

1、实验名称及目的

通过 python 脚本布置标靶场景实验：在不同场景中，通过运行 python 脚本同时创建三个标靶，并通过修改 python 脚本中对应的命令（发送的 Z 轴坐标）使之与地形匹配，使新生成的标靶与原有标靶相同。

2、实验原理

RflySim3D 内置的快捷键交互功能使得鼠标双击能捕捉到场景中的对象，获取其在全局场景中的位置（object pos），以及击中点的位置（click point）。根据获取的击中点的位置信息，之后在 python 脚本中为模型初始位置添加相应的偏移量，可以调整在场景中创建的模型的位置以使其与地形匹配。

```
ue.sendUE4Cmd('RflyChangeMapbyName Grasslands', 0)
```

调用 RflySim3D 控制台命令'RflyChangeMapbyName Grasslands'修改 UE 场景。这里的 RflyChangeMapbyName 命令表示切换地图(场景)，后面的字符串是地图名称，这里会将所有打开的窗口切换为草地地图。sendUE4Cmd 函数在 UE4CtrlAPI.py 库文件中的完整定义

```
sendUE4Cmd(cmd, windowid ==-1)
```

其中 cmd 为命令字符串，windowid 为接收窗口号(假设同时打开多个 RflySim3D 窗口)，windowid ==-1 表示发送到所有窗口。

```
ue.sendUE4Pos(1001,2150,0,[43.943,1.620,-11.465],[0,0,0])
ue.sendUE4Pos(1002,1150,0,[72.946,-4.388,-11.465],[0,0,0])
ue.sendUE4Pos(1003,151,0,[115.915,5.510,-10.448],[0,0,0])
```

创建物体并初始化物体的位置

3、实验效果

在场景原有的标靶旁，另外生成相同标靶。

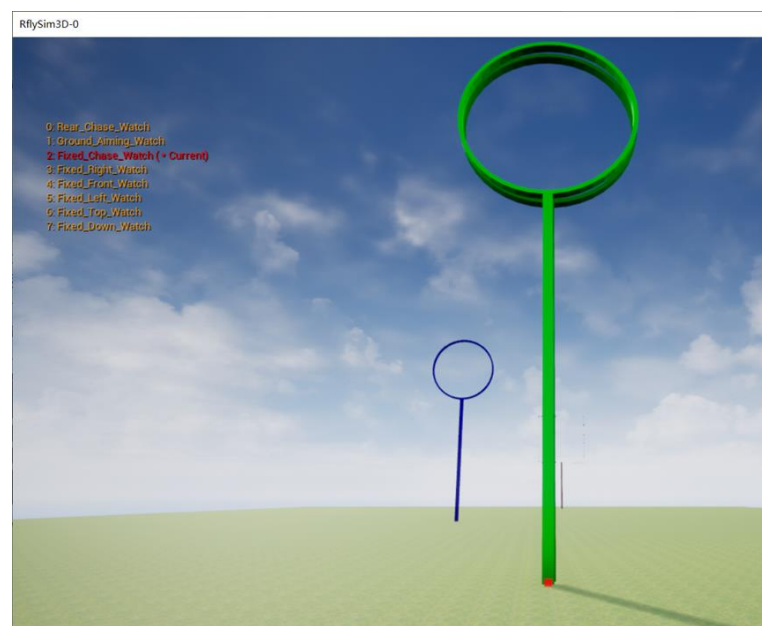


图 1

4、文件目录

文件夹/文件名称	说明
XML	标靶模型对应的 xml 文件
TargetCreateDemo.py	此文件调用了“UE4CtrlAPI.py”中的接口
TargetCreateDemo.bat	批处理脚本自动打开 RflySim3D
TargetCreateDemo2.py	此文件调用了“UE4CtrlAPI.py”中的接口
TargetCreateDemo2.bat	批处理脚本自动打开 RflySim3D
UE4CtrlAPITest.py	此文件调用了“UE4CtrlAPI.py”中的接口

