ALS V4 AI Patrol

ALS V4 AI Patrol

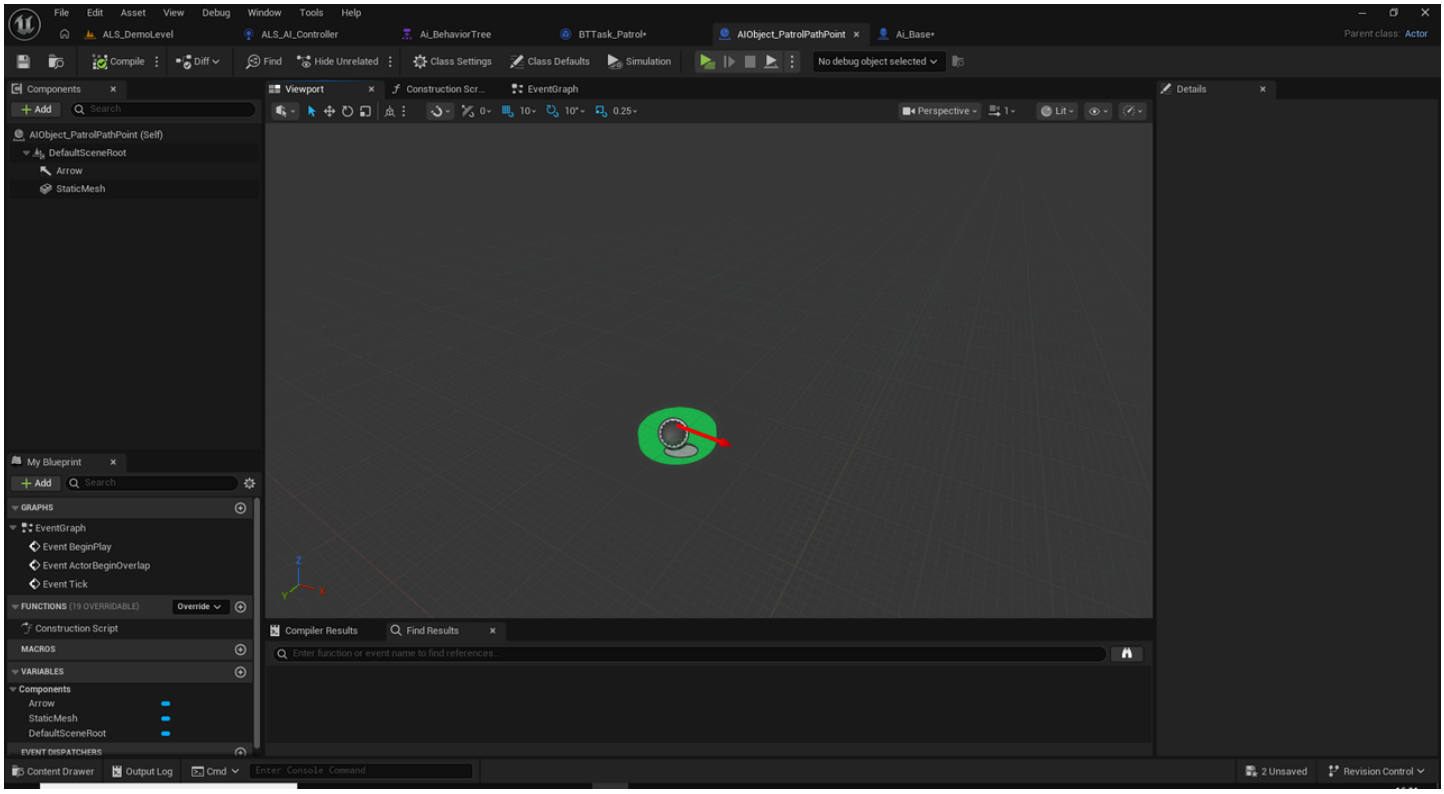
> AI 巡逻系统实现，基于 Behavior Tree + Blackboard，支持多路径点巡逻、等待时间与双模式循环（Loop / Return）。

