

绕过UE编辑器直接导入三维模型到RflySimUE5

1. 实验目的

绕过UE编辑器直接导入三维模型到RflySimUE5

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\1.BasicExps\e4_CusLoadObj\2.loadSimpleMesh](#)

例程介绍

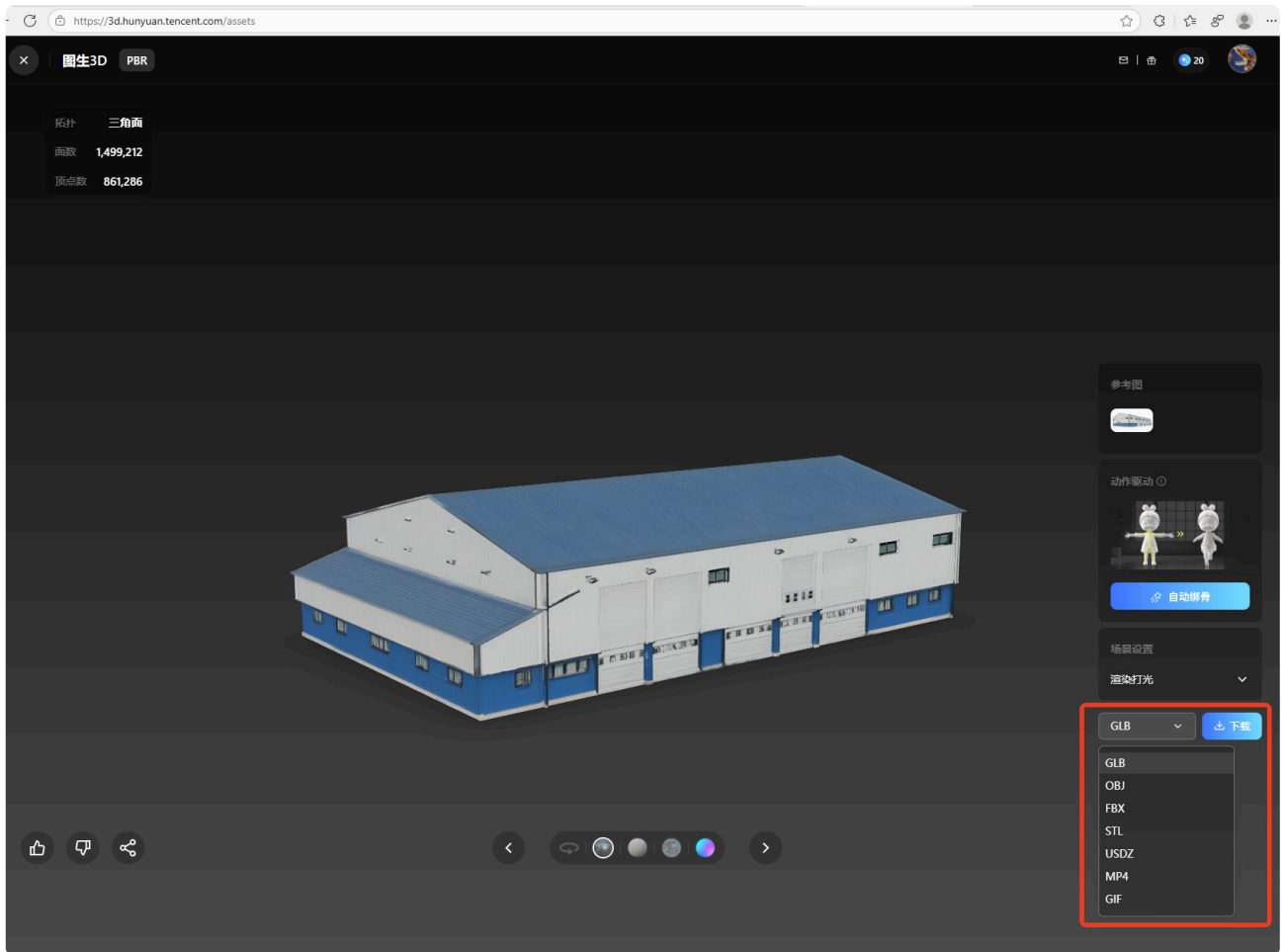
- `./Copymodel.bat`：将 `smith` 目录复制到 `[RflySim安装目录]\RflySimUE5\RflySim3D\Content` 的批处理脚本,用于导入蓝图和XML配置
- `./loadmodelUE5.bat`：启动RflySimUE5并加载特定地图与Datasmith Runtime蓝图的批处理脚本,运行后会弹出GLB文件选择窗口
- `./UEImportScript.py`：示例Python脚本(原始版本,位于例程根目录),演示如何在RflySim3D中生成ClassID=901的模型对象
- `./smith/`：自定义模型资源描述目录,包含 `smith.xml` 和UE蓝图资源(由 `Copymodel.bat` 复制到RflySim工程中)
- `./smith/smith.xml`：自定义模型描述文件,将ClassID=901映射到 `/Game/smith/NewBlueprint` 蓝图,并配置多个机载相机视角
- `./smith/NewBlueprint.uasset`：UE5蓝图资源文件,内部封装了Datasmith Runtime导入的模型呈现逻辑
- `./smith/NewBlueprint.uexp`：UE5蓝图数据文件,配合 `uasset` 使用

