

UE蓝图实现激光命中扫描与特效销毁机制

1. 实验目的

本实验旨在介绍游戏开发中经典的“命中扫描 (Hitscan)”逻辑。通过RflySim平台的自定义事件接口驱动发射器Actor发射射线，探测撞击点目标，命中后在精准位置触发Niagara爆炸特效，并在延迟后销毁目标物体与发射器自身。

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\2.AdvExps\e2_BlueprintModel\5.LasersEffect](#)

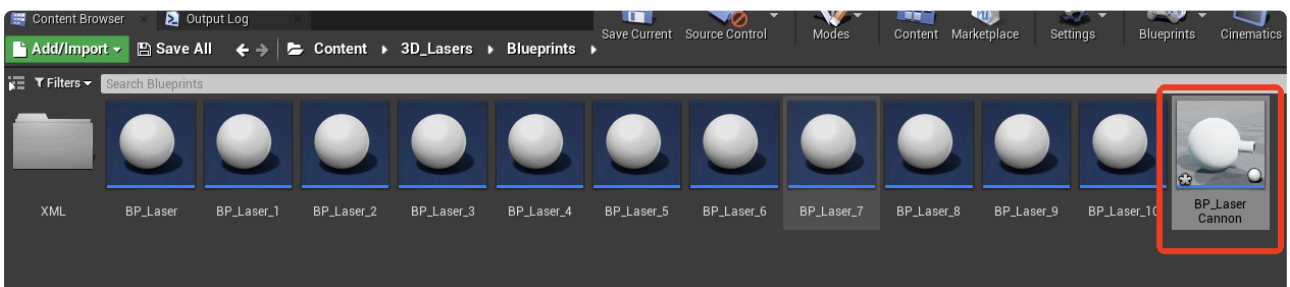
- ****.ulg：日志文件，记录了每架无人机的飞行数据、控制指令和系统响应等信息。
- ****.ulg：日志文件，记录了每架无人机的飞行数据、控制指令和系统响应等信息。

4. 实验内容或步骤

激光命中生成爆炸特效如图：



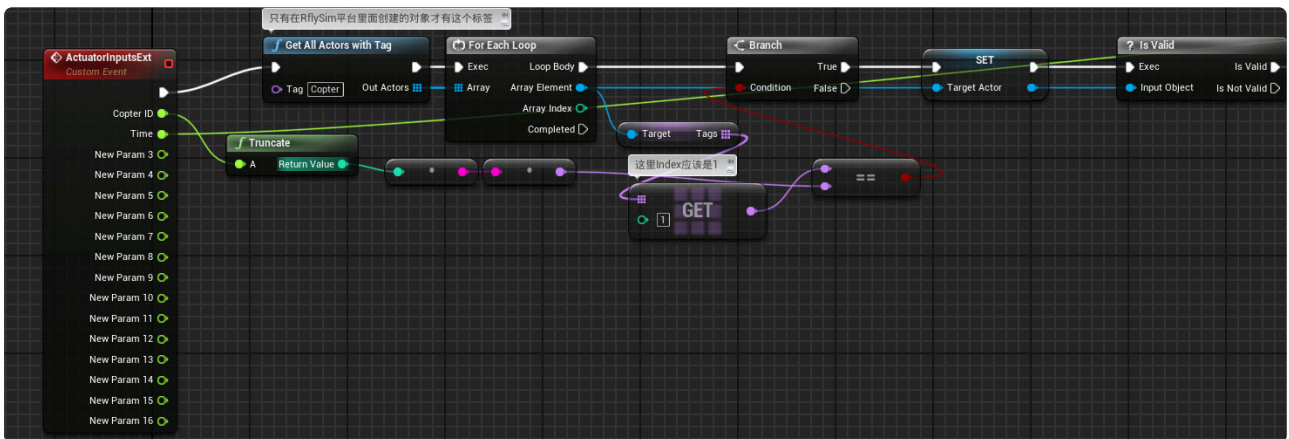
这套蓝图实现的是游戏开发中非常经典的**“命中扫描 (Hitscan)”**逻辑：即由自定义事件 `ActuatorInputsExt` 驱动（对应RflySim蓝图接口）从发射器 Actor发射一根射线（这里为这根射线添加了激光特效），探测撞击点（这里设置为目标物体），命中后一定时间在撞击点触发爆炸，爆炸后销毁目标物体和激光特效。



