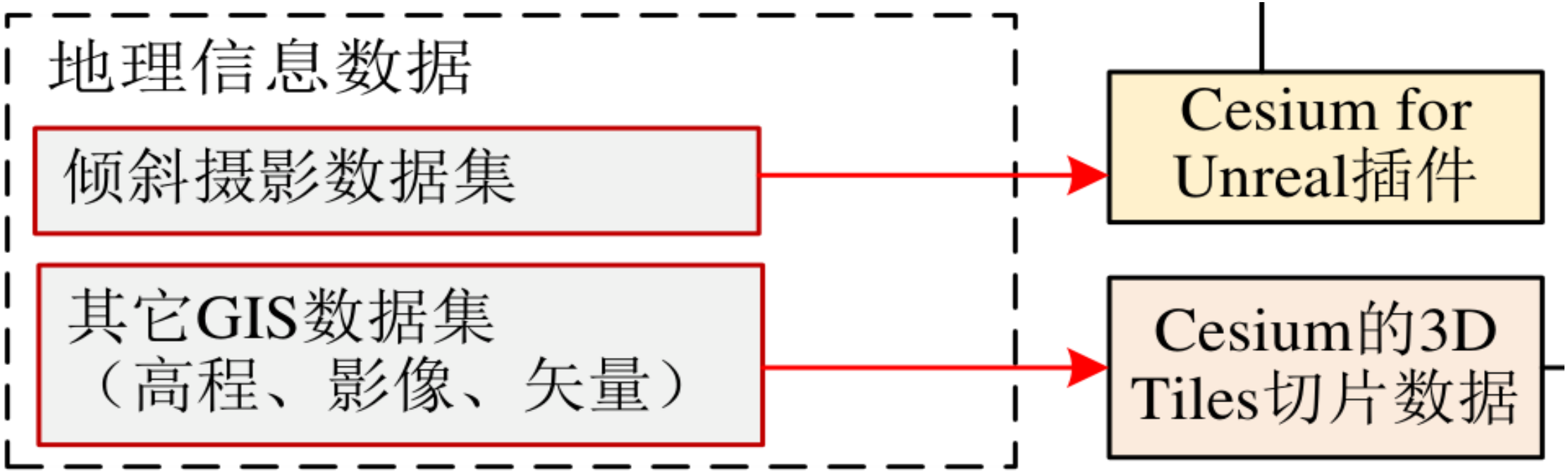


图3.7（地理数据局部）：OSGB与高程、影像经切片工具形成可加载资源。

本章目录

场景与载具资源开发

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

3.5.1 三维资产集成路线

教材对应：第3.2.2.1节

3.5.2 UE资产管理

教材对应：第3.2.2.2节

3.5.3 载具XML与执行器

教材第3.2.1.2、3.2.2.3–3.2.2.4节

3.5.4 资源验证与复用

教材第3.2.1.2、3.2.3节；验证流程补充

3.5.2 UE资产管理：场景加载路线

同一场景配置可指向局部或全球大场景

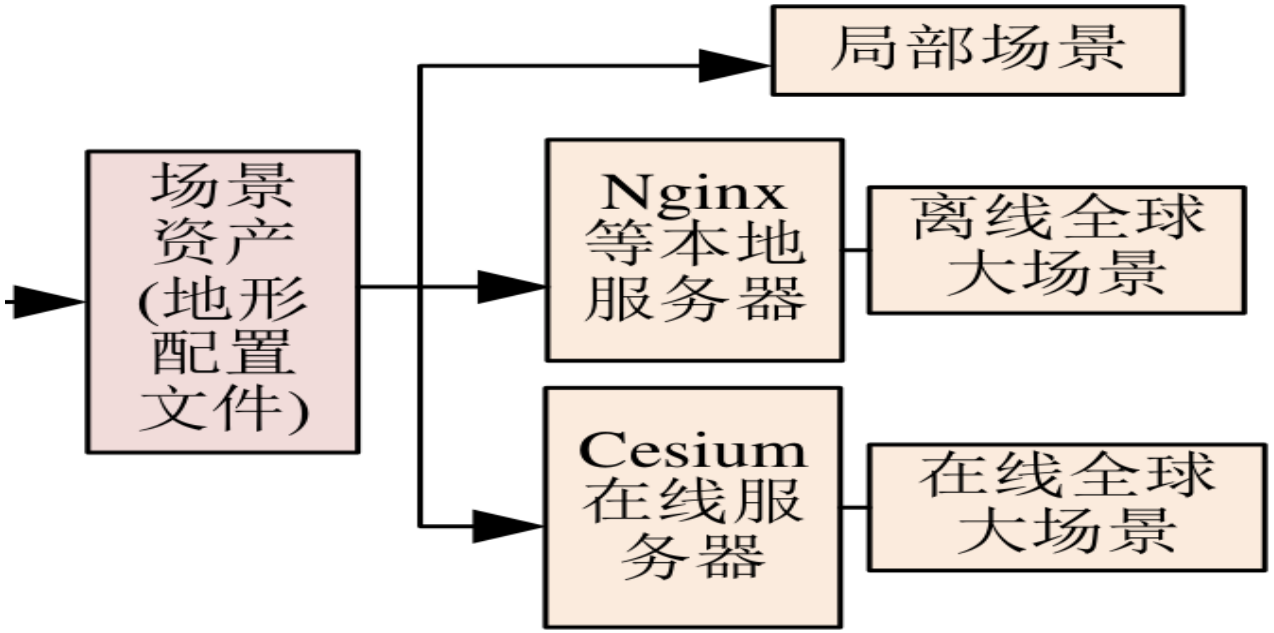


图3.7（场景局部）三种加载路径

- **局部场景：** 资产随工程烘焙并部署。
- 离线全球：本地服务提供切片数据。
- **在线全球：** Cesium服务按需加载。

局部资产、离线服务与在线服务有不同依赖。

3.5.2 UE资产管理：局部环境差异

环境资源怎样组织成可重复的对照



- **地图版本：**保存环境资源与坐标原点。
- **共用任务：**相机参数、目标轨迹和评价规则保持一致。
- **变量记录：**单独改变地表或遮挡，记录对应资源版本。