4. 实验内容或步骤

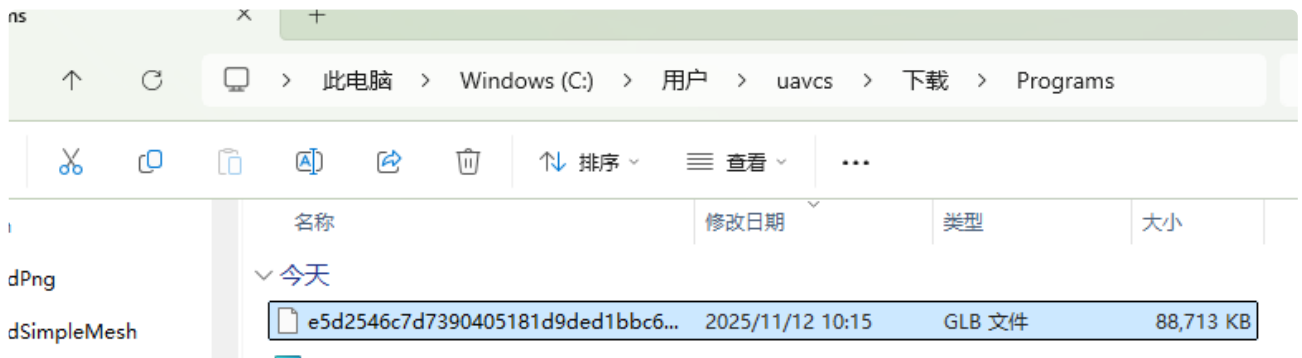
4.1 步骤1: 准备GLB三维模型

本实验导入的模型主要用做场景中的障碍物，因此可以直接用AI工具生成，比如[腾讯混元3D](#)

可以看到常见的三维格式均可下载，这里选择GLB格式下载，因为它是完全的二进制格式，datasmith runtime插件可以直接识别这种格式



例如我这里将gltf模型下载到如下目录



4.2 步骤2: 复制Datasmith蓝图到RflySimUE5工程

2.1 在文件资源管理器中进入本例程目录:

[安装目录]\RflySimAPIs\3.RflySim3DUE\1.BasicExps\e4_CusLoadObj\2.loadSimpleMesh

```
1 | [RflySim安装路径]\3.RflySim3DUE\1.BasicExps\e4_CusLoadObj\2.loadSimpleMesh
```

2.2 双击运行 **Copymodel.bat**, 该脚本将本目录下的 **smith** 文件夹复制到:

```
1 | [RflySim安装目录]\RflySimUE5\RflySim3D\Content
```

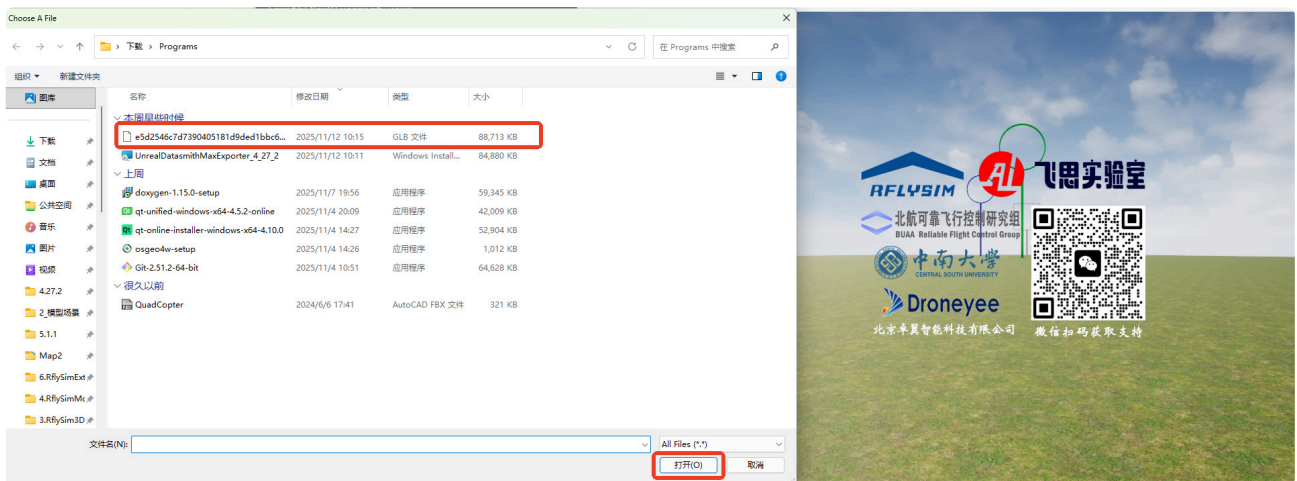
此操作会将:

