

RflySim3D场景元素创建和保存实验

1. 实验目的

本实验旨在让学生在 RflySim3D 工具链中快速构建可重复、可版本化的自定义三维仿真场景，掌握虚拟场景中的障碍物创建和导入方法。完成本实验后，学生应能够：

- 理解并掌握基本的场景要素（地形、静态物体、动态实体、光照、相机等）。
- 了解使用 RflySim3D 平台手动搭建场景的流程。
- 掌握使用脚本化方法（JSON/Python 代码）实现自动化、参数化创建场景的方法。

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

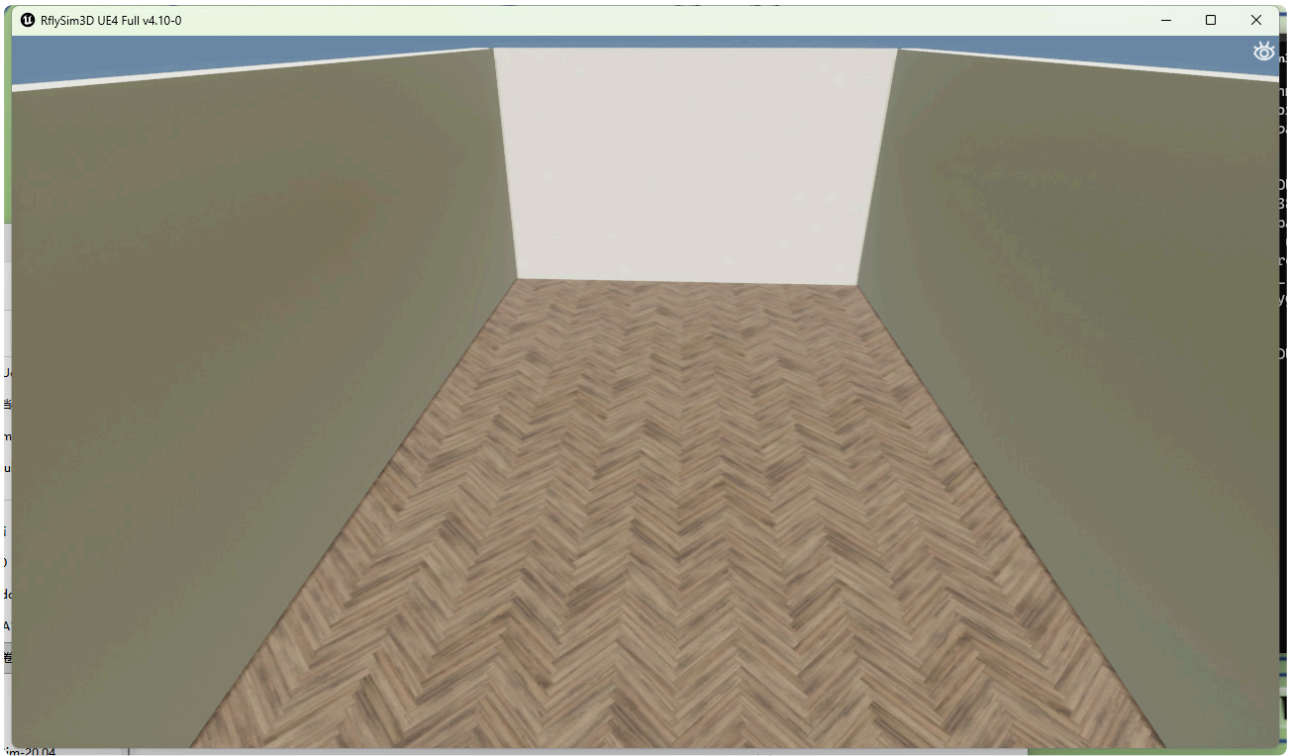
例程目录：

[安装目录]\RflySimAPIs\3.RflySim3DUE\1.BasicExps\e0_StarterContent\5.UE_CustomImport

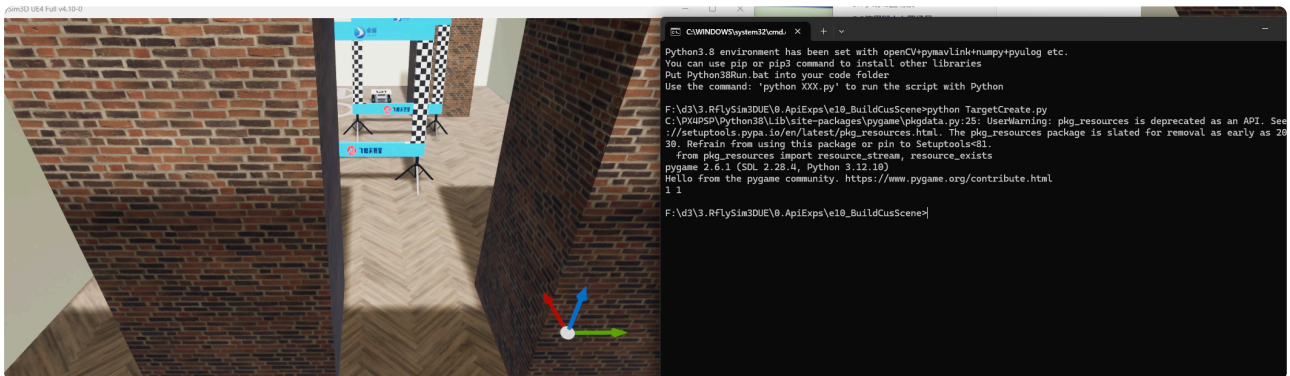
4. 实验步骤

4.1 步骤1：通过 Python 程序自动创建场景元素

重打开 RflySim3D 并切换到 ChallengeMap。



双击运行当前目录下的 `Python38Run.bat` 快捷脚本启动内部 Python 环境。
在弹出的终端中输入并执行命令：`python TargetCreate.py`



观察 RflySim3D 窗口，会发现原本的 ChallengeMap 场景中已经根据代码中设定的规则（特定坐标和随机偏移），瞬间生成了一系列排列好的障碍物和元素。

注意：关于 `TargetCreate.py` 脚本中自动创建场景元素的详细代码原理与核心接口，请参阅后文关键知识点“5.4 基于 Python 的场景元素创建原理”。

关于利用 RflySim3D 内置的模型库手动布置自定义场景的详细操作，可以参考例程：[\[RflySim安装目录\]\RflySimAPIs\3.RflySim3DUE\0.ApiExps\e10_BuildCusScene](#)

