
RflySim3D 常用特效接口实验原理

1. 文件目录.....	1
2. 总体说明.....	1
3. 关键功能的实现.....	1
3.1. 常用 python 接口	错误!未定义书签。
sendUE4Cmd	错误!未定义书签。
fillList.....	错误!未定义书签。
sendUE4Attatch	错误!未定义书签。
sendUE4ExtAct.....	错误!未定义书签。
sendUE4PosNew.....	错误!未定义书签。
3.2. 添加标靶和障碍物并与场景地形匹配	错误!未定义书签。
获取物体初始高度与场景地形表面的偏移量	错误!未定义书签。
考虑偏移量重新创建物体	错误!未定义书签。
自动获取地形偏移量并创建物体	错误!未定义书签。
4. 相关文献.....	5
附加资源.....	5

1. 文件目录

例程目录: [\[安装目录\]\RflySimAPIs\4.RflySimModel\0.ApiExps\14.inCopterData\](#)

文件夹/文件名称	说明

2. 总体说明

2.1. 粒子系统

2.1.1. 发射器

2.1.2. 生命周期

2.1.3. 粒子碰撞

2.2. 刚体模拟

2.3. 柔性物体模拟

2.4. 地形模拟

2.4.1. 分形理论

2.5. 特效的生成

3. 关键功能的实现

3.1. 生成通信特效

下面是 [Comm\CommDemo.py](#) 代码的详细解释:

导入模块

```
import time
import math
import UE4CtrlAPI as UE4CtrlAPI
import csv
import os
import random
import sys
```

这些是 Python 脚本中用到的模块。例如, `time` 模块用于处理时间相关操作, `math` 模块用于数学计算, `UE4CtrlAPI` 是一个自定义模块, 用于与 Unreal Engine 4 进行通信。

初始化 UE4CtrlAPI

```
ue = UE4CtrlAPI.UE4CtrlAPI()
```

这行代码创建了一个与 Unreal Engine 4 进行通信的对象。

主函数

```
if __name__ == "__main__":
```

这段代码确保只有在直接运行这个脚本时，以下的代码才会被执行。

```
ue.sendUE4Pos(1, 3, 0, [0, 0, -9])
ue.sendUE4Pos(2, 3, 0, [10, 0, -9])
ue.sendUE4Pos(3, 3, 0, [10, 10, -9])
ue.sendUE4Pos(4, 3, 0, [0, 10, -9])
ue.sendUE4Pos(5, 3, 0, [-10, 10, -9])
ue.sendUE4Pos(6, 3, 0, [-10, 0, -9])
ue.sendUE4Pos(7, 3, 0, [-10, -10, -9])
ue.sendUE4Pos(8, 3, 0, [0, -10, -9])
```

- 打开 RflySim3D，初始化 8 台无人机的位置，分别为[0, 0, -9], [10, 0, -9], [10, 10, -9], [0, 10, -9], [-10, 10, -9], [-10, 0, -9], [-10, -10, -9], [0, -10, -9]。其中第一位参数表示无人机的 ID，第二位参数表示无人机的类型，第三位参数表示是否启用位置控制（0 表示不启用），后面三位参数表示无人机的坐标。

```
ue.sendUE4Pos(9, 802)
```

- 创建一个通讯特效 Actor，ID 为 9，三维样式类型为 802。

```
for i in range(7):
    ue.sendUE4ExtAct(9, [1,i+2,20,i,0,0,0,0,0,0,0,0,0,0,0])
    time.sleep(0.05)
```

- 延迟 1 秒后，循环 7 次，每次给特效 Actor 发送一个数据包，其中第一位参数表示发送方 ID，第二位参数表示接收方 ID，第三位参数表示特效存在的时间（20 秒），第四位参数表示特效的样式（从 0 到 6），后面的参数表示其他信息（暂时不用）。