- **smith.xml**
- **NewBlueprint.uasset / NewBlueprint.uexp**
等文件导入到RflySimUE5工程中,使其能够识别ClassID=901对应的自定义模型类型。

验证检查点: 打开 [RflySim安装目录]\RflySimUE5\RflySim3D\Content 目录,确认存在 **smith** 子文件夹及其中的 **smith.xml** 和蓝图资源文件。

4.3 步骤3: 启动RflySimUE5并加载Datasmith Runtime蓝图

双击运行 **loadmodelUE5.bat** 启动RflySimUE5并切换到特定地图, 然后会加载上一步导入的蓝图, 此时在文件管理器弹窗中选择之前下载的glb模型



此时场景中会加载出glb模型, 拖动视角到可以观察到的位置



5. 关键知识点

关键知识点1: Datasmith Runtime运行时导入机制概述

Datasmith Runtime 是虚幻引擎提供的一组在运行时(即打包后的应用程序中)可用的导入接口,可以在不依赖编辑器工作流的前提下,加载外部描述文件(`.udatasmith`)及其关联的几何体、材质和层级结构。在RflySimUE5场景中,它被用来实现"在仿真运行过程中动态加载外部三维模型"的能力。

在本例程中,核心思路是:

1. 在RflySimUE5工程中预先配置好一个 `DatasmithRuntimeActor` 蓝图,负责:
 - 弹出文件选择窗口
 - 通过Datasmith Runtime节点加载选定的GLB/UDatasmith内容
 - 将导入的模型挂接到固定的场景锚点
2. 利用RflySim提供的资源拷贝脚本(`Copymodel.bat`),将该蓝图及配套的XML描述(`smith.xml`)复制到RflySimUE5运行目录的 `Content` 下。
3. 用户只需在运行时选取GLB模型,即可在RflySimUE5场景中看到对应的三维物体,实现"绕过UE编辑器导入模型"的效果。

与传统"在UE编辑器中导入FBX/GLTF再打包"的流程相比,Datasmith Runtime的优势在于:

- **无需重新打包工程:** 新模型可在运行时直接导入
- **支持用户自助导入:** 非开发人员也可通过简单界面选择GLB文件

- **保持场景层级与元数据:** 可访问导入对象的Actor层级和属性,方便后续控制

I 关键知识点2: Datasmith Runtime蓝图节点工作流程

在 [datasmithruntime绕过UE编辑器加载gltf模型.md](#) 中,已简要介绍了常见的Datasmith Runtime蓝图节点。典型的运行时导入蓝图大致包含以下几个关键节点:

1. 构造Datasmith运行时导入选项(Construct Datasmith Runtime Import Options)

用于设置导入层级、碰撞方式、元数据导入等选项,例如:

- 构建层级: 简化(Simplified) / 未筛选(Unfiltered)
- 碰撞类型: 用于静态障碍物时一般选择"启用碰撞(查询和物理)"
- 是否导入元数据: 需要对模型做高层次语义处理时可以启用