5、运行环境

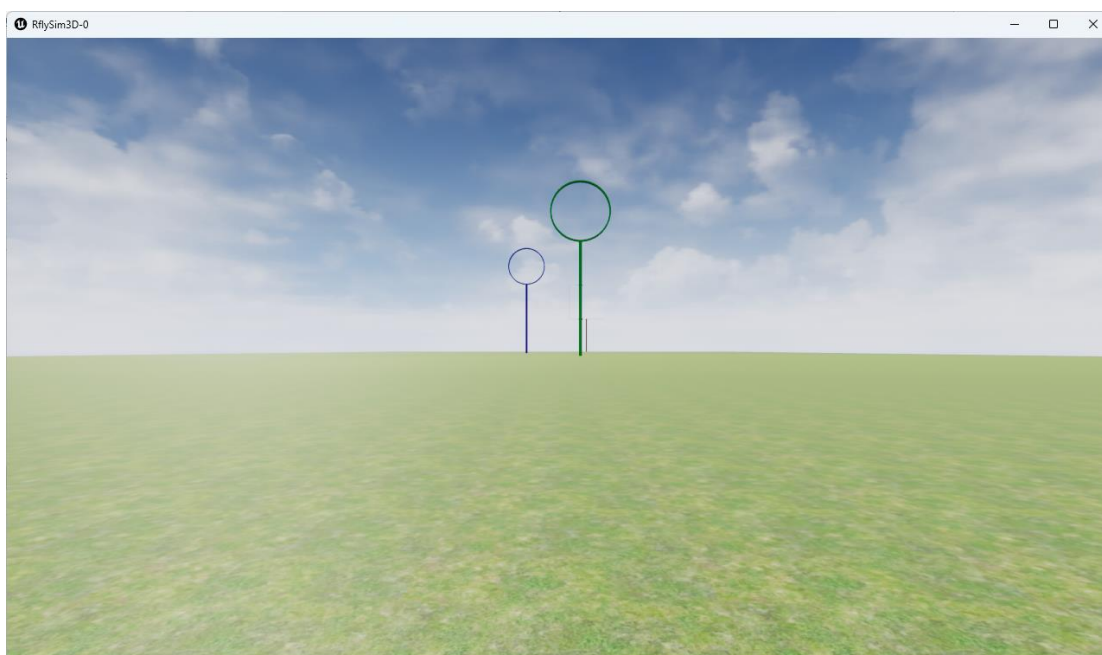
序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台完整版		
3	Python 3.11		

推荐配置请见：<https://doc.rflysim.com>

6、实验步骤

Step 1: 启动 RflySim3D 切换到对应地图

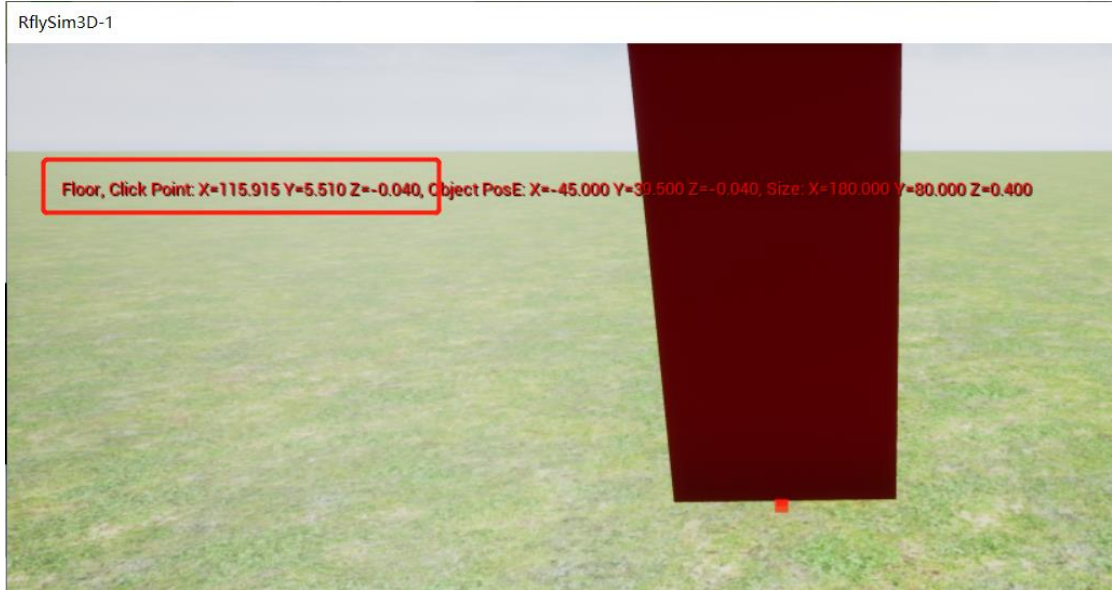
首先，打开 RflySim3D 程序，并反复按下“M”键直至切换到 VisionRing 地图。