3.2. 生成虚拟管道

三维虚拟管道主要用于可视化无人机集群飞行。它可以将不同的管段连接起来，形成一个完整的网络。生成三维虚拟管道的原理主要有以下几个步骤：

- 首先，需要定义管段的几何形状和属性，如长度、半径、弯曲度、材质等。
- 然后，需要确定管段之间的连接方式，如端点对齐、切平、旋转等。
- 接着，需要计算管段之间的变换矩阵，用于表示管段在三维空间中的位置和方向。
- 最后，需要使用图形库或者游戏引擎，根据变换矩阵，将管段的几何信息转换为三维图形数据，如顶点、面、纹理等，以便将管段渲染到屏幕上，形成三维虚拟管道。

下面是 [Tunnel\UnitPipeline.py](#) 代码的详细解释：

导入必要的库

```
import time
import math
import UE4CtrlAPI as UE4CtrlAPI # 用于与 Unreal Engine 4 交互的库
import csv
import os
import sys
```

导入 time 库，用于处理时间相关的函数

导入 math 库，用于处理数学相关的函数

导入 UE4CtrlAPI 库，这是一个用于与 Unreal Engine 4 交互的自定义库，可以控制场景中的无人机、相机等对象

导入 csv 库，用于读写 csv 格式的文件

导入 os 库，用于处理操作系统相关的函数

导入 sys 库，用于获取命令行参数等系统相关的信息

初始化 UE4CtrlAPI 对象

```
ue = UE4CtrlAPI.UE4CtrlAPI()
```

UE4CtrlAPI 是一个 Python 模块，用于控制和获取 UE4 中的无人机相关信息。该模块提供了一个 UE4CtrlAPI 类，可以创建对象来与 UE4 进行交互。这里实例化了一个名为 ue 的 UE4CtrlAPI 对象，用于后续的操作和控制

定义启动 RflySim3D 的函数

```
def start_rflysim():  
    # 打开 RflySim3D 模拟器  
    os.startfile(r'C:\PX4PSP\RflySim3D\RflySim3D.exe')  
    time.sleep(3) # 等待 3 秒以确保模拟器已启动
```

start_rflysim 函数的作用是启动 rflysim 模拟器。函数首先调用 os 模块的 startfile 方法，传入 rflysim 模拟器的路径，从而打开模拟器程序。然后调用 time 模块的 sleep 方法，传入 3 秒作为参数，让程序暂停执行 3 秒。这样做的目的是等待模拟器完全启动，避免后续操作时出现错误

定义退出 RflySim3D 的函数

```
def quit_rflysim():  
    # 强制关闭 RflySim3D 进程  
    os.system(r'taskkill /F /IM RflySim3D.exe')
```

通过这个命令，可以强制关闭 RflySim3D 程序，不管它是否正在运行或者卡住。os 是一个 Python 模块，可以用来执行操作系统命令。os.system 是一个函数，可以用来在终端中运行指定的命令。r'taskkill /F /IM RflySim3D.exe' 是一个 Windows 命令，用来强制结束名为 RflySim3D.exe 的进程，/F 表示强制结束，/IM 表示指定进程名，RflySim3D.exe 是执行文件名

定义读取初始位置的函数

```
def read_initial_location(filename, x, y, z):  
    # 读取 CSV 文件中的数据  
    filename = os.path.join(sys.path[0], filename + '.csv')  
    with open(filename) as f:  
        f_csv = csv.DictReader(f)  
        for row in f_csv:  
            x.append(float(row['x']))  
            y.append(float(row['y']))  
            z.append(float(row['z']))  
    print("读取文件完成...")
```

这个函数的作用是从一个 CSV 文件中读取一组点的坐标，并将它们分别存储到 x, y, z 三个列表中。CSV 文件的每一行代表一个点，包含三个字段：x, y, z，表示该点的 x, y, z

轴的值。参数 filename 是 CSV 文件的文件名，不包含后缀。参数 x, y, z 是三个空列表，用来存储读取的数据

自定义异常类

```
class rFlyException(Exception): # 自定义异常类
    def __init__(self, message): # 初始化函数，接收一个参数 message
        Exception.__init__(self) # 调用父类的初始化函数，将 message 作为异常对象的信息
        self.message = message # 将 message 赋值给实例属性 message，以便后续使用
```