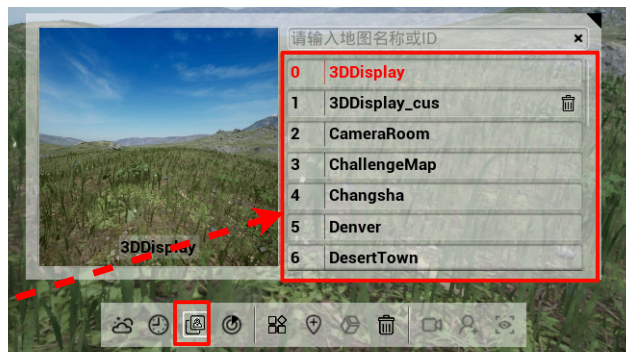
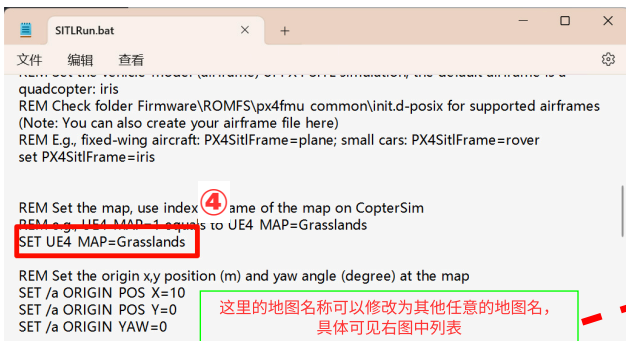
4.2 步骤2：保存和加载布置完成的场景

场景布置完成之后，依次点击保存场景-命名新关卡-保存确认，这一步会在 RflySim3D 特定目录下生成 JSON 文件存储本次创建的 Actor 信息，后续通过解析该 JSON 文件重新加载

此次保存的关卡。



调用地图时，可直接通过修改 bat 脚本中的 `UE_MAP` 变量来调用，此时点击切换地图已经可以看见刚才保存的场景，具体操作如下图。



注意：通过这种方式手动创建的地图会在重新安装工具链时，可能会因为文件的更新或清空而导致场景删除，可将通过这种方式创建的文件单独保存下来，在新版本安装完成后，再将其导入。主要需要保存的文件有：

- `[RflySim安装目录]\RflySim3D\RflySim3D\Saved\AllActorSave` : 场景信息文件 (含 `<自定义的场景名>.json` 和 `<自定义的场景名>.png`)。
- `[RflySim安装目录]\CopterSim\external\map` : 中保存的场景名相同的文件 (`<自定义的场景名>.txt` 和 `<自定义的场景名>.png` 文件)。
- `<自定义的场景名>.json` 文件可直接打开查看添加的载具ID、坐标信息等属性。
- `<自定义的场景名>.txt` 文件存储的数字依次为: 该场景右上角三维坐标点 (`xy`全为正, `z`向上为正)、左下角三维坐标点 (`xy`全为负, `z`向上为正)、场景中任意第3点三维坐标点, 单位均为厘米。前两个点的目的是为了确认地形的范围和中心坐标, 第3点坐标可自行选取, 理论上需要尽量在高度上与前两个点有落差, 用于校正高度尺度。
- `<自定义的场景名>.png` 文件记录的是场景的地势高程信息。