4.1 步骤 1：自定义事件触发与目标获取

首先，我们要建立整个逻辑的入口，并通过传入的 ID 匹配到对应的无人机模型。

1. **自定义事件**: 在蓝图面板右键搜索并添加 `Custom Event` , 命名为 `ActuatorInputsExt` 。
2. **获取所有无人机**: 引出执行线, 添加 `Get All Actors with Tag` 节点, 将 `Tag` 设为 `Copter` 。
3. **提取标签**: 使用 `For Each Loop` 遍历获取的数组。从 `Array Element` 中获取 `Tags` , 并用 `GET` 节点获取索引为 `1` 的标签 (此处图中配有注释: “只有在RflySim平台里面创建的对象才有这个标签”, 且**“这里Index应该是1”**)。
4. **ID 比对**: 将事件节点输出的 `Copter ID` 使用 `Truncate` 截断为整数, 然后与上一步提取的标签对应转换为字符串 (String) 后通过 `==` 进行比较。
5. **保存目标**: 将比较结果连入 `Branch` 的 `Condition` 。当判定为 `True` 时, 使用 `SET` 节点将当前循环的 `Array Element` 保存至 `Target Actor` 变量中, 最后连接 `Is Valid` 节点验证目标是否有效, 再进入后续逻辑。



4.2 步骤 2: 实时瞄准目标与更新发射方位

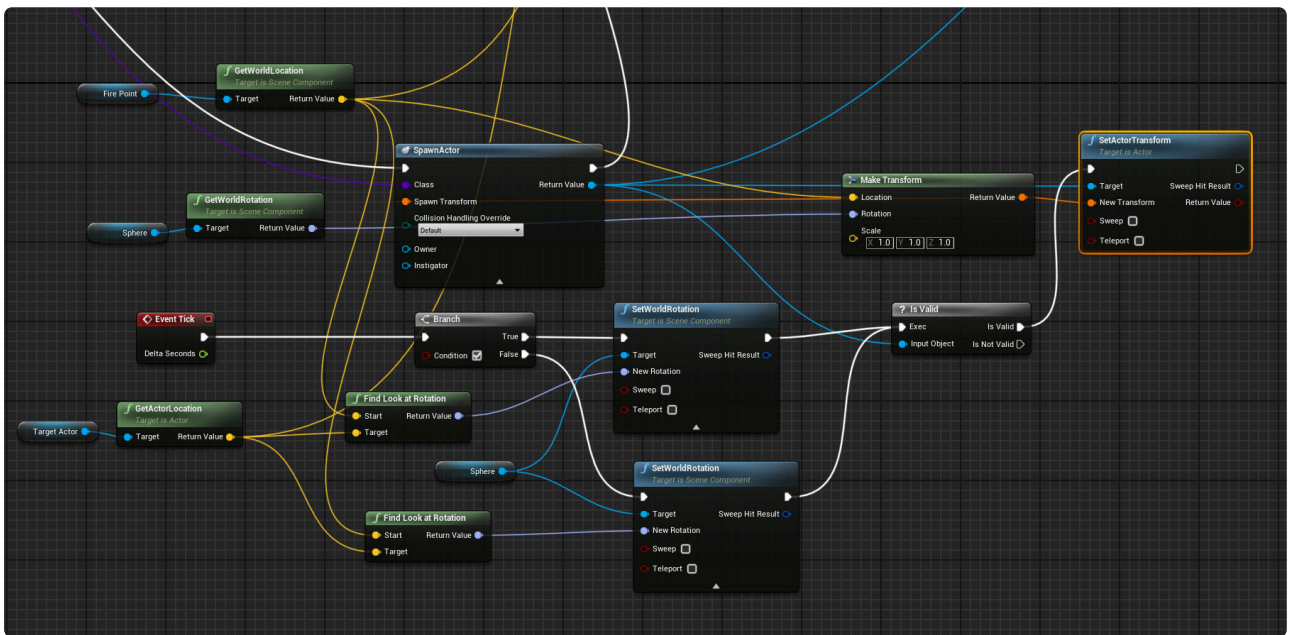
为了让发射器 (炮台) 能够实时对准目标, 且持续生成的激光效果能跟随炮口同步移动, 我们需要在 `Event Tick` (每帧更新) 中动态计算组件的朝向变换:

1. **明确组件层级**: 从组件结构面板可以看出, 蓝图中存在 `Sphere` (炮台基座) -> `Cylinder` (炮管) -> `Fire Point` (实体的开火原点) 的嵌套层级。
2. **计算炮塔朝向 (Find Look at Rotation)**:
 - 使用 `GetWorldLocation` 提取 `Fire Point` 的实时空间坐标作为起点 (`Start`)。
 - 使用 `GetActorLocation` 提取早前锁定的 `Target Actor` 坐标作为目标终点 (`Target`)。
 - 两者求出的正是需要对准目标的旋转量。
3. **整体旋转炮台 (SetWorldRotation)**:
 - 将上述旋转结果接入 `SetWorldRotation` 节点的 `New Rotation` 。

- 目标 (Target) 设为基座 Sphere 组件，保证炮管与炮口整体无缝跟随目标转向。

4. 吸附同步激光特效 (SetActorTransform):

- 过 Is Valid 验证先前生成的激光/特效 Actor (引自 SpawnActor 的返回集) 是否依然有效。
- 若有效，使用 Make Transform 节点打包新的变换状态：Location 取自 Fire Point (确保发源于炮管顶部)，而 Rotation 取自 Sphere 的世界旋转。
- 最后更新给 SetActorTransform 节点，这让激光束被“死死锁在”炮火开火位置并随炮台同步转向。



4.3 步骤 3：执行射线检测 (Line Trace)

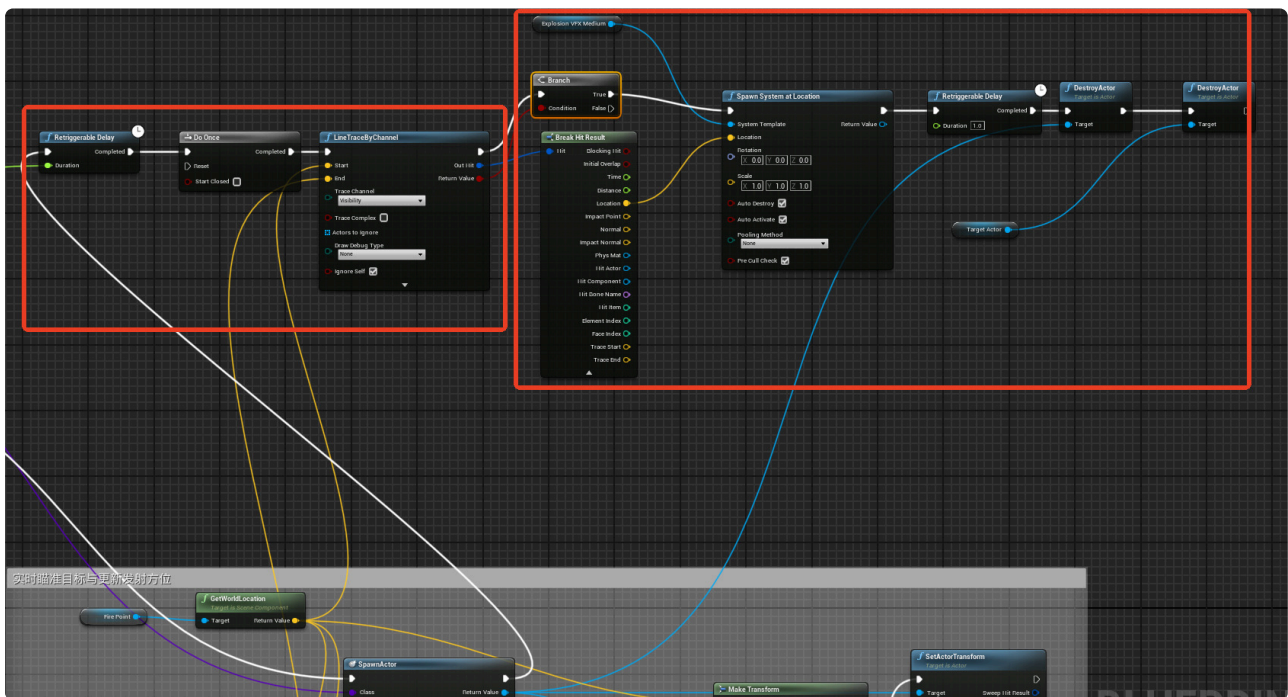
这是激光探测目标的核心步骤。为了让射线检测只被执行一次以防止重复触发爆炸，前面可加入相应的控制节点。