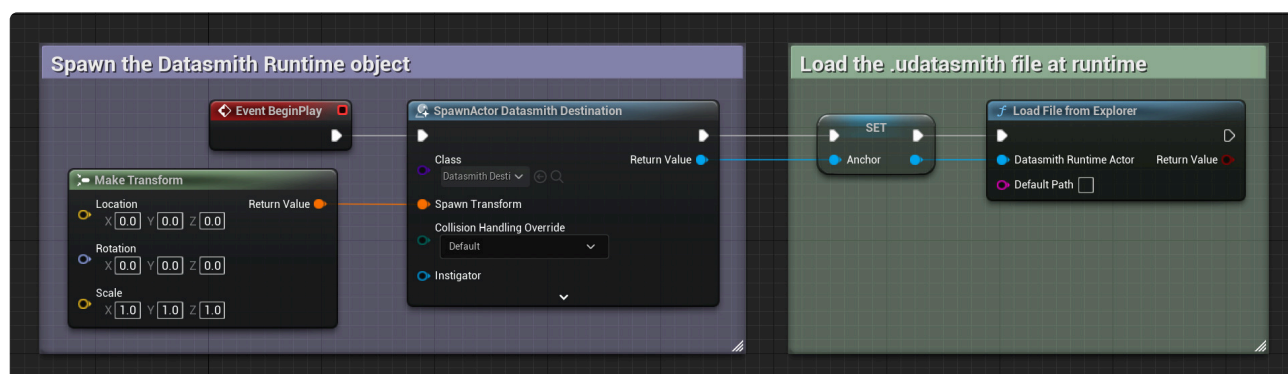
2. 设置导入选项(Set Import Options)

将上一步构造的导入选项应用到 `DatasmithRuntimeActor` 实例上,确保后续导入过程使用统一参数。

3. 从浏览器加载文件(Load File from Explorer)

在运行时弹出文件选择对话框,用户选择 `.udatasmith` 或直接支持的GLB文件后,由Datasmith Runtime完成解析和加载,并在场景中生成相应的静态网格体及材质。

整体蓝图逻辑常见模式如下:



1. 在关卡 `BeginPlay` 中 `Spawn Actor From Class` 生成一个 `DatasmithRuntimeActor` 锚点;
2. 将其保存为变量(如 `Anchor`);
3. 在适当时机(如 `BeginPlay` 或按键事件触发)调用 `Load File from Explorer`,将 `Anchor` 作为目标输入;
4. Datasmith Runtime完成模型加载,并将其放置在锚点的 `Transform` 之下。

在RflySimUE5中,这些蓝图逻辑已经封装到示例项目内,用户只需将相关资源复制到RflySim运行目录即可使用。

I 关键知识点3: smith.xml配置文件与ClassID映射

`smith/smith.xml` 是RflySim用来描述自定义三维对象(在本例中是运行时导入的GLB模型)的配置文件。其核心结构如下:

```

1  <?xml version="1.0"?>
2  <vehicle>
3      <ClassID>901</ClassID>
4      <DisplayOrder>1000</DisplayOrder>
5      <Name>smith</Name>
6      <Scale>
7          <x>1</x>
8          <y>1</y>
9          <z>1</z>
10     </Scale>
11     <AngEulerDeg>
12         <roll>0</roll>
13         <pitch>2.7</pitch>
14         <yaw>0</yaw>
15     </AngEulerDeg>
16     <body>
17         <ModelType>2</ModelType>
18         <MeshPath>/Game/smith/NewBlueprint</MeshPath>
19         <MaterialPath></MaterialPath>
20         <AnimationPath></AnimationPath>
21         <CenterHeightAboveGroundCm>1000</CenterHeightAboveGroundCm>
22         <NumberHeigthAboveCenterCm>2000</NumberHeigthAboveCenterCm>
23         <NumberSizeScale>0</NumberSizeScale>
24     </body>
25     <OnboardCameras>
26         <!-- 多个机载相机配置,用于从不同视角观察模型 -->
27     </OnboardCameras>
28 </vehicle>

```

关键字段说明:

- **ClassID**: 自定义对象的类型编号。本例中为901,后续在Python脚本中会用此编号作为 **VehicleType**,以生成对应模型实例。
- **MeshPath**: 指向UE内容浏览器中的蓝图路径 **/Game/smith/NewBlueprint**,该蓝图内部通过 **Datasmith Runtime**逻辑引用运行时导入的GLB内容。
- **ModelType**: 模型类型标志,在RflySim内部用于区分不同渲染/物理模式。
- **Scale** 与 **AngEulerDeg**: 用于对模型整体进行缩放和初始姿态调整。
- **OnboardCameras**: 配置了多个机载相机位置与姿态,方便从不同视角对导入的三维模型进行观察和调试。

通过这种方式,**smith.xml**建立了**"RflySim内部ClassID ⇔ UE5蓝图 ⇔ 运行时GLB模型"**之间的映射关系,使得后续只需通过 **sendUE4Pos** 指定 **VehicleType=901** 即可在场景中生成该模型。