Step 2: 获取放置标靶的坐标位置

依次将视角拖动到三个靶标的根部，双击根部的地面位置，得到期望放置靶标的坐标位置。如下图可知，绿、蓝、红三个靶标的地面位置分别为[43.943,1.620,-0.04]，[72.946,-4.388,-0.04]，[115.915,5.510,-0.04]。





Step 3: 用 python 脚本创建场景

以 UE4CtrlAPITest.py 为模版，创建一个三维物体创建的程序。分别放置绿、蓝、红三个靶标到场景中。

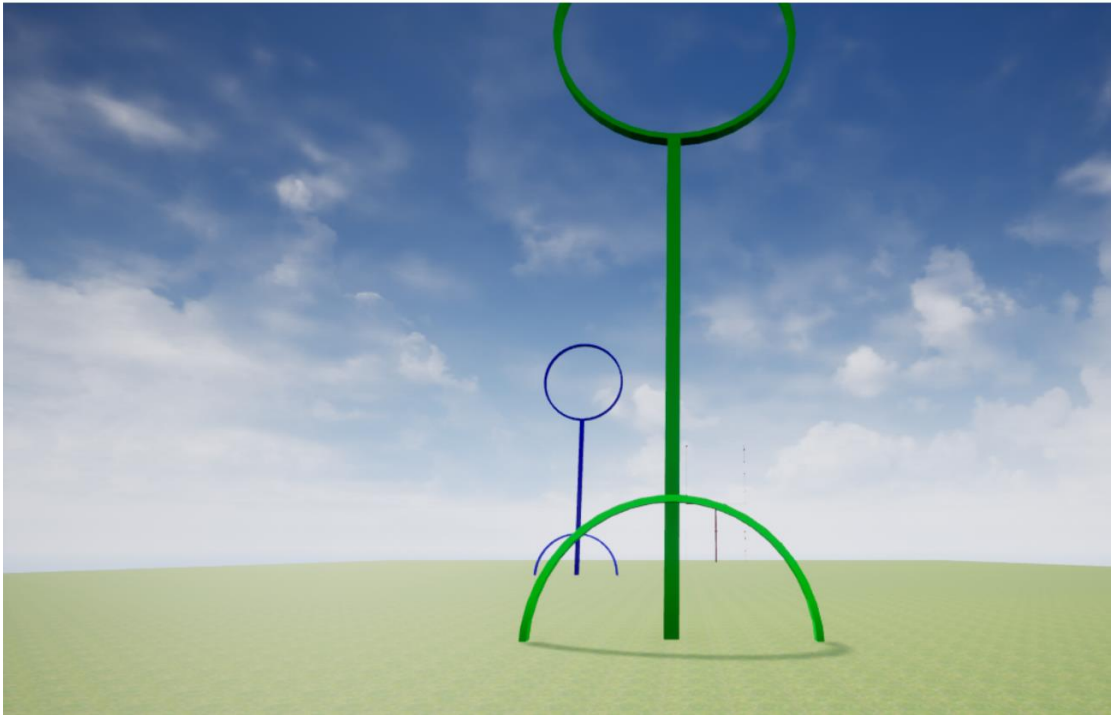
将地图切换到 VisionRing，切换地图的代码可表示为

```
ue.sendUE4Cmd(b'RflyChangeMapbyName VisionRing',0)
```

