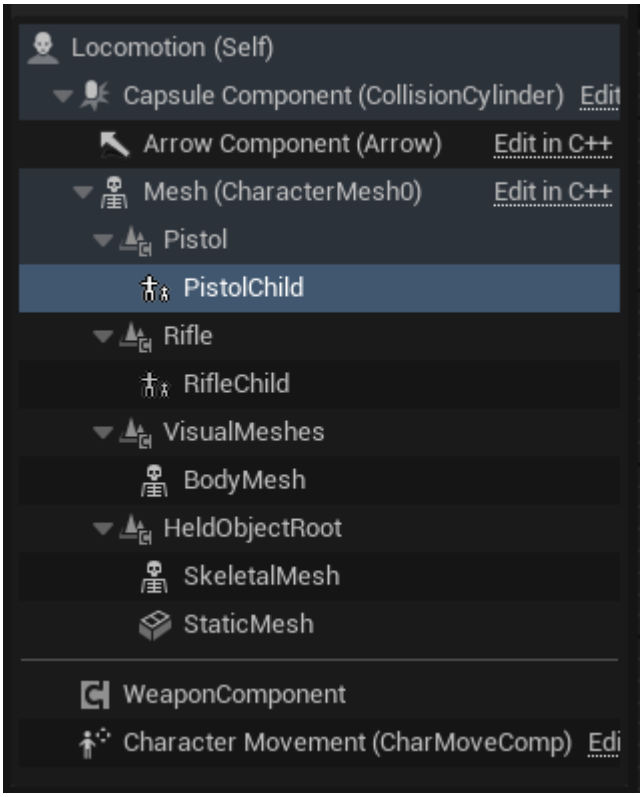
ALS V4 Weapon System

ALS V4 Weapon System

> 整个武器系统挂载在人物蓝图内，通过 Scene 组件定位到角色 Socket，再在其下挂载 Child Actor 来生成具体的 Weapon Actor。



---

1. 系统架构

1.1 核心组件构成

整个 Weapon 系统由以下三个核心组件构成：

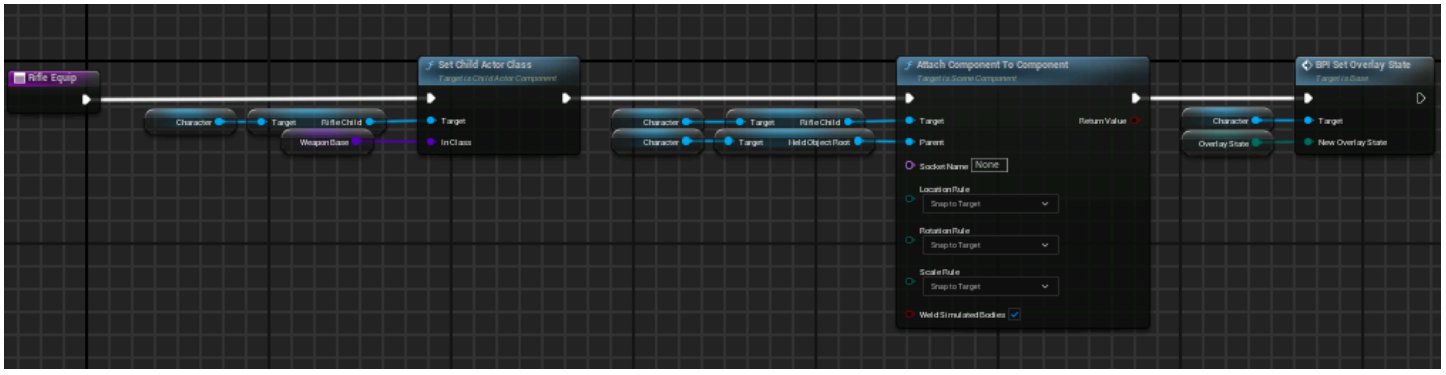
组件	类型	职责
Weapon Base	Actor	武器实体。内部挂载武器骨骼 Mesh，再通过 Child Actor 挂载到角色身上。新增武器只需基于该 Base 创建子类即可。
Weapon Component	Actor Component	武器系统的逻辑核心。负责武器装备/卸下、开火、射线检测等逻辑。
Scene / Child Actor	场景组件	在人物蓝图中用 Scene 定位到指定 Socket，其下挂载 Child Actor 用于生成 Weapon Actor。