5. 关键知识点

5.1 场景要素与层次结构

- **概念:** 一个可用的三维场景通常包含: 地形 (terrain/ground plane)、静态物体 (buildings/obstacles)、动态实体 (drones/vehicles/agents)、光源 (lights)、相机 (view points) 与物理材质 (碰撞体与可视网格)。在 RflySim3D 中, 这些场景要素统称为 **Actor (场景元素)**。
- **Actor 类型:**
 - **静态 Actor:** 地形、建筑物、障碍物等不会移动的场景元素。
 - **动态 Actor:** 无人机、车辆、可移动物体等具有物理交互的实体。
 - **功能 Actor:** 光源、相机、传感器等提供特定功能的元素。
- **作用:** 每个 Actor 负责仿真中的不同职责。所有 Actor 的属性 (类型、位置、旋转、缩放) 会被保存到 JSON 文件中, 实现场景的持久化存储与重复加载。

5.2 场景保存信息文件解析

- 场景信息文件解析: `<自定义的场景名>.json` 文件包含添加的载具 (场景元素) 类型 ID、坐标与姿态信息等; `<自定义的场景名>.txt` 文件记录地形边界范围及参考坐标, 用于校正高度尺度; `<自定义的场景名>.png` 文件记录场景的地势高程信息。
- 文件保存路径: RflySim 自动保存场景相关的目录分别为 `RflySim3D\Saved\AllActorSave` 及 `CopterSim\external\map`。

5.3 场景构建方式与参数化设计

- **动机**：通过参数化（尺寸、位置、颜色、物理属性）可以快速生成多组实验场景用于对比测试或算法鲁棒性验证。
- **构建方式**：
 - **手动布置场景方式**：通过 RflySim3D 的可视化编辑器直接拖拽场景元素。
 - **脚本化布置场景方式**：利用 `UE4CtrlAPI.py` 等 Python 接口机制发送创建命令，或者利用 JSON/YAML 作为中间场景描述文件，支持自动化编程式的场景生成。

5.4 基于 Python 的场景元素创建原理

直接使用 Python 脚本来向 RflySim3D 发送指令，能够在指定的坐标点动态地、自动化地批量创建和摆放场景元素（如障碍物）。

其核心原理是调用 RflySim 提供的 Python 接口库 `UE4CtrlAPI.py`。该接口本质上是通通过 UDP 协议与 RflySim3D 进行底层通信，向其发送封装好的控制数据包，进而触发引擎内生成或修改相应 Actor 的行为。这不仅支持单个元素的创建，也支持利用循环配合位置计算，实现复杂图案和大量元素的快速摆放（即参数化场景设计）。

我们可以参考该目录下自带的脚本示例 `TargetCreate.py`：

1. 导入接口库与初始化

首先引入接口文件并实例化控制类：

```
1 | import UE4CtrlAPI as UE4CtrlAPI
2 | ue = UE4CtrlAPI.UE4CtrlAPI()
```

2. 定义元素属性

明确需要创建的物体的类 ID（`vehicleType`）、三维空间坐标（`PosE`，分别对应 x,y,z ）、以及欧拉角姿态（`AngEuler`）。例如脚本中以列表的形式定义了多种障碍物（如 811 路径、824 立柱、815 方框等）的坐标：

```
1 | # [ vehicleType, x, y, z ]
2 | pos = [[811,6.5,-1,0], [814,7,0,0], [824,1,1,0], ...]
3 | ang = [[0,0,0], [0,0,0], [0,0,0], ...]
```

3. 批量生成元素

使用一个循环遍历定义好的属性列表，通过调用核心接口函数 `sendUE4PosNew` 将指令逐个发送给 RflySim3D：

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

提示：其中 `copterID` 代表该生成物体的特有识别 ID，通过设定不同的 ID 可以对创建出的目标进行独立的追踪、移动或后续删除；`runnedTime=-1` 代表创建一个全新的普通场景对象。详细的接口参数定义，强烈推荐阅读 `[RflySim安装目录]\RflySimAPIs\3.RflySim3DUE\API.pdf`。

6. 参考资料

1. [RflySim官方文档](#)
2. [RflySim3DUE API手册](#)
3. 通用场景描述建议：glTF 2.0 规范 (<https://www.khronos.org/glTF/>)
4. JSON 基础教程：Mozilla 开发者网络 (MDN)

7. 常见问题

Q1：重装工具链后，自定义创建的障碍物或地图场景丢失？

A1：通过 RflySim3D 手动创建保存的地图，在重新安装工具链时可能会被清除。建议在重装前将以下目录中相对应的 `<自定义的场景名>.json`、`<自定义的场景名>.png` 和 `<自定义的场景名>.txt` 文件进行备份：

- `[RflySim安装目录]\RflySim3D\RflySim3D\Saved\AllActorSave`
- `[RflySim安装目录]\CopterSim\external\map`

在新版本工具链安装完成后，将备份后的各类文件对应放回同名目录中即可正常加载该虚拟场景。

Q2：如何解决场景元素无法加载的问题？

A2：请检查 JSON 文件路径是否正确，或者场景元素是否存在于素材库中。

Q3: 脚本运行时报错 "ModuleNotFoundError" 怎么办?

A3: 请确保已正确安装 Python 环境，并安装了所需的依赖库。

Q4: 保存的场景无法在 RflySim3D 中显示?

A4: 请确认保存的 JSON 文件是否完整，或者尝试重新加载关卡。

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