1. **触发控制**：在光束特效生成的执行流后，依次接入 **Retriggerable Delay** 节点与 **Do Once** 节点。
2. **节点设置**：搜索并放置 **LineTraceByChannel** 节点。
3. **坐标连线**：
 - 射线的起点 (Start) 连入炮口 **Fire Point** 组件取到的 **GetWorldLocation**。
 - 射线的终点 (End) 连入早前锁定的 **Target Actor** 取到的 **GetActorLocation**。

4.4 步骤 4：命中检测与触发爆炸销毁机制

当激光“射中”目标时，要在相交位置生成爆炸并在短暂延迟后销毁双方。

1. **分支判断 (Branch)**: 将射线检测的 **Return Value** 连入 **Branch** 判断是否发生了碰撞 (**True**)。
2. **提取坐标数据 (Break Hit Result)**: 从 **Out Hit** 引出并使用 **Break Hit Result** 节点。这里我们只需要提取出 **Location**，也就是激光命中目标表面的精确三维坐标。
3. **引爆 Niagara 特效**:
 - 使用 **Spawn System at Location** 节点生成 Niagara 粒子特效。
 - **System Template**: 为此引出一个变量（如图中命名为 **ExplosionVFX_Medium**，类型设为 **Niagara System**），并在详情面板给予一个合适的默认特效资产，例如 **NS_Explosion_Big_1**。
 - **Location**: 直接连入前一步 **Break Hit Result** 所提取出的 **Location** 引脚。
4. **延迟与双重销毁**:
 - 在爆炸特效生成后，接入一次 **Retriggerable Delay** 设定一定时长（比如由图中所示的 **1.0** 秒）让爆炸演完。
 - 紧跟其后，通过连续调用两次 **DestroyActor** 清除实体：第一次接上目标变量的引用，将对方无人机 (**Target Actor**) 销毁；第二次目标保持默认的自身 (**Self**)，将火炮发射器自身也一并销毁。



5. 关键知识点

- 命中扫描 (Hitscan) 逻辑在此类射击需求场景的应用。
- UE 蓝图射线检测 (LineTraceByChannel) 与世界坐标转换。
- Niagara 系统的动态生成 (Spawn System at Location) 与连带销毁回收机制。

6. 参考资料

此处编写参考资料，编写样式如下：

1. [RflySim官方文档](#)
2.
3.

7. 常见问题

Q1：射线没有正常检测到目标怎么办？

A1：请检查目标 Actor 是否阻挡了对应通道的射线检测碰撞，以及 Target Actor 的标签 (Tag) 是否匹配。

Q2：爆炸特效始终在同一个地方生成或方向不对？

A2：请检查 Break Hit Result 提取出的 Location 是否正确传入粒子生成节点。如果对方向有要求，可以通过 Impact Normal 计算特效的旋转朝向。

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