

# PS\_AI\_Behavior

---

*Personality-Driven NPC AI for Unreal Engine*

User Documentation

Civilians • Enemies • Protectors • Cover System • Splines

# Plugin Overview

## What it does

- Personality-driven AI for NPCs
- Three archetypes: Civilian, Enemy, Protector
- Trait-based reactions: Courage, Aggressivity, Caution
- Tactical combat + cover system + EQS
- Spline-based patrol networks
- Interface-driven — no Pawn class dependency

## Primary Use Case

*Populate your world with intelligent NPCs that perceive threats, evaluate risk based on personality, and respond appropriately — civilians flee from gunfire, enemies engage in tactical combat, protectors defend civilians.*

## Key Differentiators

- Data-driven personality profiles
- Built-in cover cycle state machine
- Spline network with junction detection
- Gaze bridge to ConvAgent (reflection)

# Core Architecture

*The plugin is built around four concepts:*

## Interface

*IPS\_AI\_Behavior\_Interface*

Implemented on host project Pawns. Zero compile dependency — identity, team, combat, movement.

## NPC Type

*Civilian / Enemy / Protector*

Determines team, spline access, default tree. Enemies can be disguised (hostile=false).

## State

*Behavior State enum*

Idle, Patrol, Alerted, Combat, Fleeing, TakingCover, Dead, Scripted. Written to Blackboard.

## Profile

*PersonalityProfile data asset*

Traits, thresholds, target priority, speed per state, default BT.

# NPC Types & TeamId

*Identity drives perception, targeting, and access control.*

## CIVILIAN

*TeamId = 0x00*

Non-hostile. Flees. Allied with Protector.

## ENEMY

*TeamId = 0x10*

Hostile. Tactical combat. Can be disguised.

## PROTECTOR

*TeamId = 0x20*

Police / guard. Defends Civilians.

### TeamId Encoding

*uint8 = (high nibble: NPCType) | (low nibble: Faction 0-15)*

- Same type + same faction → Friendly (allies)
- Same type + different faction → Hostile (rival gangs)
- Civilian ↔ Protector → always Friendly
- Everything else → Hostile
- Disguised Enemy (hostile=false) → TeamId 0x01 (civilian disguise)

# Behavior States

*Enum written to Blackboard, drives BT branching.*

|                                                                      |                                                                        |                                                                 |                                                                             |
|----------------------------------------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------|
| <div>Idle</div> <div>Resting / waiting. Proximity gaze active.</div> | <div>Patrol</div> <div>Wandering on waypoints or splines.</div>        | <div>Alerted</div> <div>Suspicious, heightened awareness.</div> | <div>Combat</div> <div>Engaged with a threat, attacking.</div>              |
| <div>Fleeing</div> <div>Running away from danger.</div>              | <div>TakingCover</div> <div>Behind tactical cover during combat.</div> | <div>Dead</div> <div>NPC deceased, all systems shut down.</div> | <div>Scripted</div> <div>Manual override (conversations, cinematics).</div> |

# AI Controller

*APS\_AI\_Behavior\_AIController* — the central orchestrator.

## Auto-setup on Possession

- Discovers PersonalityComponent on Pawn
- Computes TeamId from NPCType + Faction
- Initializes Blackboard with required keys
- Binds gaze bridge to ConvAgent (if present)
- Starts BT (unless bAutoStartBehavior = false)

## Key Methods

- StartBehavior() / StopBehavior()
- SetBehaviorState() / GetBehaviorState()
- SetTeamId() — runtime team change
- GetBehaviorPerception()
- GetPersonalityComponent()

### Gaze Bridge (Proximity)

*Reflection-based integration with PS\_AI\_ConvAgent. During Idle/Patrol, NPC glances at nearby actors within GazeProximityRadius. Configurable duration and cooldown. Auto-pauses during active conversations.*

# Personality Component

*Runtime traits & state reactions for the NPC.*

## The Three Traits

### Courage

*0 = coward, 1 = fearless*

Gates flee threshold & cover advancement

### Aggressivity

*0 = peaceful, 1 = violent*

Drives attack threshold, combat/cover cycle

### Caution

*0 = reckless, 1 = prudent*

Flee threshold, cover duration, spline threat

## Properties & Events

- Profile — PersonalityProfile DataAsset
- bAutoStartBehavior — per-NPC override
- RuntimeTraits — editable at runtime
- PerceivedThreatLevel — replicated
- CurrentState — replicated with OnRep
- EvaluateReaction() / ApplyReaction()
- ModifyTrait() / GetTrait()
- ForceState() — scripting override
- OnBehaviorStateChanged delegate

# PersonalityProfile DataAsset

*The behavioral recipe — one profile per NPC archetype.*

## Identity

- ProfileName
- NPCType
- Faction (0-15)