---

1. 前期准备

1.1 创建 Target Point Actor

首先创建一个 **Target Point Actor**，作为场景中 AI 巡逻路径的定位点。

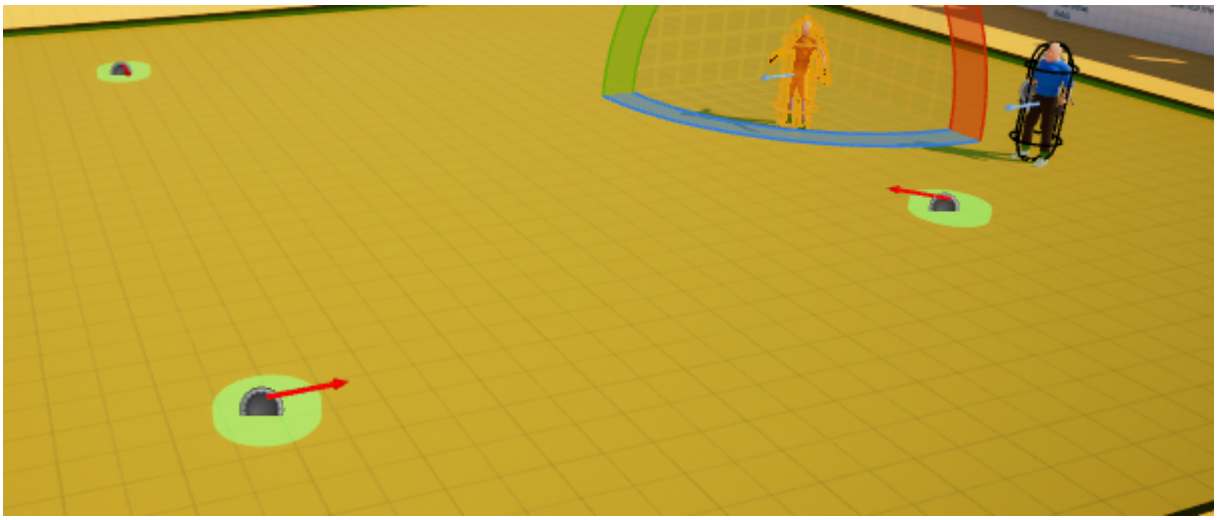


1.2 创建 Behavior Tree 与 Blackboard

创建 Behavior Tree、Blackboard 等 AI 所需组件。

1.3 场景布置

在场景中摆放好若干个 **Target Point**，构成 AI 的巡逻路线。



---

2. AI Component 配置