参考对应的 xml 文件，在 ue.sendUE4Pos 的命令中直接发送 vehicleType 来直达需要的样式，格式是：样式序号*1000+classid（其中，样式序号=相同 classid 下模型的排列顺序-1，排列顺序由 displayorder 标签决定）。绿色圆形靶标的 ID 为 2150（参考 Ring_Target_Green.xml），同样的道理蓝色为 1150（参考 Ring_Target_Blue.xml），红色方框靶标的 ID 为 151 或 0151（需要看 XML 文件 Quad_Target_Red.xml）。因此，发送靶标的代码可以表示为

```
ue.sendUE4Pos(1,2150,0,[43.943,1.620,-0.04],[0,0,0])
ue.sendUE4Pos(2,1150,0,[72.946,-4.388,-0.04],[0,0,0])
ue.sendUE4Pos(3,151,0,[115.915,5.510,-0.04],[0,0,0])
```

完成修改后，打开 RflySim3D 并运行 UE4CtrlAPITest.py 脚本。这里在 visionring 地图原靶对应位置生成了新的靶标，但是由于没有考虑圆心到地面的高度，我们生成的圆靶标都不是底部贴在地面上。

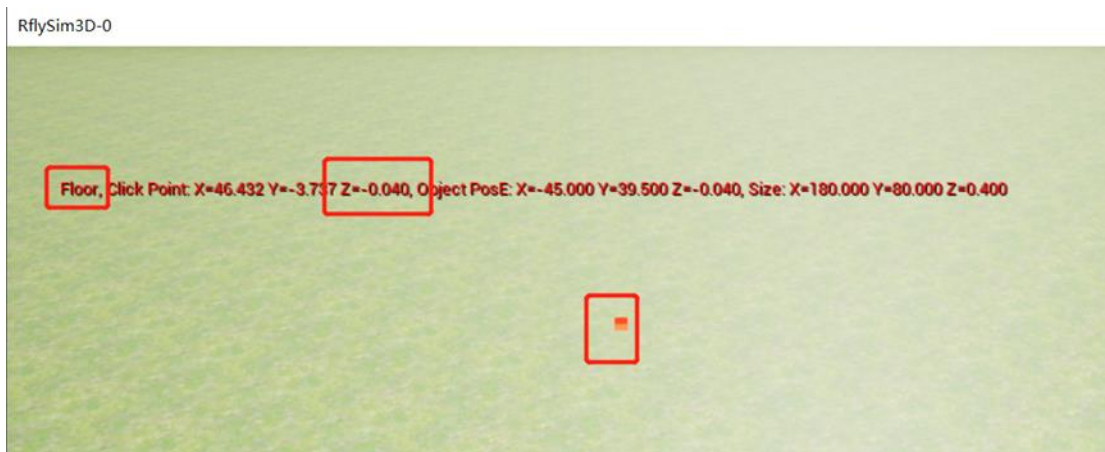


Step 4: 修改 python 脚本

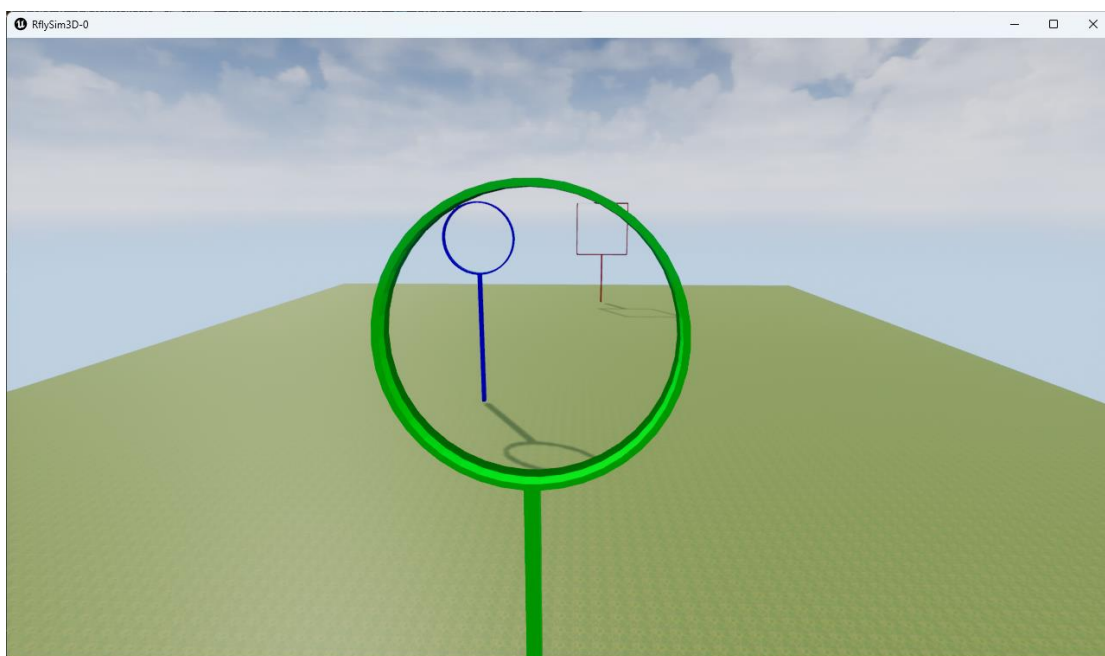
鼠标双击地面和每个标靶的底部，得到每个物体中心到底部的高度差，可以得到绿色和蓝色圈需要上移 11.463m（这里不需要考虑 XML 里面的缩放），红色方框靶标需要 10.448m。因此，重新调整后的物体创建语句为

```
ue.sendUE4Pos(1,2150,0,[43.943,1.620,-11.465],[0,0,0])
ue.sendUE4Pos(2,1150,0,[72.946,-4.388,-11.465],[0,0,0])
ue.sendUE4Pos(3,151,0,[115.915,5.510,-10.448],[0,0,0])
```





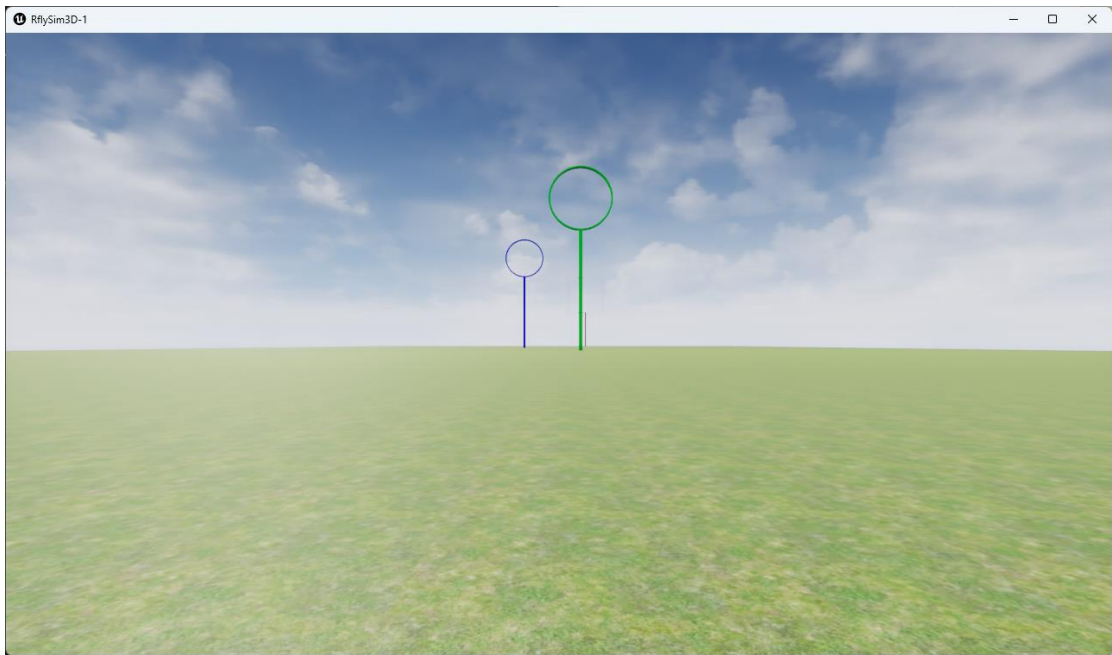
重启 RflySim3D 并运行修改后的 python 脚本。可见场景原有标靶旁生成了相同的新标靶。