---

2. 武器装备与卸下

2.1 装备逻辑



装备武器的流程如下：

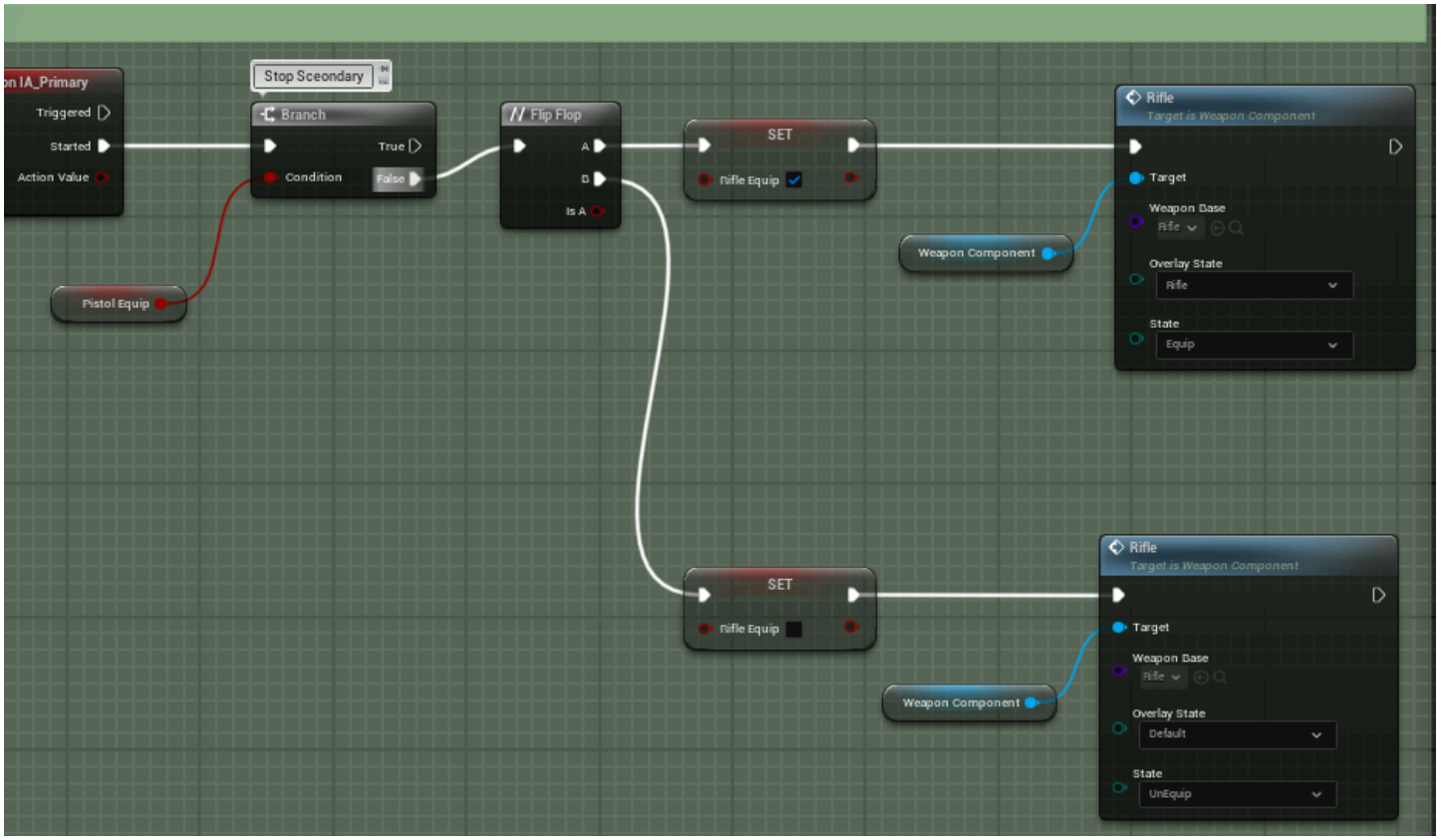
1. **获取 Child Actor 组件** — 从角色蓝图中取得用于挂载武器的 Child Actor。
2. **设置 Weapon Base** — 指定要生成的武器子类。
3. **Attach 到 Held Object** — 将武器 Attach 到角色蓝图中的 `HeldObject` Socket 上（手持位置）。
4. **更新 Overlay State** — 修改 Overlay State，切换动画蓝图中的状态，播放持枪动画。