2.1 添加 Patrol Path Map 变量

在 AI Component 中添加一个 `Patrol Path` **的 Map 变量**，用于同时配置：

Key	Value	说明
Target Point (Actor)	Waiting Time (Float)	每个巡逻点对应的停留等待时间



> 使用 Map 结构的好处：每个目标点可以独立配置在该点的等待时长。

---

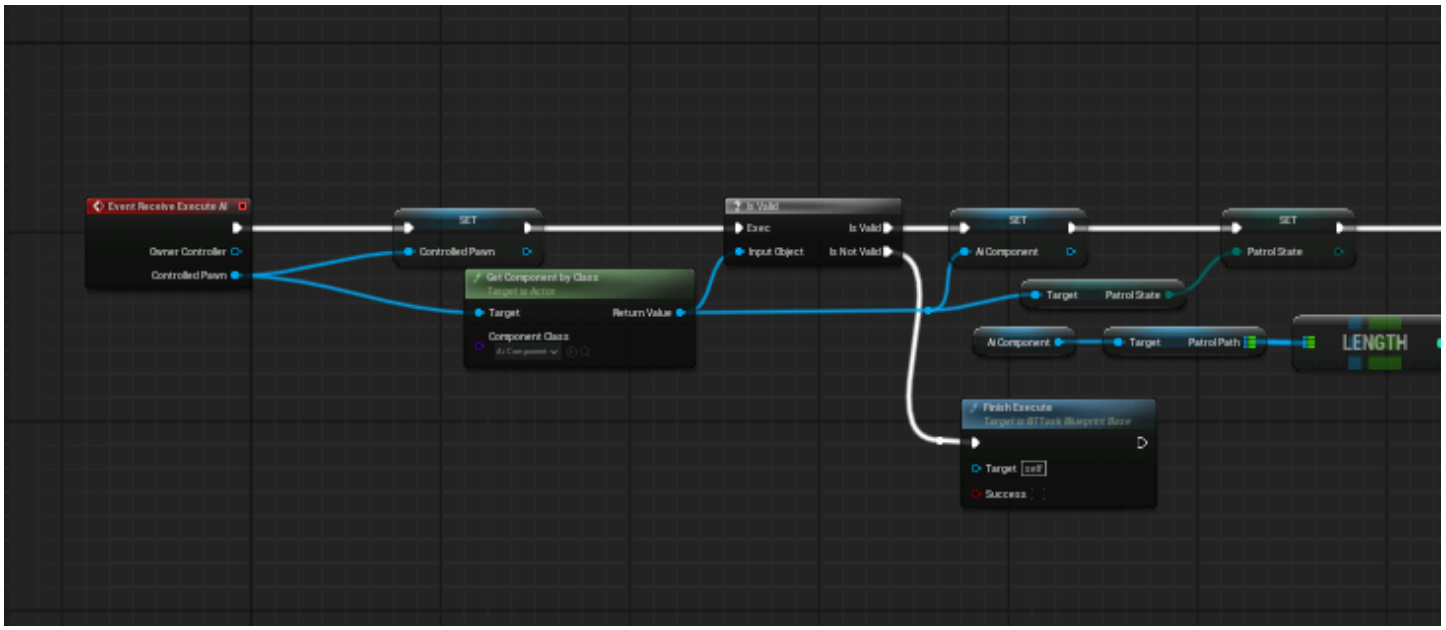
3. Behavior Tree Task 实现