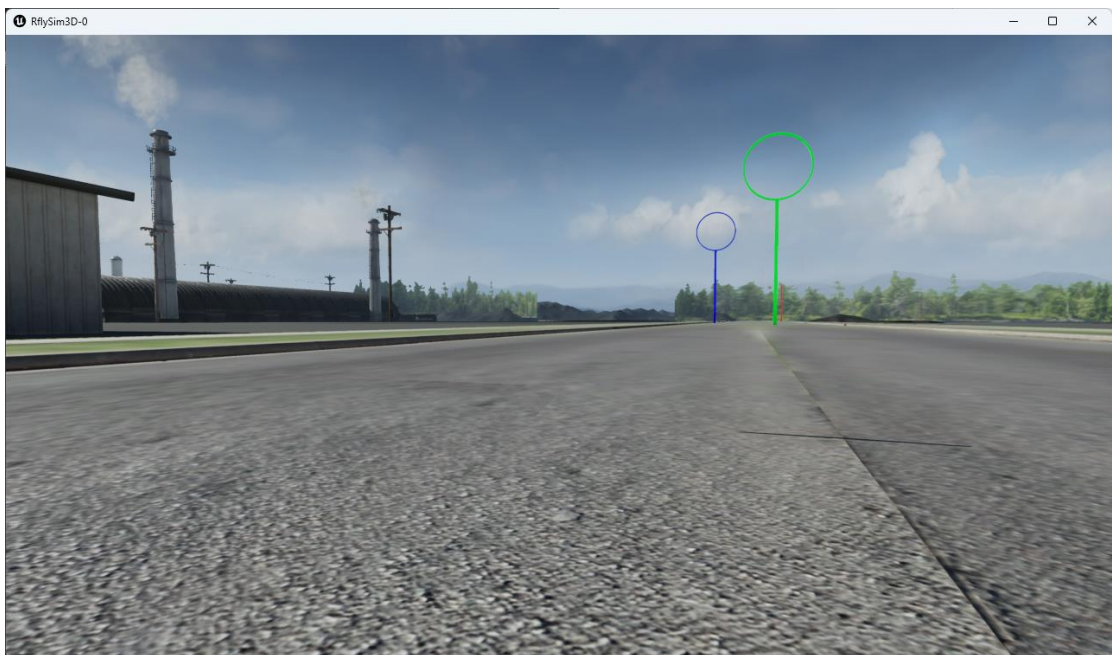
Step 5: 在不同场景生成标靶

按前述方法修改 python 脚本，可在不同场景中生成这三个标靶。

运行 TargetCreateDemo.bat 和 TargetCreateDemo.py 可在空白场景中复现 VisionRing 中的三个标靶。



运行 TargetCreateDemo2.bat 和 TargetCreateDemo2.py 可在 OldFactory 中创建三个标靶。



7、参考资料

- [1]. XML 文件规则 [\(见 API 文档\)](#)
- [2]. RflySim3D 快捷键接口总览 [\(见 API 文档\)](#)
- [3]. RflySim3D 控制台命令接口总览 [\(见 API 文档\)](#)

8、常见问题

- 1. 无