这个类是用来自定义一个异常类型，用于在程序运行过程中发生错误时抛出，并显示相应的信息。

定义创建目标管道的函数

```
def create_initial_target(initx, inity, initz):
    # 在 Unreal Engine 中创建管道蓝图
    ue.sendUE4Pos(100, 803)
    print("开始创建管道...")
    time.sleep(2)
    RadiusX = 50 # 半径 X
    RadiusY = 50 # 半径 Y
    R = 0.6 # 颜色 R
    G = 0.6 # 颜色 G
    B = 0.6 # 颜色 B
    Opacity = 0.6 # 透明度
    flag = 0
    for (x, y, z) in zip(initx, inity, initz):
        # 发送管道参数到 Unreal Engine
        ue.sendUE4ExtAct(100, [0, x/100.0, y/100.0, z*-1/100.0, RadiusX, RadiusY, R, G, B, Opacity, 0, 0, 0, 0, 0, 0])
        # 通过修改第一个参数可以修改管道的样式
        # 0: 圆形管道
        # 1: 方形管道
        time.sleep(0.01)
```

这个函数的作用是根据传入的三个列表，分别代表管道上一系列点的 x, y, z 坐标，来在 Unreal Engine 中绘制出一条管道。首先，调用 ue.sendUE4Pos 函数在场景中添加管道的三维模型。然后，设置管道的半径、颜色和透明度，并遍历三个列表，每次取出一个点的坐标，乘以相应的比例（单位换算为厘米），转换为 Unreal Engine 的坐标系，然后调用 ue.sendUE4ExtAct 函数，将这个点的信息发送到 Unreal Engine，让它在该点处添加一个管道节点。根据需要，可以修改管道的样式，例如圆形或方形。

主程序

```
if __name__ == "__main__":
    # 启动 rflysim
    print("启动 RflySim3D...")
    start_rflysim()
    time.sleep(3)
    # 切换场景
    ue.sendUE4Cmd('RflyChangMapbyName Grasslands')
    try:
        initX = []
        initY = []
        initZ = []
```

```

# 读取文件中的坐标
read_initial_location('Pipeline', initX, initY, initZ)
try:
    # 创建管道
    create_initial_target(initX, initY, initZ)
    time.sleep(2)
    # ue.sendUE4ExtAct(100, [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]) #
替换为不透明材质
    # 参数一 0: 添加节点 1: 更换材质
    # 参数十一: 0: 透明材质 1: 不透明
    # Move()
except:
    print("错误")
    quit_rflysim()
except rFlyException as e:
    quit_rflysim()
    print(e.message)

```

这是主程序的部分，首先调用 `start_rflysim` 函数，启动 RflySim3D 模拟器，并等待三秒。然后调用 `ue.sendUE4Cmd` 函数，切换到草地场景。接下来，创建三个空列表，分别用来存储 x, y, z 坐标。然后调用 `read_initial_location` 函数，从 Pipeline.csv 文件中读取坐标，并存入相应的列表。如果读取过程中发生异常，就调用 `rFlyException` 类，抛出异常，并结束模拟器，打印异常信息。如果读取成功，就调用 `create_initial_target` 函数，根据读取的坐标，创建管道。如果创建过程中发生异常，也同样结束模拟器，并打印错误信息。如果创建成功，就可以对管道进行进一步的操作，例如改变材质或移动位置。

4. 相关文献

- [1]. [..\API.pdf](#)
- [2]. [..\e3_InitAPI\Intro.pdf](#)
- [3]. [..\e4_UAVCtrl\Intro.pdf](#)
- [4]. [IConsoleManager | Unreal Engine Documentation](#)

附加资源

官方文档：RflySim 官方文档： <https://rflysim.com/doc/zh/>

社区交流：加入 RflySim 技术交流群：951534390