I 关键知识点4: UEImportScript.py代码解析

本例程中,Python脚本 **UEImportScript.py**展示了如何在RflySim3D中生成一个基于GLB模型的三维对象。代码位于 **UEImportScript.py**:

```

1  import time
2  import math
3
4  import UE4CtrlAPI as UE4CtrlAPI
5
6  # 创建RflySim3D控制接口
7  ue = UE4CtrlAPI.UE4CtrlAPI()
8
9  # 切换到包含障碍物场景的地图(例如 VisionRing)
10 ue.sendUE4Cmd('RflyChangeMapbyName VisionRing')
11
12 time.sleep(3) # 等待地图加载完成
13
14 # 通过XML中配置的ClassID=901加载运行时导入的三维模型
15 # sendUE4Pos(CopterID, VehicleType, RotorSpeed, PosM, AngEulerRad)
16 ue.sendUE4Pos(
17     100,      # CopterID: 对象ID,避免与无人机ID冲突
18     901,      # VehicleType: 在smith.xml中定义的ClassID,对应运行时加载的glb模型
19     0,        # RotorSpeed: 对于静态障碍物设为0
20     [2.5, 0, 0],# PosM: 物体在世界坐标中的位置(米)
21     [0, 0, math.pi] # AngEulerRad: 姿态欧拉角(弧度),这里绕Z轴旋转180°
22 )

```

整体框架

1. 初始化RflySim控制接口(`UE4CtrlAPI.UE4CtrlAPI()`)
2. 通过文本命令切换RflySim场景地图
3. 延时等待地图加载
4. 使用 `sendUE4Pos` 根据 `ClassID=901` 在场景中生成自定义三维模型实例

关键函数解析

1. `ue.sendUE4Cmd('RflyChangeMapbyName VisionRing')`
 - 向RflySim发送控制命令,切换当前场景到名为 `VisionRing` 的预设地图
 - 该地图通常包含适合障碍物测试的飞行环境
2. `ue.sendUE4Pos(CopterID, VehicleType, RotorSpeed, PosM, AngEulerRad)`
 - `CopterID=100` : 为当前对象分配唯一ID,避免与无人机等其他实体冲突
 - `VehicleType=901` : 对应 `smith.xml` 中的 `ClassID` ,RflySim会根据此编号加载 `/Game/smith/NewBlueprint` 蓝图
 - `RotorSpeed=0` : 对于静态障碍物无需旋翼转速,设为0
 - `PosM=[2.5,0,0]` : 对象在世界坐标中的位置(米),可根据需要调整
 - `AngEulerRad=[0,0,math.pi]` : 以弧度表示的欧拉角姿态,此处绕Z轴旋转180°,使模型朝向相机或飞行路径

运行结果

执行脚本后,RflySim3D会在指定位置生成一个基于用户所选GLB模型的三维障碍物,其外观和尺寸由运行时导入的资源决定,物理位置和姿态由脚本控制

6. 参考资料

1. [RflySim官方文档](#)
2. [RflySim UE4CtrlAPI示例](#)

7. 常见问题

Q1: 运行 `loadmodelUE5.bat` 后没有弹出GLB文件选择窗口

A1: 可能原因包括RflySimUE5版本不匹配或Datasmith Runtime插件未正确启用。请检查:

1. 当前使用的RflySimUE5版本是否支持Datasmith Runtime;
2. `Copymodel.bat` 是否成功将 `smith` 目录复制到RflySimUE5工程 `Content` 下;
3. 如有源工程,请确认其中 `Datasmith Runtime` 相关插件已启用,并重新生成示例蓝图。

Q2: 选择GLB文件后模型未显示或显示位置异常

A2: 请检查以下几点:

1. GLB文件是否损坏或包含不支持的几何/材质特性,可用第三方工具预览;
2. 模型可能被导入到较远位置或缩放不合适,尝试在RflySim中调整相机位置或修改蓝图中的 `Transform`;
3. 检查 `smith.xml` 中的 `Scale` 设置,必要时适当放大或缩小模型比例。

Q3: 运行 `UEImportScript.py` 时报

`ModuleNotFoundError: No module named 'UE4CtrlAPI'`

A3: 说明当前Python解释器未能找到RflySim提供的 `UE4CtrlAPI` 库。解决方法:

1. 使用RflySim安装目录中自带的Python环境运行脚本;
2. 或将 `UE4CtrlAPI` 所在路径添加到 `PYTHONPATH` 环境变量;
3. 确认RflySim工具链安装完整且未被移动或删除关键文件。

-
1. <https://rflysim.com/> ↩

2. 推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf> ←