2.2 卸下逻辑

卸下装备的逻辑与装备相反：

- 武器初始状态是挂在背上的（`Rifle` Socket）。
- 卸下时只需将武器 Attach 回到背部的 `Rifle` Socket 即可。

2.3 角色状态判断

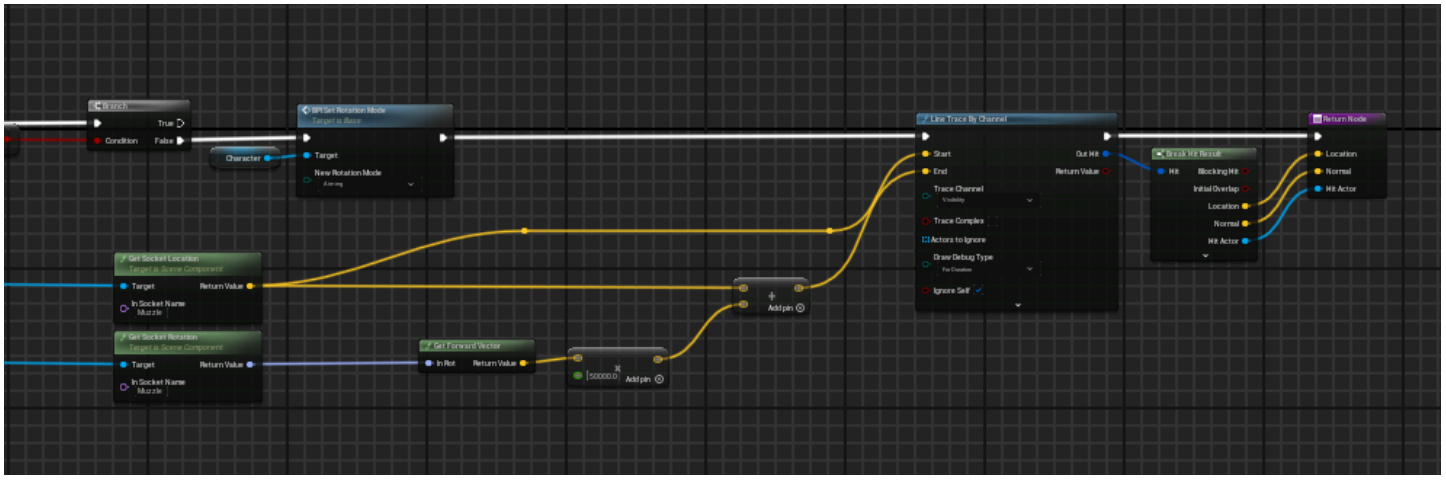


回到角色蓝图中，做一些 **Boolean 判断与设置**，用于判断当前人物是否处于装备武器状态，从而驱动后续的动画与逻辑切换。

---

3. 开火与射线检测

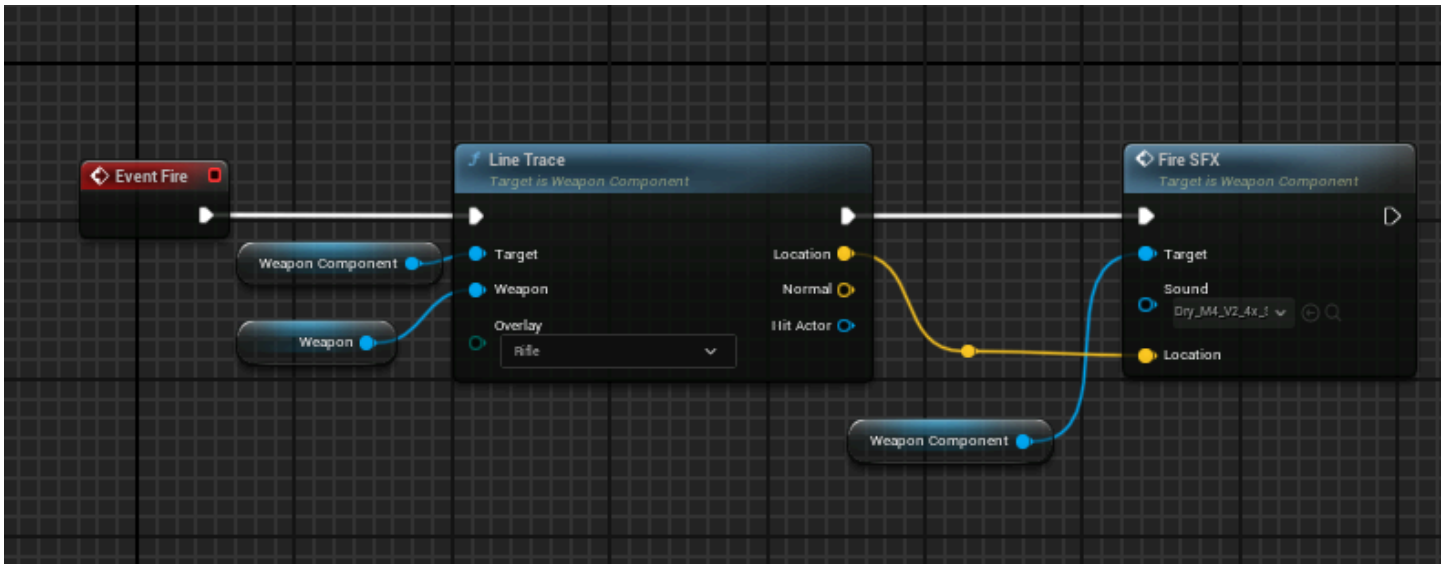