图3.8(a) 局部场景模型

环境对照固定任务条件，以资源版本记录所改变的背景与遮挡。

3.5.2 UE资产管理：尺度轴向枢轴

把部件轴心和方向落实到资源配置

- **配置单位：**本例XML相对位置25.426厘米对应0.25426米；按字段单位写值可避免百倍尺度错误。
- **轴心位置：**旋翼部件原点应落在桨轴，XML相对位置把这个轴心放到机身对应位置。
- **转轴方向：**方向向量只表示绕哪根轴旋转，不是第二个位置，也不表示位移距离。
- **动作对应：**部件绕自身正确轴心转动；若轴心偏离，桨叶会绕偏心轨迹扫动。

部件原点确定轴心，相对位置安放轴心，方向向量指定转轴。

3.5.2 UE资产管理：单位与轴向

先统一厘米和前右上，再放置旋翼轴心

- 尺度：本例UE中1个虚幻单位对应1厘米，导入前统一模型单位。
- **坐标：**UE采用前右上；本例3ds Max模型的横向轴与UE相反。
- 核对：右前旋翼应落在机身前方和右侧，轴心在桨轴。
- 例：机臂长100 cm时，在UE中应量得100单位。

$$y_{\mathrm{UE}} = -y_{\mathrm{source}}$$

不换单位或Y轴方向，模型会缩放百倍或左右镜像。

3.5.2 UE资产管理：坐标换算算例

X680右前旋翼相对机身的位置怎样算

- 已知：机身中心为(-0.449,-0.363,0)厘米。
- 旋翼轴心为(24.977,-24.439,11.222)厘米。
- 结果：源横向差为-24.076厘米，UE横向取反为+24.076厘米。

$$\Delta x = 24.977 - (-0.449) = 25.426\ \text{cm}$$

$$\Delta y = -24.439 - (-0.363) = -24.076\ \text{cm}$$

先相减求局部位置，再把源模型横向轴换成UE方向。

3.5.2 UE资产管理：可见网格与碰撞

材质、碰撞和细节层次各自影响输出

- **材质：**纹理引用完整时，表面颜色和细节才随Cook资源一同加载。
- **碰撞：**盒、球、胶囊或简单凸包参与检测；贴地任务依赖可用地表。
- **预处理：**DCC减面（Decimate）减少面数；重拓扑（Retopology）重排面连接。按需去除不可见细节，再导入UE。
- **LOD交接：**静态网格可在UE生成或导入LOD；骨骼网格也支持生成或导入，还要核对骨架、蒙皮权重与动作变形。
- **结果判断：**按屏幕占比切换LOD，保留任务轮廓；同镜头比帧时间（课件3.2），贴地和射线另核碰撞。

可见表面、碰撞几何和远近细节分别配置。

3.5.2 UE资产管理：光照与材质

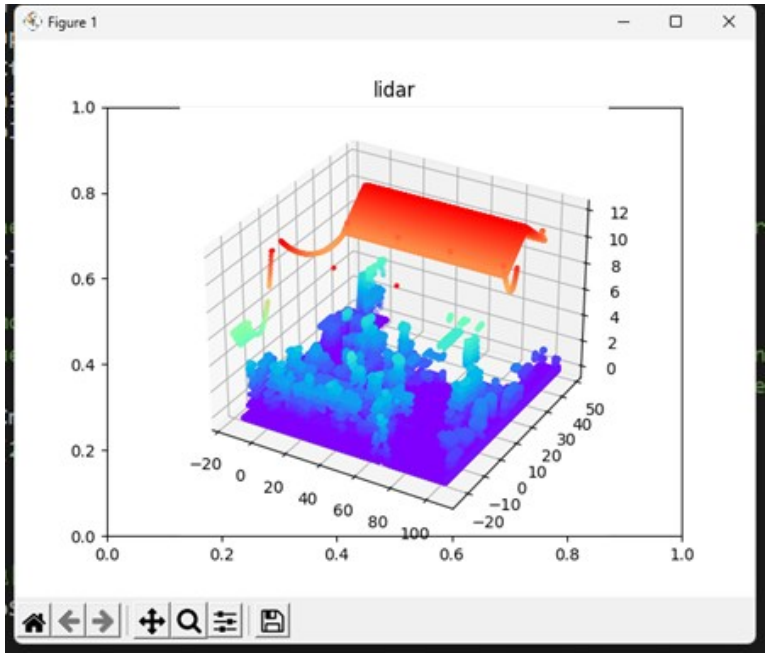
导入后逐项检查灯光、材质与贴图引用

- 灯光：场景交接可能转换灯光，实时效果仍需在UE中核对。
- **材质：**FBX不能完整保留原软件的着色网络，检查贴图通道。
- 引用：材质和纹理随项目交付，烘焙后再核外观。
- 检查顺序：先找丢失贴图，再核材质槽与灯光类型。
- 验收：同一镜头比较Cook前后外观，并保留差异截图。

