The Four Brains of the Ambient System

The ambient system has four distinct roles:

1. **AmbientEmitter**: sound sources in the world
2. **AmbientProfile**: soundscape definitions (mix presets)
3. **AmbientZone:** triggers that activate profiles
4. **AudioManager:** the conductor

---

1. AmbientEmitter — “I exist and I play a track”

Each AmbientEmitter:

- Has an AudioSource
- Has a **track name** (string or enum like "firesound")
- Registers itself with AudioManager on Start

Conceptually:

“Hi AudioManager, I am track = firesound, and here is my AudioSource.”

AudioManager stores this in something like:

trackName → list of emitters

So at runtime it knows:

- firesound → [emitter A, emitter B]
- birdssound → [emitter C]

---

2. AmbientProfile — “This is what this place should sound like”

This is the soul of the system.

An AmbientProfile is a **data asset** that contains layers, like:

Layer:

- Track = firesound

- TargetVolume = 1.0
- FadeTime = 2.0

Another layer:

- Track = birdssound
- TargetVolume = 0.3
- FadeTime = 4.0

Conceptually, a profile says:

“In this area, here is the desired mix.”

---

3. AmbientZone — “When player enters, use this recipe”

The zone is just a trigger with a reference to an AmbientProfile.

OnTriggerEnter(player):

AudioManager.ApplyProfile(myProfile)

The zone does NOT care about emitters.
It just says: use this soundscape now.

---

4. AudioManager — “Apply the mix to the real world”

This is where the interesting logic happens.

When a profile is applied, AudioManager:

Step 1 — Build desiredVolumeByTrack

It reads the profile and builds a dictionary:

firesound → 1.0

birdssound → 0.3

wind → 0.0

Step 2 — Match profile tracks to emitters

For each registered emitter:

- Look at emitter.Track
- Look up desiredVolumeByTrack[Track]
- If found → fade AudioSource to that volume
- If NOT found → fade to 0 (or ignore)

This is how zones affect emitters anywhere in the scene.

Why emitters do NOT need to be inside zones

Because zones do NOT search the scene for emitters, the AudioManager already knows all emitters globally, zones just change the global mix.

- One fire sound can be heard across multiple zones
- A far-away waterfall can fade in gradually
- Etc

How multiple zones really work

Imagine player walking:

Forest Zone → Camp Zone → Cave Zone

Each zone applies a different profile:

ForestProfile:

- birds 1.0
- wind 0.4
- fire 0.0

CampProfile:

- birds 0.2
- wind 0.1
- fire 1.0

CaveProfile:

- birds 0.0
- wind 0.0
- drip 0.6

Each time:

- Same emitters
- Different recipe
- AudioManager fades volumes accordingly

Support

For further support, contact **snogdev@gmail.com**