3.1 射线检测原理

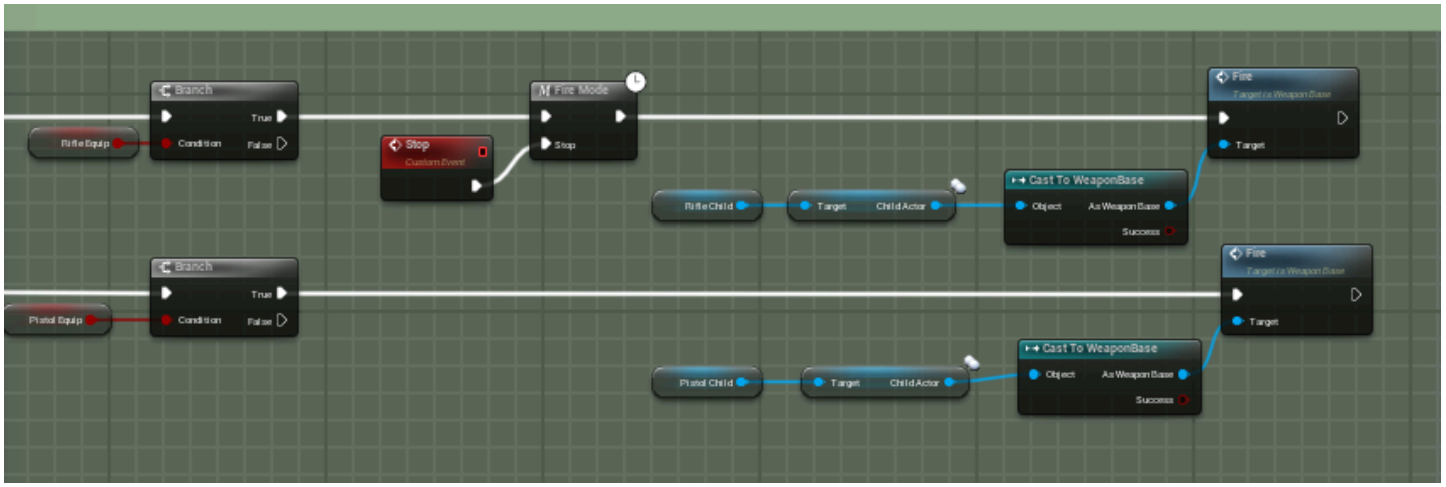


射线检测封装在 **Weapon Component** 中：

- 在武器的骨骼 Mesh 上添加一个 **Muzzle Socket**（枪口位置）。
- 从该 Socket 的位置和方向出发进行射线检测（Line Trace）。
- 因为每把武器的枪口位置不同，所以射线检测逻辑是**运行在每一个具体的 Weapon Base Actor** 内部的，而不是统一写在 Component 里。