几何正确但画面全灰时，先查材质与依赖。

3.5.2 UE资产管理：地形采样表示

地形高度栅格可转换为采样点



- **输入：** 高度PNG与TXT标定。
- **变换：** 平面采样位置加高程，形成三维点。
- **来源：** 地形栅格读取，非激光雷达采集实录。

图： 地形高程采样点显示

高度点云来自栅格采样及标定。

本章目录

场景与载具资源开发

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

3.5.1 三维资产集成路线

教材对应：第3.2.2.1节

3.5.2 UE资产管理

教材对应：第3.2.2.2节

3.5.3 载具XML与执行器

教材第3.2.1.2、3.2.2.3–3.2.2.4节

3.5.4 资源验证与复用

教材第3.2.1.2、3.2.3节；验证流程补充

3.5.3 载具XML与执行器：整机与部件

X680由机身和四个旋翼组成



图： Droneyee X680模型

- **入口：** 1.BasicExps/ e1_CusLoadDroneyeeX680。
- 模板：解压包含FBX、XML。
- 运行资产另按版本核对。
- 组织：body是机身。
- ActuatorList登记部件。
- **两路：** XML列静态部件；或索引蓝图，以事件驱动。

资源几何与运动配置共同构成载具外观。

3.5.3 载具XML与执行器：实际索引公式

Unique3DClassID按最新XML字段计算

Unique3DClassID = DisplayOrder * 100000 + ClassID

X680: 1015 * 100000 + 3 = 101500003

核对：加载XML后按F2，在ModelData.csv中查101500003。
调用：vehicleType填样式索引，copterID另选场景实例编号。
复用：同一样式创建ID100和101；只更新100时，101的位置应保持。

样式索引选资源，实例ID选场景对象。

3.5.3 载具XML与执行器：配置结构

XML把模型类别、显示排序与资产路径连起来

- ClassID与DisplayOrder：共同决定样式索引。
- body中的ModelType与MeshPath：选择加载方式和资产位置。
- Actuator节点：静态网格体路线写部件位置、轴向与运动模式。
- 例：先查ClassID和DisplayOrder，再核MeshPath是否存在。
- 验收：启动扫描后能按样式索引创建所选模型。

配置文件需放在三维端可扫描的Content目录。

3.5.3 载具XML与执行器：样式与实例

一种样式可创建多个独立实例

- 注册：启动扫描XML，样式索引进入模型库。
- **创建：**给新实例ID100、101各发样式索引与初始位姿。
- 更新：只给ID100发新位置，ID101应保持原位。
- 判据：只更新ID100后，ID101位置和姿态不变。

多个实例共用一种资源，位置和动作仍按各自ID更新。

3.5.3 载具XML与执行器：X680机身配置

机身XML定位资产并确定显示基准

```
<ClassID>3</ClassID>
<DisplayOrder>1015</DisplayOrder>
<ModelType>0</ModelType>
<MeshPath>/Game/DroneyeeX680/DroneyeeX680Body</MeshPath>
```

机身： MeshPath从/Game定位Content中的静态网格。
部件： ActuatorList逐项列出旋翼，再定义相对位置与转轴。
范围： 外观缩放改变显示尺寸；质量、惯量和推力仍由动力学模型定义。

ModelType选择加载方式，MeshPath定位资源。

3.5.3 载具XML与执行器：执行器位置与轴

一个旋翼需要位置、转轴和运动模式

```
<RelativePosToBodyCm>
  <x>25.4260</x><y>24.0760</y><z>11.2220</z>
</RelativePosToBodyCm>
<RotationModeSpinOrDefect>0</RotationModeSpinOrDefect>
```

