Overview

Taximus is the Tax Officer antagonist and the central threat of the game. He is not a simple enemy that chases the player on sight. He is a **rule-following authority figure** who operates on a priority-based behavior tree, responding to the player's choices in a logical and escalating way. The design goal was to make Taximus feel like a real investigator: he wanders when nothing is wrong, grows suspicious when things seem off, and launches a structured inspection process when evidence crosses a threshold. Every part of his behavior is driven by data that the player themselves creates through their own decisions.

Taximus is built on three interconnected systems: a **Behavior Tree (BT_Taximus)** that controls his decision making, a **Detection System** that reads the player's actions in real time, and an **Audit System** that executes the game's primary consequence loop. These three systems are described in full below.

The Behavior Tree (BT_Taximus)

Taximus uses a **Selector node at the root of his behavior tree**, meaning he evaluates each branch from top to bottom and executes the first one whose condition is true. This creates a clean priority hierarchy where distraction always beats auditing, auditing always beats patrolling, and patrolling only runs when nothing else is happening. A **BTS_UpdateSuspicion service** runs on the root selector every 1.9 to 2.1 seconds, continuously reading the player's suspicion value and proximity to decide whether to escalate his state.

Priority	Branch	Condition (Blackboard)	What Taximus Does
1 (Highest)	**Distraction**	bIsDistracted is set to true	Stops all activity, waits 10 seconds, clears the flag and resumes
2	**Active Audit**	bIsAuditing is set to true	Finds a hiding spot, navigates to it, performs a discovery roll, repeats until quota is met
3	**Start Audit**	bShouldStartAudit is set to true	Triggers the warning UI, sets audit state, picks a random inspection quota of 2 to 5 spots
4 (Default)	**Patrol**	None (always runs if above conditions are clear)	Waits 4 seconds, picks a random navigable point within 1000 units and walks to it

Blackboard Keys

All of Taximus's state is stored in a Blackboard asset called BB_Taximus. Behavior tree tasks and services read and write to these keys to communicate with each other without being directly connected. This keeps each task self-contained and easy to debug or replace independently.

Blackboard Key	Type	Purpose
SelfActor	Object	Reference to the Taximus actor itself, used internally by the behavior tree
TargetHidingSpot	Object	The current hiding spot Taximus is navigating to during an audit. Set by BTT_FindHidingSpot, cleared after inspection.
bIsDistracted	Boolean	Set to true when the Dog companion distracts Taximus. Triggers the highest priority branch, halting all other behavior for 10 seconds.
bShouldStartAudit	Boolean	Set to true when suspicion triggers an audit event. Causes Taximus to run BTT_StartAudit on the next behavior tree tick.
bIsAuditing	Boolean	Active during the inspection loop. While true, Taximus continuously finds and inspects hiding spots until the quota is met.
AuditSpotsInspected	Integer	A counter that increments after each inspection. BTT_CheckAuditComplete compares this to MaxSpotsToInspect to decide when to end the audit.
MaxSpotsToInspect	Integer	A random integer between 2 and 5, rolled at the start of each audit. Creates unpredictable audit pressure across every session.

The Detection System

Taximus detects suspicious activity through two parallel mechanisms that run at the same time: an **Awareness Zone** that reacts to actor tags in real time, and a **Behavior Tree Service (BTS_UpdateSuspicion)** that ticks on a timer and checks proximity and world state. Together they cover both immediate events and ongoing pressure.

Tag-Based Awareness Zone

Taximus has a sphere collision component on his character with a radius of 600 units. This is his awareness zone. The component fires **AC_SuspicionDetector** on every overlap event. This component reads the actor tag of whatever entered the zone and applies a suspicion increase based on what it finds.

The tag system works because it is completely decoupled from the player character. Taximus does not look at the player directly. He looks at the world and reads what is in it. A gold pouch sitting on the floor has the **evidence** tag. A player crouching near a hiding spot has the **sus** tag. Taximus reacts to tags, not to the player, which means the player can manage their threat level by managing what tags exist in the world around Taximus.

