**AI Controller：**

开始运行使用延迟+运行行为树（延迟用于AI使用蓝图开始动画）

增加AI Perception感知组件，于感官配置中添加AI视力配置、伤害感官配置、预感感官配置和聆听配置（归属检测中全部启用）

事件图表中使用目标感知更新事件对于探测到的目标进行不同的感官事件分配

（使用从刺激物获取感官类节点的命名字符串选择）

各感官事件大致以Black Board设置为布尔值，从中断AI模拟中获取是否成功检测的布尔值进行输入，Target Location以将值设置为向量进行获取

失去目标事件首先将值设置为对象，而后清空焦距，事件于视觉检测事件中的False事件中进行定时器设定且请求控制器预感事件，视觉事件的True事件的结尾进行清除定时器

**Env Query：**

首先进行场景的导航体网格的配置，而后进行Testing Pawn创建，将建立的EQS场景查询添加到Testing Pawn的查询模板中

场景引导EQS：  
添加Simple Grid，网格半大小：500，之间的空间：250，于EQS Context周围生成

测试：

仅过滤的Distance测试，测试模式为距离2D，过滤选择最小随机浮点值

仅得分的Dot测试，得分指定最大值0.5

EQS Context：重载函数Actor Set，获取类的所有Actor（Bot Interest）返回数组

Bot Interest：任意Actor，放置于场景进行场景引导的路径设置

Combat EQS:

创建Pathing Grid

测试：

仅过滤的Dot测试，过滤类型设置为最大值0.5（绝对值）

仅得分的Distance测试

仅过滤的Trace测试，情境为Target Context，布尔匹配False

仅过滤的Dot测试，线条A旋转Target Context，线条B两点from Context to Item，过滤最大值0.95

Target Context：重载函数Single Actor，获取AI控制器进行是否可用分析，从控制中获取聚焦Actor再次进行是否可用分析，输出为聚焦Actor

Retreat EQS：

创建Simple Grid：

仅过滤的Trace测试，情境为Target Context

仅得分的Distance测试，得分因数-1

仅得分的Distance测试，得分因数0.5

**Behavior Tree：**

关键帧：

Can See Player

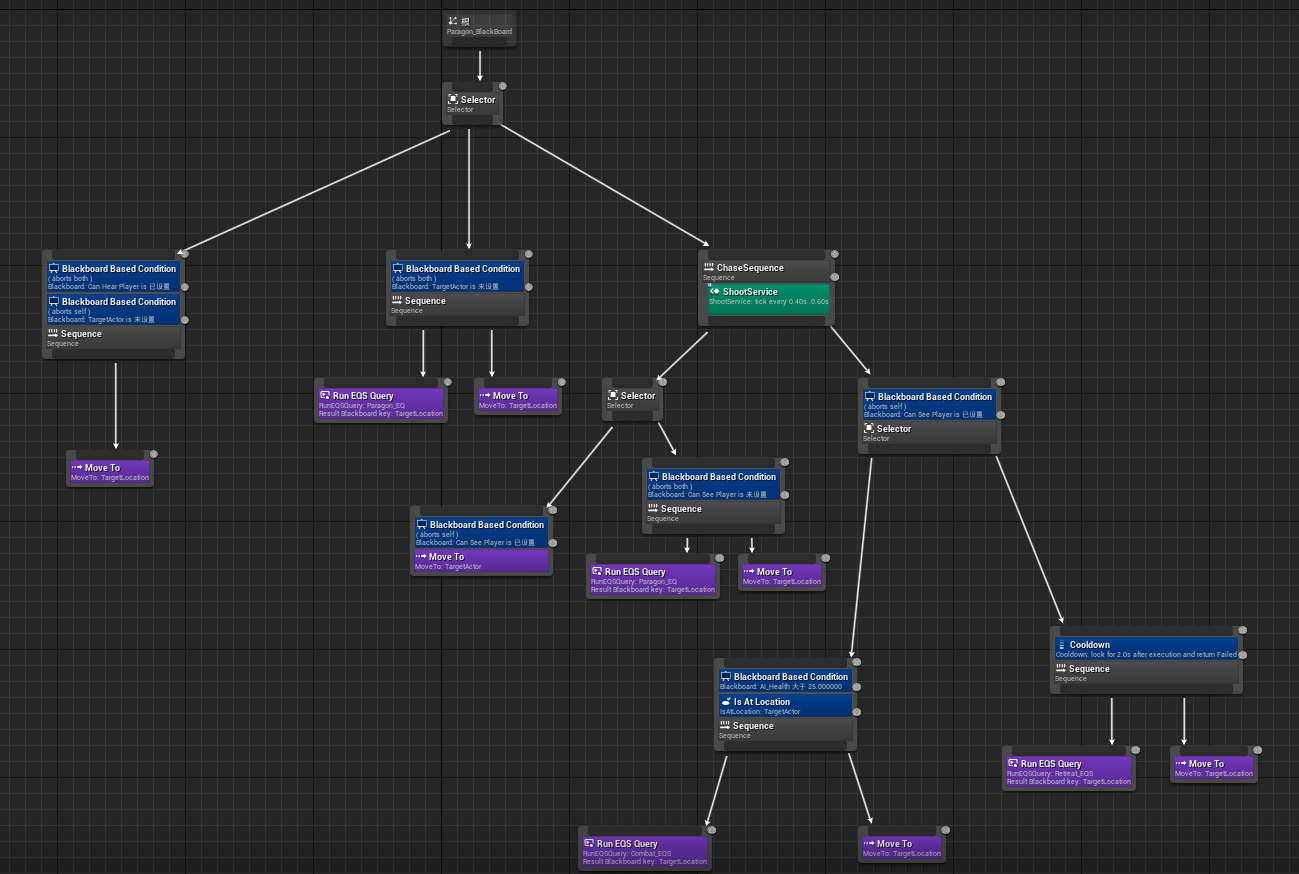
Can Hear Player

Target Actor（Key type：Actor）

Target Location

AI Health

树的本体：



首先使用Selector进行事件拆分，

从左子树开始运行，左子树事件为Hearing事件，用黑板进行布尔获取判断事件，若True则开始Move To操作。

中间为场景的巡逻，通过黑板事件对于Target Actor进行判定，若为False则运行场景引导EQS和Move To事件

右分支为Chase事件的序列，于序列下添加射击事件服务

左选择中，左子树为Move To事件（目标位置），右子树为丢失目标下的巡逻操作（同中分支），右选择首先添加Can see的黑板条件，选择器左子树为AI生命>25情况下运行Combat EQS和Move To事件，右子树为生命值低的情况下的周旋事件，运行Retreat EQS和Move To

注：使用EQS时运行模式请选择最佳5%