转轴：RotationAxisVectorToBody=[0,0,1]，绕局部Z轴。
模式0：输入为转/分；位置字段单位为厘米。

三个字段分别回答在哪里、绕哪根轴、怎样运动。

3.5.3 载具XML与执行器：运动模式与单位

XML模式决定输入数值的含义

XML模式	输入单位	动作
0	转/分（rpm）	按转速连续自转
1	度（deg）	到给定角度偏转
2	静止模式	保持部件静止
3	厘米（cm）	按给定位移平移

XML模式0： 60 rpm = 360 deg/s。

本例蓝图： 60 deg/s = 10 rpm，速度由蓝图事件定义。

同一输入值在不同模式下表示不同物理量。

3.5.3 载具XML与执行器：输入与可见运动

控制输入经部件配置产生旋翼转动



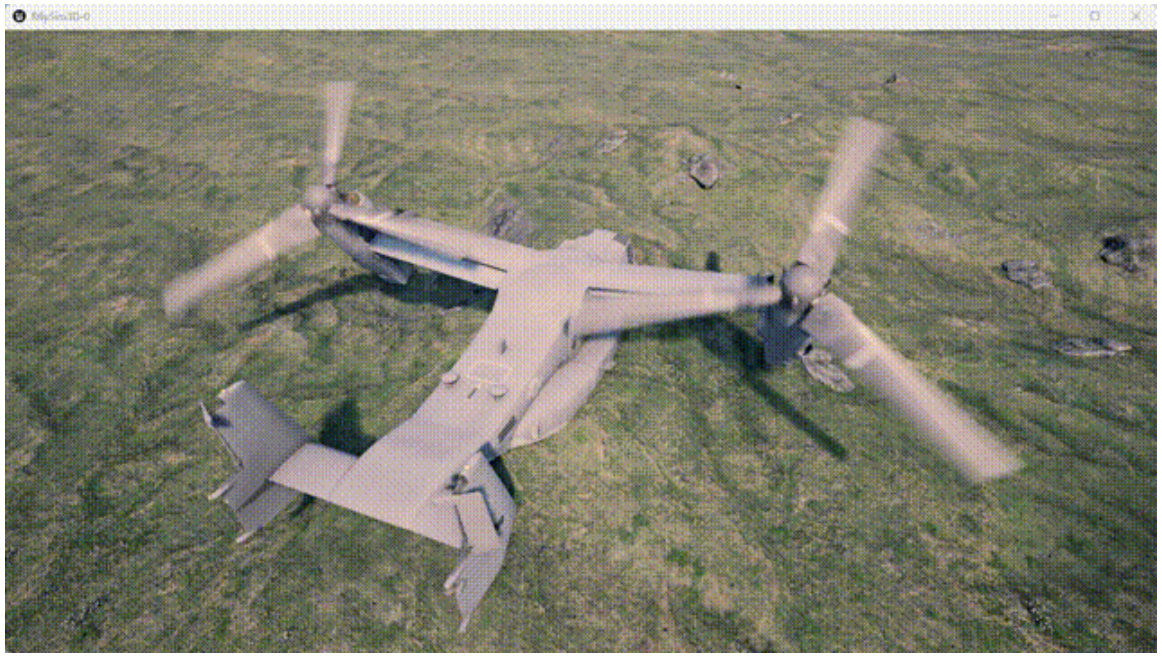
图： 输入驱动显示模型的旋翼

- **通道：** 数值进入指定执行器。
- **配置：** 列表顺序、旋转轴与模式决定动作。
- **结果：** 指定旋翼持续转动。

输入通道须与部件、转轴和模式配对，旋翼才按预期转动。

3.5.3 载具XML与执行器：父子执行器

倾转电机改变子旋翼的空间姿态



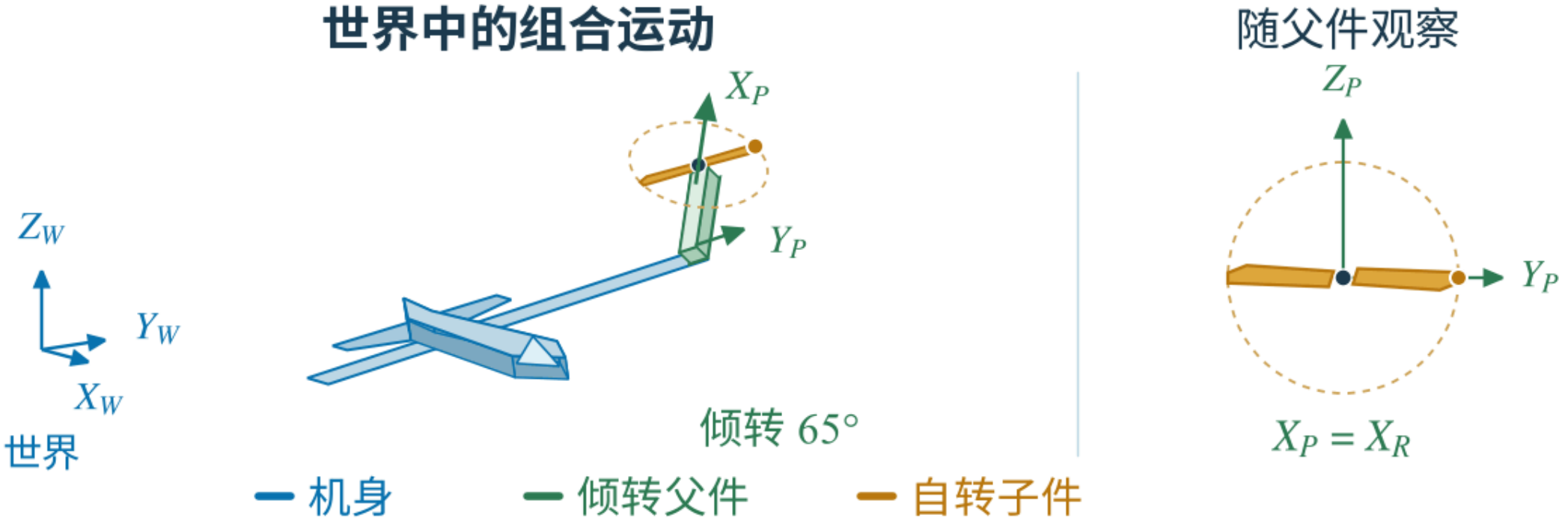
图： 倾转电机与子旋翼同时运动

- **父项2：** 右侧电机，绕Y轴偏转。
- **子项4：** 右侧螺旋桨，绑定父项2并绕X轴自转。
- **变换：** 父件姿态带动子件的局部坐标。

父件倾转与子件自转可以同时发生。

3.5.3 载具XML与执行器：父子执行器

父件倾转后，子旋翼仍绕自己的轴转动



图：电机绕父轴倾转，旋翼绕随父件变化的局部轴自转

3.5.3 载具XML与执行器：父子配置与通道

XML父项序号与Python数组下标对应

<RotationModeSpinOrDefect>0</RotationModeSpinOrDefect>
<AttatchToOtherActuator>2</AttatchToOtherActuator>

lis = [0,0,0,100,0,0,0,100]
lis[1] = i

编号：XML序号从1开始、Python下标从0开始；第2项→lis[1]，第4项→lis[3]。

父件：右电机是第2项；lis[1]=i输入角度，子桨绑定父项2。

子件：右桨是第4项；lis[3]=100保持100转/分，父件倾转时子件仍自转。

3.5.3 载具XML与执行器：绑定实验入口

对照父件倾转与子件自转

- **入口：**0.ApiExps/e4_UAVCtrl/1.ActuatorBinding；模板QZXY.zip，脚本ActuatorBindingDemo.py。
- **准备：**先创建实例1000；在脚本副本中固定其他角度项，只保留lis[1]=i。
- **对照：**先固定lis[3]=100改变右电机角度；再固定lis[1]，只改变lis[3]的右桨转速。