3.2 角色蓝图调用开火

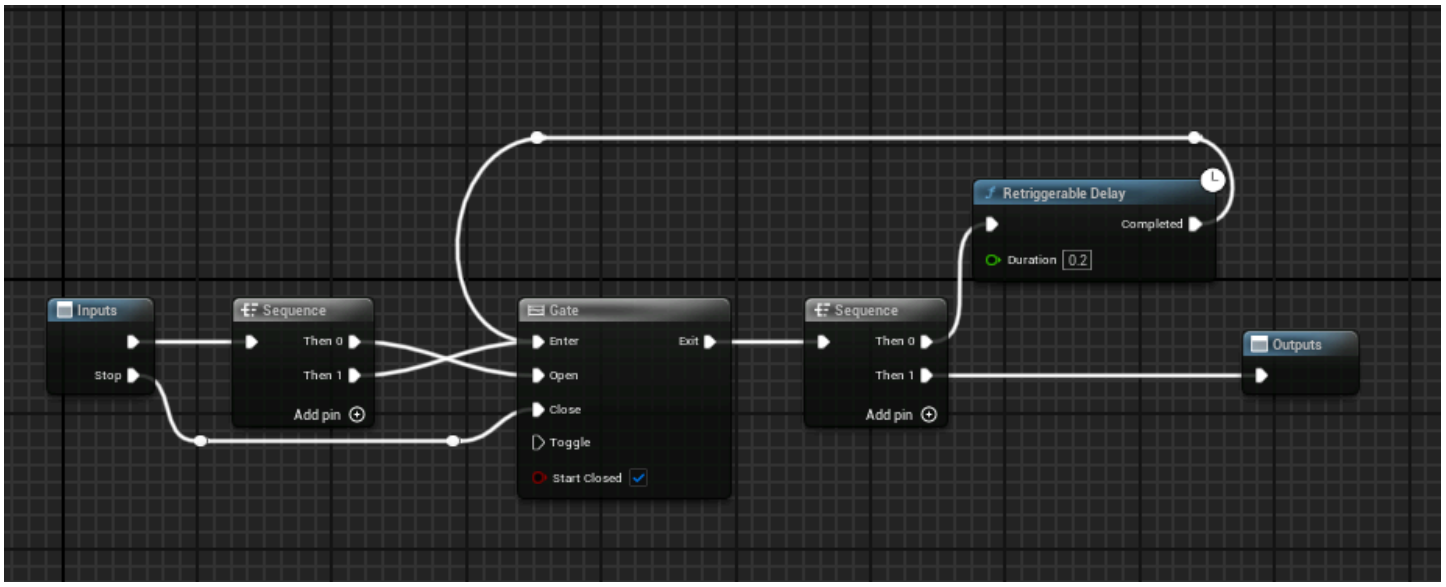




然后在角色蓝图中调用该事件，完成最终的开火操作。

---

4. 全自动射击循环（宏实现）



这里是一个**宏（Macro）**，核心目的是实现**全自动射击的循环功能**。

4.1 核心流程拆解

代码块

```
┌──────────────────────────────────────────────────────────────┐
│  Start（开始射击）                                            │
│    ├─ Then 0: 打开 Gate（Open）                               │
│    └─ Then 1: 进入 Gate（Enter）                              │
│           ↓                                                  │
│  Gate（门，默认 Start Closed）                                │
│           ↓ Exit                                             │
│    ├─ Then 1: 立即触发 Output（执行射击逻辑）                  │
│    └─ Then 0: 进入 Retriggerable Delay（0.2s）                │
│           ↓ Completed                                        │
│      绕回到 Gate 的 Enter 引脚 ──────────────────────────┘    │
│                                                              │
│  Stop（停止射击）                                             │
│    └─ 直接关闭 Gate（Close），循环中断                        │
└──────────────────────────────────────────────────────────────┘
```

4.2 各节点说明

节点	作用
Sequence	将执行流程按顺序拆分，Then 0 先执行，Then 1 后执行
Gate	类似水龙头开关，只有 Open 状态下，Enter 的电流才能从 Exit 流出
Start Closed	勾选后 Gate 默认关闭，需要显式 Open 才能通过
Retriggerable Delay	可重触发的延迟节点，0.2 秒后输出 Completed
循环回路	Delay Completed 后绕回 Gate 的 Enter，若 Gate 仍 Open，则再次触发射击