## Traits

- Courage (0-1)
- Aggressivity (0-1)
- Caution (0-1)

## Thresholds

- FleeThreshold
- AttackThreshold
- AlertThreshold
- ThreatDecayRate

## Combat

- TargetPriority[]
- MinAttackRange / MaxAttackRange
- CombatCoverCycleDuration

## Movement

- SpeedPerState map
- DefaultWalkSpeed

## Behavior

- DefaultBehaviorTree (soft ref)

*Example presets: Cowardly Villager • Aggressive Guard • Rival Gang Member*



# Components

*Attach to Pawns as needed — most are optional.*

|                 |                                           |          |
|-----------------|-------------------------------------------|----------|
| Personality     | Traits, profile, state reactions          | Required |
| Perception      | AI Perception wrapper, threat computation | Auto     |
| Team            | Identity for non-AI pawns (player)        | Optional |
| Combat          | Attack events, fire/kill delegates        | Optional |
| Spline Follower | Spline walking, junction detection        | Optional |

# Perception Component

*Sight, hearing, damage senses — auto-configured.*

## Senses

- Sight — vision cone + line of sight
- Hearing — noise events with tags
- Damage — damage received tracking

## Threat Scoring

- Target priority (from Profile)
- Recent damage weight
- Proximity weight
- Target persistence (80% score hold)

## Key Methods

- `GetHighestThreatActor()`
- `CalculateThreatLevel()`
- `GetThreatLocation()`
- `ClassifyActor()`

## Perception / Threat Resolution

*Detects ALL affiliations → filters by Hostile attitude. Supports separate ThreatTarget (e.g. AimTargetActor sphere) vs. owning Pawn. Line-of-sight tracking with last-known-position memory.*

# Combat Component

*Attack execution, cooldown, damage events.*

## Properties

- AttackRange — max distance (cm)
- AttackCooldown — minimum between attacks
- AttackDamage — base damage value
- CurrentTarget — replicated

## Methods & Events

- CanAttack() — cooldown check
- IsInAttackRange() — distance test
- ExecuteAttack() — apply damage + multicast
- OnAttackExecuted (multicast)
- OnDamageReceived (server-only)

## Combat Type

*NPC's combat type (Melee / Ranged) comes from `IPS_AI_Behavior::GetBehaviorCombatType()`. Melee = rush the target (no cooldown between re-positions). Ranged = maintain distance, use cover cycle, peek to fire.*

# World Actors

*Placed by level designers — drive navigation and tactics.*

## SplinePath

Placeable spline defining a navigation corridor. Filtered by NPCType. Bidirectional, with priority for junction selection.

## SplineNetwork

WorldSubsystem — auto-detects intersections between SplinePath actors. Provides navigation queries, threat-aware switching.

## CoverPoint

Manually placed tactical position. Cover or HidingSpot type, quality score, crouch flag, max occupants, type filter.

# Behavior Tree — Services & Tasks

Reusable BT nodes — build custom trees per NPC type.

## Services (root-level)

|                  |                                           |
|------------------|-------------------------------------------|
| UpdateThreat     | Queries perception, writes threat BB keys |
| EvaluateReaction | Computes state from traits + threat       |
| UpdateGaze       | Manages eye contact, ConvAgent bridge     |

## Tasks (leaves)

|                     |                               |
|---------------------|-------------------------------|
| FollowSpline        | Drives SplineFollower         |
| FindAndFollowSpline | Auto-locate + follow          |
| Patrol              | Cycle PatrolPoints            |
| FindCover           | Manual + procedural cover     |
| CoverShootCycle     | Cover → peek → fire cycle     |
| Attack              | Chase + attack (Melee/Ranged) |
| FleeFrom            | Move away from threat         |

# Cover System

*Manual CoverPoints + procedural raycast fallback + EQS refinement.*

## Manual CoverPoints

- Placed by level designer
- Type: Cover or HidingSpot
- Quality score (0–1)
- MaxOccupants, Crouch flag
- Type filter (Any / Civ / Enemy / Protector)

## Procedural

- Fallback if no manual points
- Raycast sampling around threat
- LOS blockage test
- Advancement bias toward threat

## EQS Refinement

- OnCircle generator around cover
- LineOfSight filter
- Distance + quality tests
- Optional firing position refine

## Cover Cycle State Machine (CombatSubState)

Engaging

AtCover

MovingToFire

Peeking

ReturningToCover

Advancing

# IPS\_AI\_Behavior\_Interface

*What your Pawn must implement (C++ or Blueprint).*

## Identity

- `GetBehaviorNPCType()`
- `SetBehaviorNPCType()`
- `IsBehaviorHostile()`
- `SetBehaviorHostile()`