创建：按F2查询QZXY样式索引，用sendUE4Pos创建ID1000；脚本只发部件命令。

分别改变父角度与子转速，观察两种运动叠加。

3.5.3 载具XML与执行器：盒体例程

盒体例程：从建蓝图到看转速的完整步骤

安装根目录：[安装目录]\RflySimAPIs\3.RflySim3DUE
相对路径：0.ApiExps\e0_DevToolsUsage\1.UEBlueprintUsage

建蓝图：按本目录Readme.pdf在UE4.27.2中新建空白项目；以Actor为父类建蓝图，添加StaticMesh并选Cube，编译后拖入关卡。
写节点：Event Tick接AddActorLocalRotation，Delta Rotation=(0,0,1)；点击Play运行。

判据：盒体绕自身Z轴持续转动，每秒转角等于帧率乘1°；帧率120时约120°/s。

常见问题：Static Mesh里找不到Cube时，在内容浏览器设置中勾选“显示引擎内容”。

盒体每帧固定转1°，所以同一段时间的转角随帧率变化。

3.5.3 载具XML与执行器：蓝图资源索引

蓝图路线把动作逻辑封装在资产中

```
<ClassID>3</ClassID>
<DisplayOrder>1100</DisplayOrder>
<ModelType>2</ModelType>
<MeshPath>
  /Game/Droneyee_WithBP/DroneyeeX680_BP
</MeshPath>
```

样式索引：1100×100000+3=110000003（模板原值）；ModelType=2加载蓝图。部署前查重；若已占用，在资源副本改DisplayOrder并重算。

XML定位蓝图，蓝图事件把输入映射到组件。

3.5.3 载具XML与执行器：蓝图入门·循环与调试

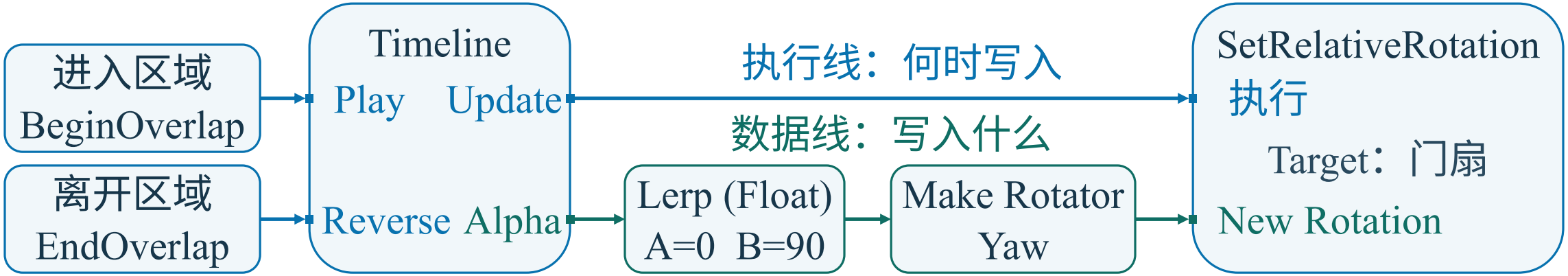
循环逐项处理对象，断点核对执行路径

- **场景：** DoorLeaves数组保存3个有效门扇组件；BeginPlay时统一关门。
- **循环：** BeginPlay接ForEachLoop；DoorLeaves接Array。
  Loop Body接SetRelativeRotation，Array Element接Target，三轴转角设0。
- **完成：** Completed接Print String，仅输出一次“初始化完成”。
  循环批量处理有限对象；它本身不提供跨帧动画进度。
- **断点：** PIE（编辑器内运行）选定当前蓝图实例，在SetRelativeRotation加断点；
  单步跟随执行路径高亮，检查是否到达该节点。
- **观测：** Watch this value查看Array Index和Array Element；索引为0、1、2，
  引用应逐项改变。未命中查事件和执行连线；命中却不动查Target与转角。

3.5.3 载具XML与执行器：蓝图入门·时间轴与插值

时间轴给出进度，插值确定门扇转角

- **题设：**门扇可移动，枢轴位于竖直铰链；局部Yaw从0°转到90°，其余两轴为0。触发区只响应一个指定对象。
- **时间轴：**2 s线性轨道，Alpha从0到1；关闭Loop，正向开门、反向关门。