3.1 创建 Patrol Task

创建一个新的 **Task**，用于处理巡逻逻辑。

3.2 识别判定与变量设置

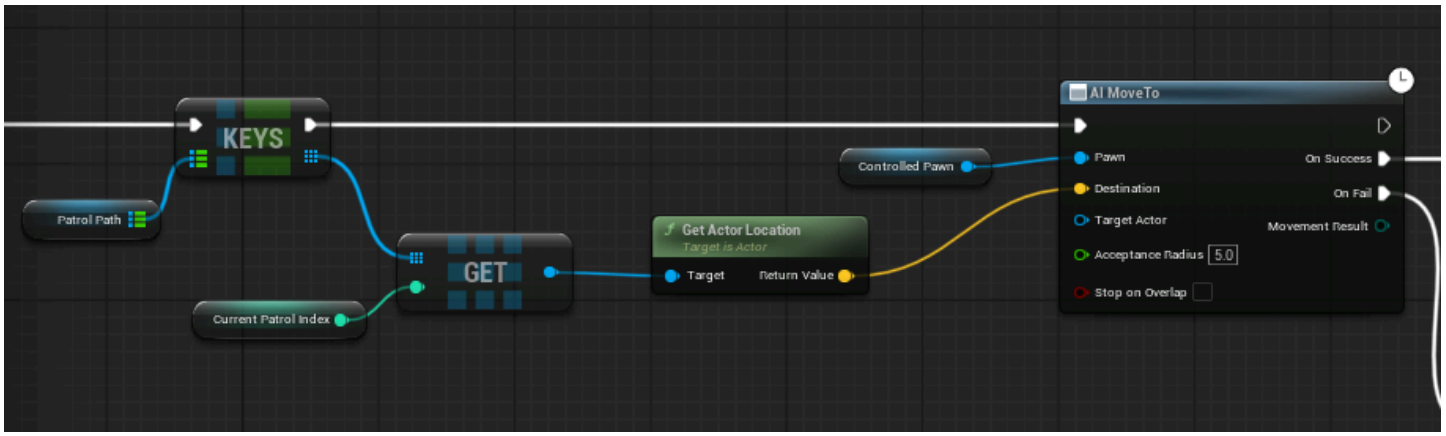
首先就是各种识别判定以及设置变量。



4. 巡逻核心逻辑

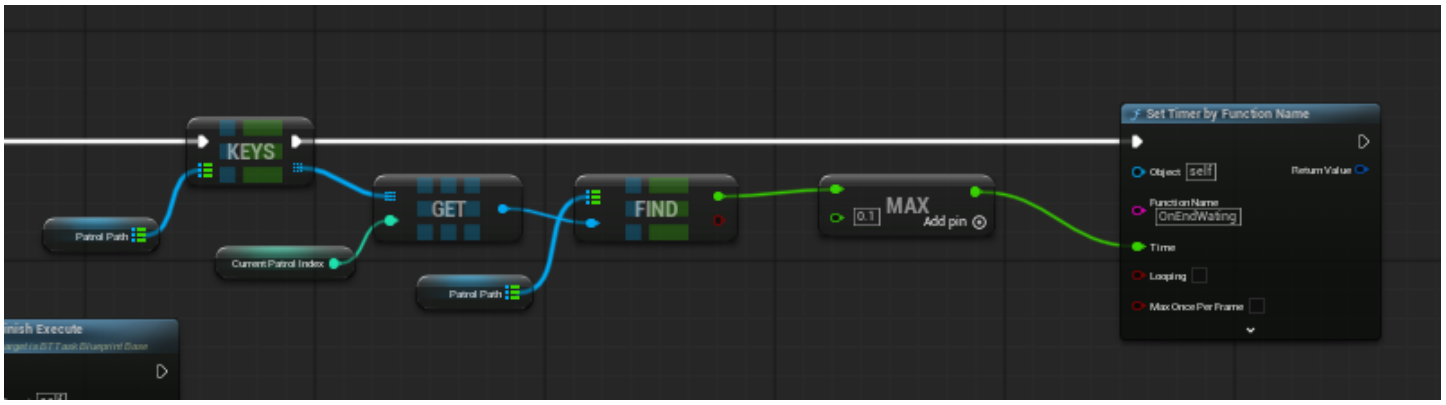
4.1 移动到目标点

从 `Patrol Path` 这个 Map 变量中获取对应的 **Target Point Actor**，然后执行 **Move To**，前往该 Target Point 的 Location。