Detection Source	Tag	Suspicion Added	How It Works
Player near Taximus	None	+1 per tick	BTS_UpdateSuspicion ticks every 2 seconds. If player is within 600 units, suspicion rises by 1.
Player at a hiding spot	sus	+15 flat	Interacting with a hiding spot applies the sus tag. If Taximus detects this inside his awareness zone, suspicion rises by 15.
Gold pouch left in the open	evidence	Gold value divided by 100	An unattended gold pouch automatically receives the evidence tag. Taximus reads the gold amount from the pouch and scales the suspicion increase accordingly.
Untaxed sale processed	None	+15 flat	Every time the player hides a sale through WBP_JournalDecision, the game mode calls UpdateSuspicion directly with a base value of 15.

The Audit System

An audit is triggered when suspicion reaches 50 or above and a probability roll succeeds. The roll compares a random float between 0 and 100 against the current suspicion value. At exactly 50 suspicion the audit has a 50% chance of firing on any given check. At 100 suspicion, an audit fires instantly with no roll at all.

Once triggered, the audit runs through four phases in sequence. The player cannot stop the process once it begins, only reduce how damaging it is by preparing during the warning window.

Phase	Name	Duration	What Happens
1	**Warning**	120 seconds	WBP_AuditWarning appears on screen. The player can still move, sell, and

Phase	Name	Duration	What Happens
			interact. This is their only window to secure hiding spots before inspection begins.
2	**Inspection**	Per spot	Taximus navigates to a randomly selected hiding spot. BTT_FindHidingSpot picks from all BP_HidingSpot actors in the level. He will inspect between 2 and 5 spots total per audit.
3	**Discovery Roll**	Instant	At each spot, the system calculates a discovery chance using the formula below. A random float is generated and compared to the result. This determines whether the stash is found.
4	**Consequences**	Instant	If discovered, Taximus chases the player. Catching the player results in full seizure of all untaxed gold plus a 50% fine on taxed gold. If the player escapes, no fine is applied and suspicion resets to 0.

Discovery Chance Formula

The core of the audit system is the discovery roll performed at each hiding spot. The formula is:

```
Discovery Chance = ((Stashed Gold / 10) minus Security Bonus) x
                   Spot Modifier, clamped 0 to 100
```

The Spot Modifier is **0.25** if the player secured the hiding spot before the audit began, reducing discovery chance by 75%, and **1.0** if the spot was left unsecured. The Security Bonus comes from the player's skill path upgrades and is subtracted directly from the base chance before the modifier is applied. This means a player with a high Security Bonus can reach zero discovery chance on secured spots even when gold amounts are significant.

Every hiding spot gets its own independent roll. A player with five hiding spots in the house faces five separate dice rolls during a single audit. The random number of spots inspected per audit (2 to 5) adds another layer of unpredictability: the player never knows how many spots Taximus will check, so leaving even one spot unsecured is a genuine risk.

Consequences and the Chase

If Taximus finds gold during a discovery roll, he does not immediately apply a fine. Instead, he enters a **chase phase**. This is a deliberate design choice: the player still has a chance to escape even after being caught with evidence. If Taximus catches the player during the chase, the consequences are severe: **all untaxed gold is seized in full, and an additional fine equal to 50% of the seized amount is deducted from the player's taxed gold.**

If the player escapes the chase, no fine is applied and suspicion resets to 0. This outcome rewards skillful play and gives the player a meaningful reason to invest in movement and distraction tools like the Dog companion. It also means that a successful escape after a failed audit is not just a reprieve. It is a full reset, letting the player rebuild from a clean slate.

The Dog Companion as a Counter

The only way to directly interrupt Taximus's behavior tree mid-execution is the **Dog companion**. When the player activates the dog, it navigates to Taximus and calls the **Event_GetDistracted** event, which sets **bIsDistracted** to true on the Blackboard. Because distraction is the highest priority branch in the tree, this immediately overrides whatever Taximus was doing, including active inspection. He stops, waits 10 seconds, clears the flag, and then resumes from the top of the tree.

This interaction was designed to give the player one emergency tool that is powerful but limited. The dog can only be used once per feeding cycle, and taming it requires a bone item. Using it at the wrong moment means wasting a rare resource. Using it at the right moment can save the entire stash.