- **解释：**播放1 s时Alpha为0.5；Lerp算出45°，Update触发写入门扇。

$$0° + (90° - 0°) \times 0.5 = 45°$$

3.5.3 载具XML与执行器：蓝图事件链

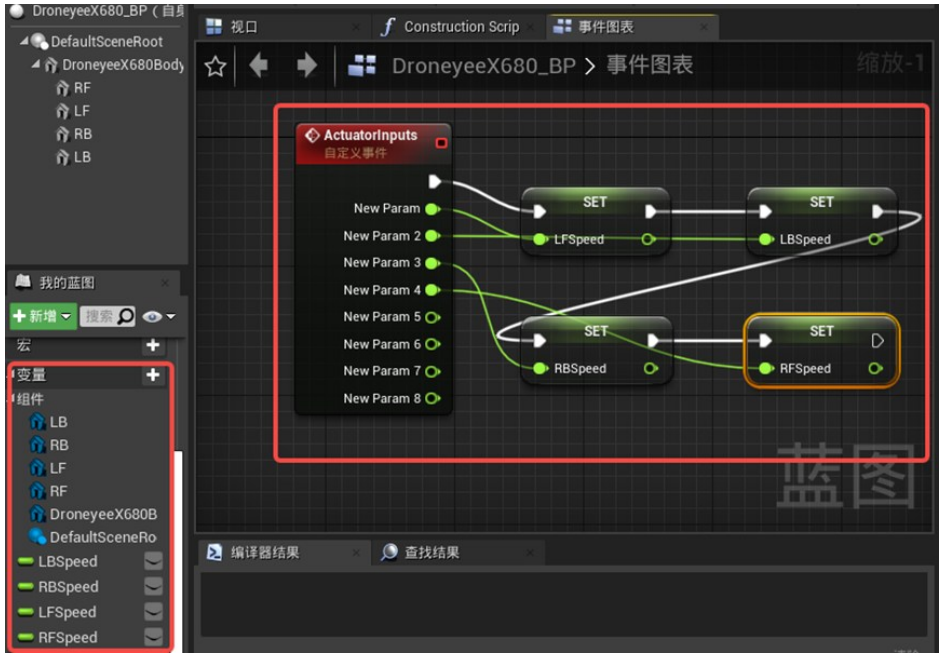
输入事件存变量，Tick逐帧驱动组件

- 接收：ActuatorInputs把各通道输入写入对应变量。
- **计算：**Tick取帧间隔，按每个变量算角增量。
- 执行：更新指定组件的局部转角；单通道置零可查绑定。
- 例：四路速度为60、80、100、120度/秒，逐路绑定旋翼。
- 判据：单路输入置零时，仅该路角增量归零。

蓝图把复杂部件层级和动作逻辑封装在模型资产内。

3.5.3 载具XML与执行器：输入事件

让输入通道与旋翼组件一一对应



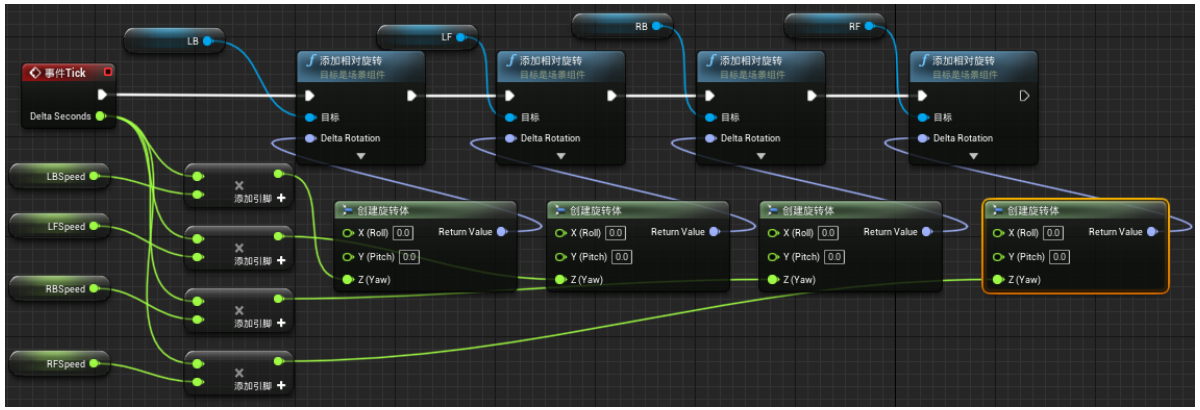
图： ActuatorInputs事件与变量存储

- **对应关系：** 前四项输入分别进入四个旋翼速度变量。
- **单通道验证：** 其余项置零，只改变一项，观察哪个组件转动。
- **资源改装：** 更换机型时同时核对变量、组件和通道次序。

逐项输入与组件动作对照，可以定位通道次序或部件绑定错误。

3.5.3 载具XML与执行器：逐帧旋转

核对蓝图的目标组件、转轴与单位



图：四个旋翼的完整Tick动作链

- **目标组件：**各速度变量进入同名旋翼支路，LB支路最终更新LB组件。
- **局部转轴：**图中角增量接入Yaw，绕该组件的局部Z轴旋转。
- **量纲核对：**本例60度/秒×0.02秒=1.2度；XML模式0的60转/分则是360度/秒。

同一个输入数字，只有结合资源的通道、转轴与单位才确定动作。

3.5.3 载具XML与执行器：真实蓝图调用

一条命令分别设置四个旋翼速度

RflySetActuatorPWMs 1000 60 80 100 120 0 0 0 0

60 * 0.02 = 1.2

前提：按F2核验X680蓝图的实际样式索引，再创建实例1000。

输入：前四项为60/80/100/120度每秒，后四项未用于本例旋翼速度。

题设：帧间隔0.02秒时，四项角增量为1.2/1.6/2.0/2.4度。

停止：把前四项置0，新的Tick角增量为0。

3.5.3 载具XML与执行器：X680例程

X680例程：用控制台命令让四个旋翼按指定速度转

安装根目录：[安装目录]\RflySimAPIs\3.RflySim3DUE

相对路径：2.AdvExps\e2_BlueprintModel\1.BlueprintBuild

部署：解压1.BlueprintBuild.zip及其中的Droneyee_WithBP_cooked_UE4.zip，按本单元索引规则查重后再运行Copy.bat部署。