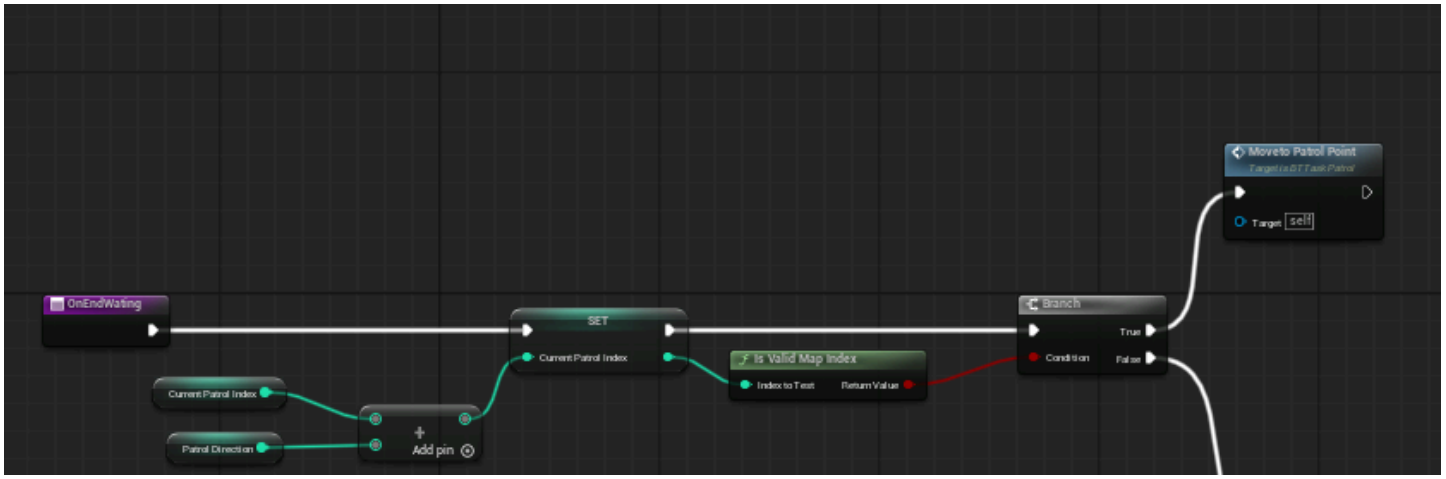
4.2 到达后等待

AI 走到目标点后，在原地停留一段时间。停留时长取自 Map 变量中对应 Target Point 填写的 **Float 值**，并通过一个**计时器**来控制等待时间。

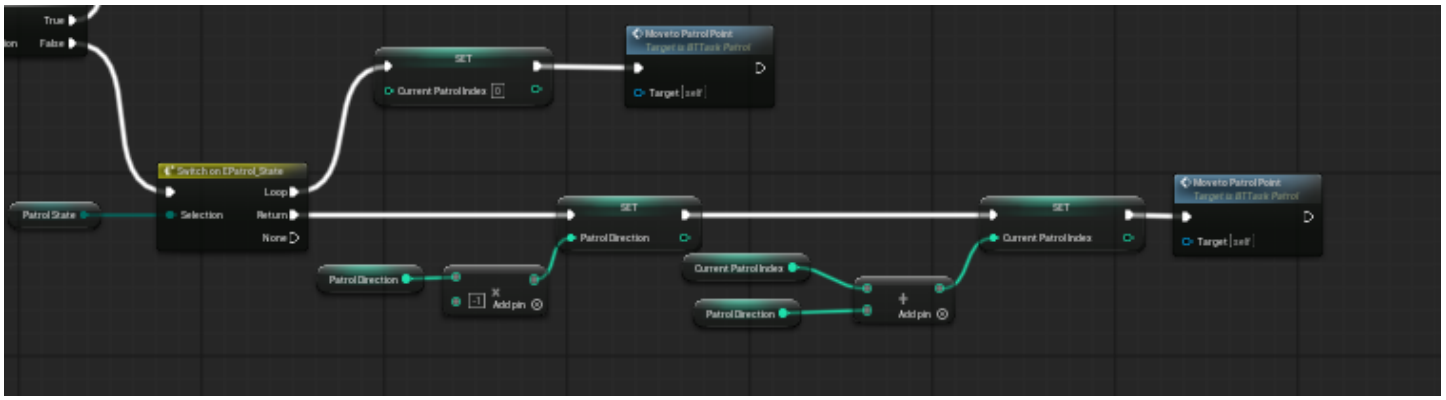


4.3 切换到下一个目标点

当计时器时间到了，索引 **+1**，AI 前往下一个目标点。



- 如果目标点还没走完，继续循环：**Move To → Wait → +1 → Next Point**
- 如果所有 Target Point 都走完了，流程走到 **False**，进入模式选择分支。