4.3 射速计算

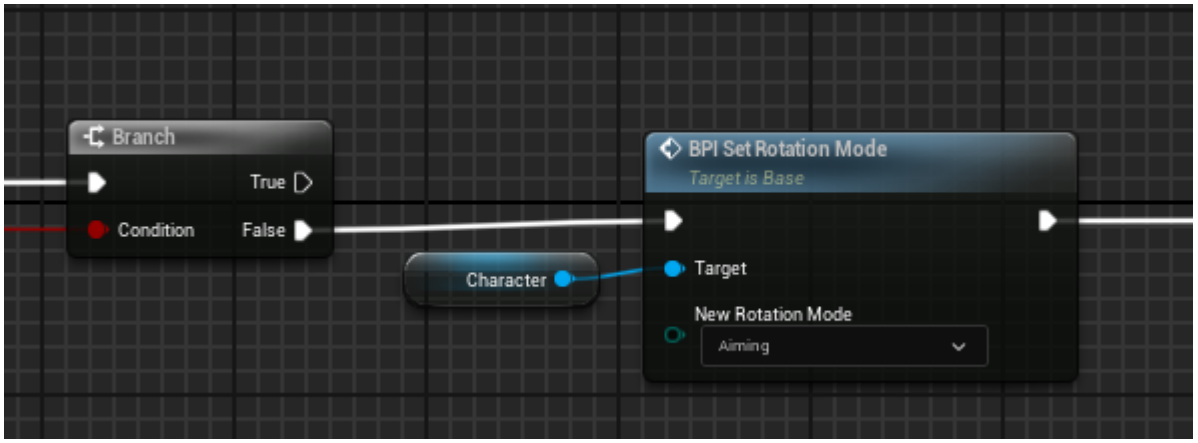
```
1   射速 = 1s / 0.2s = 5 发/秒
```

按下 **Stop** 时，Gate 被 **Close**，即使 Delay 结束也无法再进入循环，射击停止。

---

5. 待解决问题与后续方案

5.1 武器乱射问题

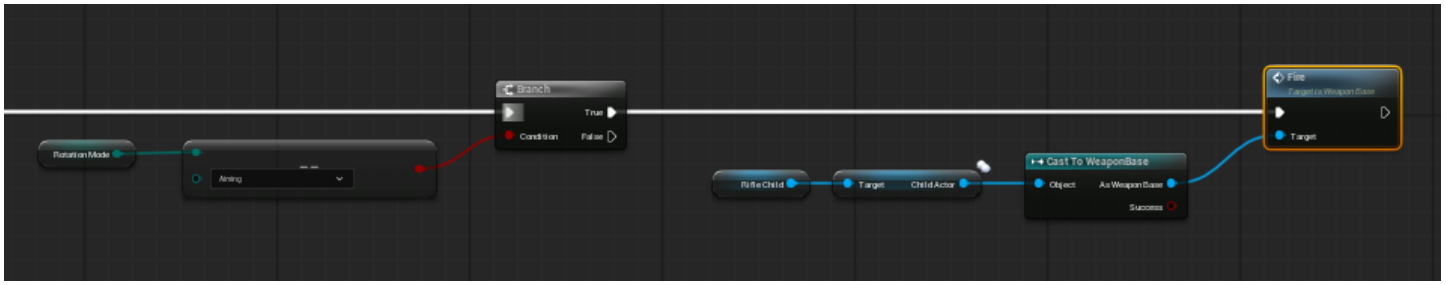


当前方案：利用 Aiming Pose 解决武器乱射问题 —— 一旦按下左键开火，角色会直接切换到 Aiming 状态。

存在的问题：

- 一旦进入 Aiming 状态，**不会自动解除**。
- 必须**再按一次右键**才能退出瞄准。

5.2 后续解决方案



改进方案：在开火前增加 **Rotation 判定**，只有朝向合适时才允许开火。

伴随的新问题：

- 必须按右键才能开火。
- **没有腰射模式（Hip Fire）**。

未来规划：

- 考虑增加一个独立的 **ADS State（瞄具瞄准状态）**。
- 常规移动时保持 Aiming，按右键进入 ADS。
- 这是一个较大的工程，难度较高，**暂时搁置**。

---

6. 总结

功能模块	关键实现	所在位置
武器挂载	Scene + Child Actor 生成 Weapon Base	人物蓝图
装备/卸下	Attach 到 HeldObject / Rifle Socket	Weapon Component
状态切换	Overlay State + Boolean 判断	人物蓝图
射线检测	Muzzle Socket → Line Trace	各 Weapon Base Actor
全自动射击	Gate + Retriggerable Delay 循环宏	Weapon Component
瞄准控制	Aiming Pose（待优化为 ADS State）	人物蓝图