## Movement

- `SetBehaviorMovementSpeed()`
- `GetBehaviorMovementSpeed()`
- `SetBehaviorCrouch()`

## State Feedback

- `OnBehaviorStateChanged()`
- `IsTargetActorValid()`
- `GetBehaviorThreatActor()`

## Combat

- `BehaviorStartAttack()`
- `BehaviorStopAttack()`
- `CanBehaviorAttack()`
- `GetBehaviorCombatType()`

# Setup Workflow

*From zero to a behaving NPC in 6 steps.*

1

## Personality Profile

*Create DataAsset — traits, thresholds, DefaultBehaviorTree.*

2

## Implement Interface

*Pawn: implement IPS\_AI\_Behavior\_Interface (identity, combat, movement).*

3

## Add Components

*Pawn: add PersonalityComponent. Optional: Combat, SplineFollower.*

4

## Behavior Tree

*Branch on BehaviorState. Add services UpdateThreat + EvaluateReaction at root.*

5

## Set AIController

*Pawn's AIController class = APS\_AI\_Behavior\_AIController.*

6

## Place World Actors

*Optional: SplinePaths, CoverPoints, SplineNetwork (auto).*



# Blackboard Keys

*Auto-created by the AIController — use in your BT decorators and tasks.*

|                |        |                                       |                         |        |                               |
|----------------|--------|---------------------------------------|-------------------------|--------|-------------------------------|
| BehaviorState  | Enum   | Current state (Idle, Combat, etc.)    | ThreatActor             | Object | Current highest-scored threat |
| ThreatLocation | Vector | Last known threat position            | ThreatLevel             | Float  | Perceived threat (0–1+)       |
| CoverLocation  | Vector | Target cover position                 | CoverPoint              | Object | Current CoverPoint actor      |
| PatrolIndex    | Int    | Current patrol waypoint index         | HomeLocation            | Vector | Spawn / home point            |
| CurrentSpline  | Object | Active SplinePath                     | SplineProgress          | Float  | Distance along current spline |
| CombatSubState | Enum   | Cover cycle sub-state (Peeking, etc.) | LastKnownTargetPosition | Vector | LOS-lost target memory        |
| PreferCover    | Bool   | Combat/cover cycle toggle             | ConversationPaused      | Bool   | Active dialogue pause flag    |

# Networking

*Server-authoritative AI, minimal replication for visuals.*

## Server-only

- All AI planning & decision-making
- Threat evaluation, state transitions
- Perception stimuli processing
- Path selection, cover choice
- Combat targeting

## Replicated to clients

- CurrentState (OnRep fires delegate)
- PerceivedThreatLevel (HUD/debug)
- CurrentTarget (animations/visuals)
- CurrentSpline, bIsFollowing
- OnAttackExecuted (multicast)

*Character movement syncs via CharacterMovementComponent. Spline follower updates server-side, position replicates naturally. Clients receive enough state for animation and HUD without any AI logic running locally.*

# Integration with PS\_AI\_ConvAgent

*Optional — behavior and conversation plugins talk via reflection.*

## Gaze Bridge

AIController discovers GazeComponent on the Pawn, sets target during Idle/Patrol (proximity gaze) and forwards current threat during Combat.

## Conversation Pause

When ConvAgent is conversing, AI planning pauses — NPC stops moving, waits. Local VAD detects user speech.

## Disguise Reveal

Enemy NPC with IsBehaviorHostile=false starts as Civilian TeamId. Flips to hostile via interface when triggered by dialogue.

## Forced State

ConvAgent LLM can invoke tools that call ForceState(Scripted) or ForceState(Fleeing) for narrative control.

# Debug Tools

*Visualize state, perception, and targeting at runtime.*

## Per-NPC Debug

- `PersonalityComponent.bDebug` — floating text
- Name, NPCType, TeamId, State
- ThreatLevel, ThreatActor
- Hostile, active spline
- SplineFollower debug sphere (green)

## Global Logging

- CVar: `ps.ai.Behavior.Debug 1`
- Toggles verbose logs globally
- `LogPS_AI_Behavior` category
- BTVue / AIDebug (Unreal built-in)
- EQS Visualizer for cover queries

### Pro tip

*Enable `bDebug` only on specific NPCs (or via a debug HUD key). Global debug logs can be overwhelming in crowded scenes. The floating text is cheap but accumulates.*

# Summary

*Complete NPC behavior stack for tactical game AI.*

## Personality-driven

*Three traits modulate every decision*

## Spline navigation

*Network with junction detection, type filtering*

## Server-authoritative

*Minimal replication, client just animates*

## Cover-aware

*Manual CoverPoints + procedural fallback + EQS*

## Perception built-in

*Sight + hearing + damage, priority-based threat*

## Plugin-friendly

*Gaze bridge to ConvAgent, interface-only dependency*