创建载具：打开RflySim3D，双击地面后按O和3创建四旋翼，按C切换为DroneyeeX680_WithBP；按S可见编号从1000起。

发命令：按~键打开控制台，输入RflySetActuatorPWMs 1000 60 80 100 120 0 0 0 0 0

判据：LF、LB、RB、RF约以60、80、100、120°/s转动，与帧率无关；以LB为例，60FPS时每帧约$80/60 \approx 1.33°$。

外部命令只送来角速度，旋翼每帧转多少由蓝图按帧间隔算出。

3.5.3 载具XML与执行器：蓝图逐帧更新

Tick把角速度乘帧间隔，再累加到当前角度

- 输入：本例四路速度依次为60、80、100、120度每秒。
- **一帧：**0.02秒内四个角增量依次为1.2、1.6、2.0、2.4度。
- 运行：每帧累加；某路输入置零，该旋翼停在当前角度。

$$\Delta\theta = \omega\Delta t$$

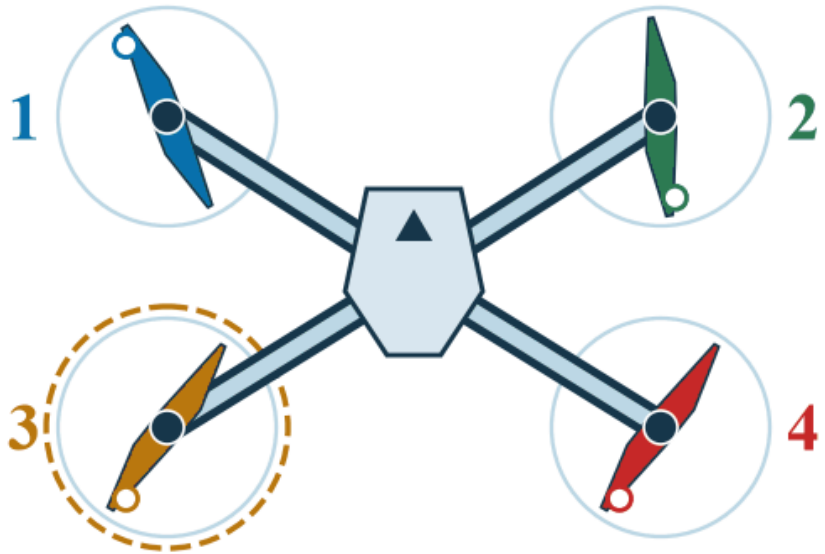
输入单位先由蓝图定义，不能套用XML模式0的转速单位。

3.5.3 载具XML与执行器：Tick过程示意

四路输入在相同帧间隔下累加不同角度

$\Delta\theta = \omega \times 0.02\ \mathrm{s}$

第 400 步　模型时间 8.00 s



组件局部角度

通道	ω (°/s)	Δθ (°)	累计角 (°)
1	60	1.2	480.0
2	80	1.6	640.0
3	**0**	**0.0**	**600.0**
4	120	2.4	960.0

第3路归零，保持当前角度

图：蓝图Tick逐帧累加角度（原创教学示意）

3.5.3 载具XML与执行器：扩展输入与版本

扩展事件把输入映射为旋翼偏移

```
cmd = 'RflySetActuatorPWMsExt 1000 0 0 20 ' + '0 '*13
ue.sendUE4Cmd(cmd)
```

事件： ActuatorInputsExt接收本例16个浮点值。
映射： 前3项为XYZ偏移；[0,0,20]使四个旋翼局部Z增加20厘米。
版本： 完整版；UE4.27.2成品或面向UE5.8重新Cook的资产。

ActuatorInputsExt按本例蓝图定义XYZ偏移。

3.5.3 载具XML与执行器：蓝图应用

复杂执行器、命中事件和特效分别建模

- 复杂部件：起落架、舵面与倾转机构用层级和事件组织。
- **交互事件：**命中检测改变模型状态，记录触发条件。
- 视觉特效：粒子或材质表现爆炸、烟雾；不代替损伤模型。
- 触发：按实例ID和命中组件定位对象，记录事件时刻。
- 验收：视觉特效可复现；损伤量须另有模型输出。

动作入口按资产蓝图定义，不能把视觉效果当物理结果。

3.5.3 载具XML与执行器：XML与蓝图选型

同一架载具有两种动作组织方式

- **XML静态网格：** ActuatorList组织部件；位置、轴向、父项与模式可直接读写。
- **蓝图资产：** ActuatorInputs接收输入，变量与Tick组织动作；XML索引该蓝图。
- **选择：** 规则部件先用现成XML；定制事件采用对应蓝图，并沿用其输入单位。

入口： 2.AdvExps/e2_BlueprintModel/1.BlueprintBuild。

UE4部署： 解压1.BlueprintBuild.zip，再解压Droneyee_WithBP_cooked_UE4.zip；查重后将Droneyee_WithBP复制到RflySim3D/RflySim3D/Content。