---

5. 巡逻模式

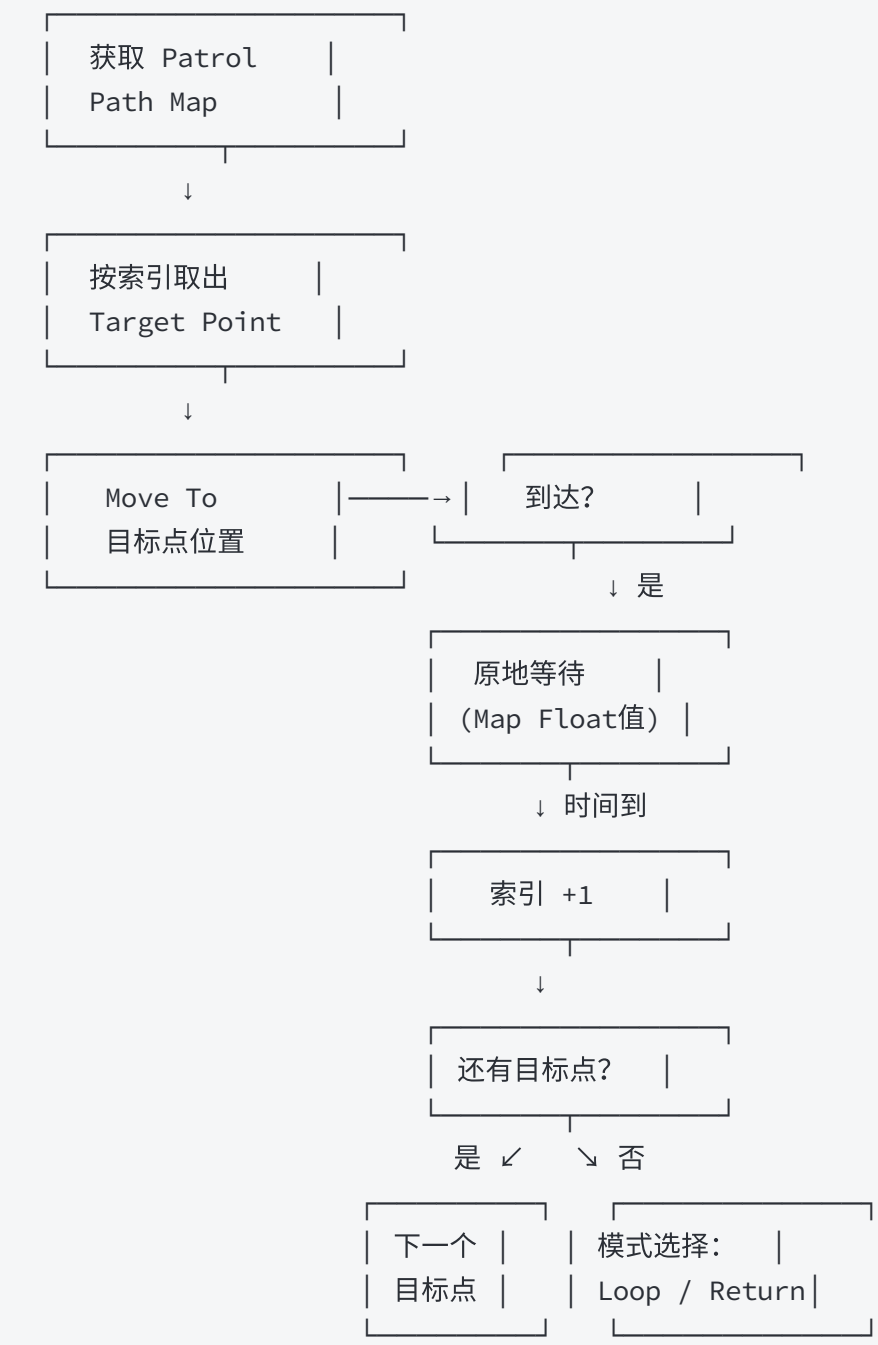
当所有目标点走完后，提供两种巡逻模式：

模式	说明
Loop	AI 根据目标点顺序**循环巡逻**，走完最后一个点后回到第一个点重新开始
Return	AI **原路返回**，按反方向依次经过之前走过的目标点，形成来回巡逻

---

6. 整体流程图

代码块



7. 总结

模块	关键实现
路径点定义	Target Point Actor 摆放在场景中
数据存储	AI Component 中的 Patrol Path Map（Target Point → Waiting Time）
移动控制	BT Task：获取 Map 中的 Target Point → Move To
等待控制	BT Task：到达后读取对应 Float 值 → 计时器等待
模式切换	走完所有点后进入 Loop（循环）或 Return（原路返回）分支