规则部件可用XML，定制动作可封装为蓝图事件。

本章目录

场景与载具资源开发

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

3.5.1 三维资产集成路线

教材对应：第3.2.2.1节

3.5.2 UE资产管理

教材对应：第3.2.2.2节

3.5.3 载具XML与执行器

教材第3.2.1.2、3.2.2.3–3.2.2.4节

3.5.4 资源验证与复用

教材第3.2.1.2、3.2.3节；验证流程补充

3.5.4 资源验证与复用：多类载具

不同载具复用相同的资源接口



(b) 载具模型

图3.8(b) 载具模型

- **共用：**样式索引、实例ID、位置姿态。
- **差异：**旋翼自转、舵面偏转、倾转层级。
- **配套：**载具动力学由相应模型提供。

外观资源可复用，部件机制仍按载具定义。

3.5.4 资源验证与复用：地理场景联动

大场景把地理资源与仿真状态配合



(c) 大场景模型

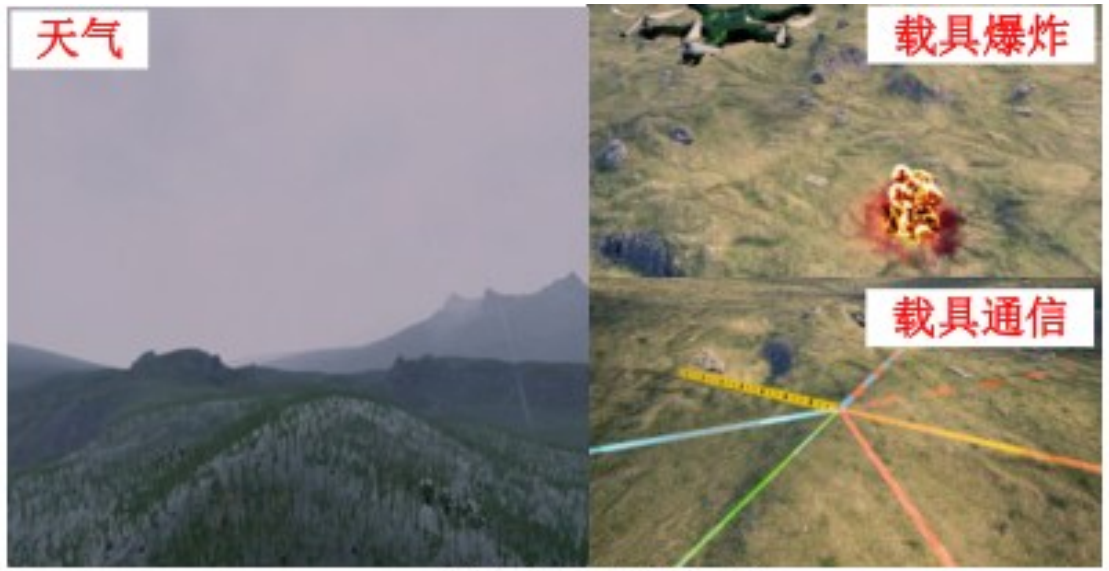
图3.8(c) 大场景模型

- **环境：** 香港三维重建提供地理几何。
- **坐标：** 地图原点连接全球位置与局部仿真状态。
- **联动：** 同一载具状态分别送往地面站和视景。

地图提供环境，状态链路驱动载具和地面站。

3.5.4 资源验证与复用：效果与数据连接

天气、爆炸和通信线表达不同事件



(d) 常用特效模型

图3.8(d) 常用特效模型

- **天气：** 参数改变环境外观。
- **爆炸：** 事件触发粒子或动画。
- **通信：** 连线表示关系；统计值由通信模型提供。

可视效果由参数或事件驱动，统计量由数据模型提供。

3.5.4 资源验证与复用：排错链条

从缺失对象定位到具体依赖

- **未注册：**XML位置、ClassID与DisplayOrder决定可查到的样式条目。
- **已注册但缺几何/贴图：**MeshPath、目标版本和Cooked依赖决定加载结果。
- **几何存在但部件不动：**实例ID、输入通道、模式或蓝图事件决定动作。

复用验证：创建并改位姿，退出后重新加载同一资源再创建；几何、材质和动作应与所存配置一致。

按注册、路径和状态三个阶段定位加载问题。

3.5.4 资源验证与复用：资源交付与练习

交付一份能解释的资源包

- **基础任务：**选一条导入路线，交付资源文件、配置、样式索引和实例创建调用。
- **机制题：**解释XML模式0的60与本例蓝图输入60为何产生不同转速。
- **对照题：**固定子转速改变父角度，记录父子动作；再创建第二实例独立更新。

提交：分别写出两种输入60的单位、换算后的角速度，并按实例ID说明第二个对象是否随第一个对象更新。

资源、配置与调用对齐后，动作才能被稳定复用。

练习参考要点

用索引、单位和实例ID解释三类常见故障

缺模型：

先查XML是否被启动扫描，再核ClassID、DisplayOrder、MeshPath与Cook依赖。

转速错：

XML模式0：60转/分=360度/秒。

本例蓝图：输入60=60度/秒。

串对象：

给实例100发位姿只应更新100，实例101的位置和姿态保持不变。

3.5.4 资源验证与复用：学习资料

查XML字段、样式索引和资源类别

XML字段、运动模式与样式索引

https://rflysim.com/doc/zh/4.DevRef/3d-import.html

已有场景与模型资源类别

https://rflysim.com/doc/zh/4.DevRef/3d-res.html

按具体学习任务查文档，再回到所选模型。

本节小结

先选资源路线，再按对应接口加载与验证

- **UE编辑器：**核资产及依赖，Cook后部署；载具按需登记XML。
- **GLB运行时：**核解析插件、模型与贴图，按仓库条目创建实例。
- **GIS地理数据：**核瓦片格式、坐标基准、覆盖范围及服务可达性。
- **共同设置：**核单位、坐标轴、原点与外观，记录程序和资源版本。
- **运行验证：**重启复用；载具按实例ID检查位姿、部件与碰撞。

加载成功只是起点，资源还要在目标任务中可靠复用。