

PS_AI_ConvAgent

Conversational AI for MetaHumans in Unreal Engine

User Documentation

ElevenLabs • MetaHuman • Listen Server

Plugin Overview

What it does

- Full conversational AI NPC integration for Unreal Engine
- Real-time voice capture, turn-taking, TTS playback
- Emotion-driven MetaHuman animations (face, body, gaze)
- Multiplayer support with speaker arbitration
- Tool system for LLM-triggered game actions

"Drop components on a MetaHuman, configure a few data assets, and talk."

Primary Use Case

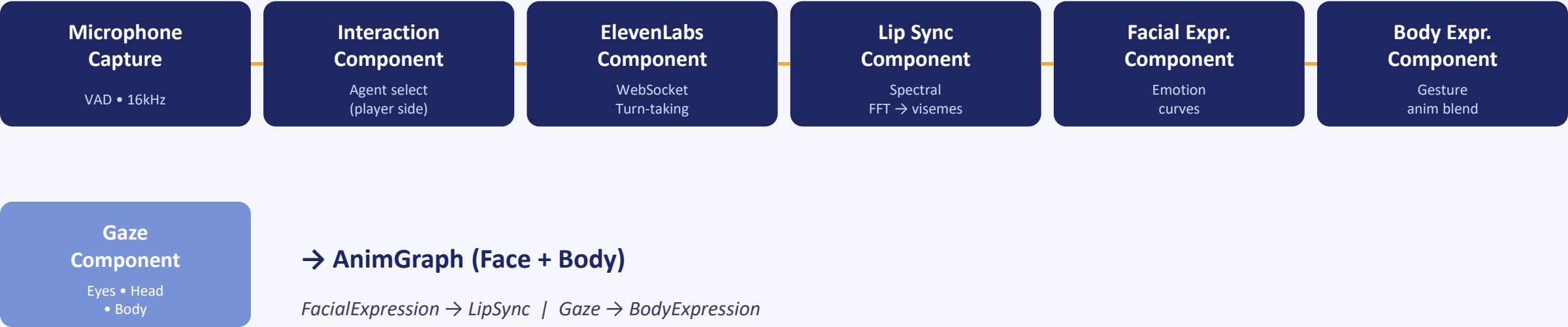
Interactive, emotionally expressive voice NPCs that listen to players, respond with contextual answers, express emotion through facial animation and body language, and trigger physical actions based on conversation flow.

Key Differentiators

- Full-duplex Server VAD or push-to-talk
- Adaptive pre-buffering & speculative TTS
- Spectral lip sync + artist-driven poses
- Opus codec + distance LOD for multiplayer

Architecture

Component chain on a MetaHuman NPC



PROJECT SETTINGS — API Key • Server Region • Global Context Prompt

Data Assets: AgentConfig • EmotionPoseMap • LipSyncPoseMap • BodyPoseMap • ActionSet • Tool

ElevenLabs Component

The core conversation engine — WebSocket, turn-taking, audio pipeline.

Key Properties

- AgentConfig — identity, voice, tools
- TurnMode — ServerVAD / Client (push-to-talk)
- MicChunkDurationMs (50–250 ms)
- AudioPreBufferMs / bAdaptivePreBuffer
- bAllowInterruption, bPersistentSession
- SoundAttenuation (optional 3D audio)
- AudioLODCullDistance / LipSyncLODDistance

Key Methods

- StartConversation() / EndConversation()
- StartListening() / StopListening()
- SendTextMessage() — text without voice
- InterruptAgent() — force stop mid-speech
- ForceDisableConversation(Action) — blend to neutral
- ForceEnableConversation() — re-enable
- FeedExternalAudio() — external mic source

Major events: OnAgentConnected • OnAgentTranscript • OnAgentTextResponse • OnAgentStartedSpeaking • OnAgentEmotionChanged • OnAgentActionRequested • OnActiveSpeakerChanged • OnReadyForAction

Microphone Capture Component

Captures, resamples, and performs local voice activity detection.

Audio Pipeline

- WASAPI 48kHz stereo float32 input
- Resample to 16kHz mono float
- Convert to int16 LE byte stream
- Delivered via multicast delegate

Voice Activity Detection

- VolumeMultiplier — mic gain
- VoiceActivityThreshold — RMS threshold
- VoiceOnsetTime — debounce on speech start
- VoiceSilenceTime — debounce on end
- OnUserVoiceActivityChanged event

Public API

StartCapture() • StopCapture() • IsCapturing()

The component auto-opens the default input device. Output is consumed by ElevenLabsComponent (or InteractionComponent, which relays via Server RPC in networked games).

Lip Sync Component

Spectral analysis of agent audio → 15 OVR visemes → ARKit blendshapes.

How it works

- FFT on agent audio stream
- Estimates 15 OVR viseme weights
- Maps to ARKit blendshapes (jawOpen, mouthFunnel...)
- Optional: custom PoseMap with artist-crafted poses
- Thread-safe curve output for AnimGraph

Key Properties

- LipSyncStrength — overall intensity
- SmoothingSpeed (35–65) — viseme interp
- SpeechBlendDuration — fade in/out
- EmotionExpressionBlend (0–1) — emotion bleed through mouth
- EnvelopeAttackMs / EnvelopeReleaseMs
- PoseMap — optional artist override

AnimGraph placement (Face AnimBP): Live Link → FacialExpression → LipSync → mh_arkit_mapping_pose

Facial Expression Component

Emotion-driven face animation via real-time AnimSequence sampling.

Design

- Plays AnimSequences in real-time (looping)
- Extracts emotion curves (eyes, brows, cheeks, mouth mood)
- Injects into AnimGraph via FacialExpression AnimNode
- Lip sync overrides mouth area during speech
- 3 intensity levels: Normal / Medium / Extreme
- Crossfades between emotions

Properties

- EmotionPoseMap — data asset of AnimSequences
- ActivationBlendDuration (default 0.5s)
- EmotionBlendDuration — crossfade time
- bActive — auto-managed by conversation

Emotions covered

Joy • Sadness • Anger • Fear • Surprise • Disgust • Neutral

Body Expression Component

Upper-body gestures and posture driven by emotion + speaking state.

Animation lists per emotion

Idle	Played while listening (not speaking)
Normal	Low-intensity speaking anims
Medium	Medium-intensity speaking
Extreme	High-intensity speaking

Key Properties

- BodyPoseMap — emotion → anim lists
- ActivationBlendDuration (0.5s)
- EmotionBlendDuration (1.0s)
- BlendWeight (0–1)

AnimNode behaviour

- Override mode, per-bone mask
- Default BlendRootBone = spine_02
- Excludes neck_01 (Gaze handles it)
- Lower body passes through upstream

Gaze Component

Multi-layer look-at: body • head • eyes.

BODY

full 360°

Actor yaw — follows target around the character

HEAD

limited

Head bone yaw + pitch, clamped

EYES

limited

ARKit eye curves, fine targeting

Key Properties

- TargetActor — who to look at
- MaxHeadYaw / MaxHeadPitch
- MaxEyeHorizontal / MaxEyeVertical
- BodyInterpSpeed / HeadInterpSpeed / EyeInterpSpeed
- HeadAnimationCompensation (0.9)
- EyeAnimationCompensation (0.6)
- BodyDriftCompensation (0.8)
- NeckBoneChain — multi-bone distribution

Interaction Component (Player)

Placed on the player pawn — selects NPCs, routes mic, manages gaze.

Selection Logic

- Proximity + view cone detection
- MaxInteractionDistance (300 cm default)
- ViewConeHalfAngle (45° default)
- SelectionStickyAngle — hysteresis
- ConversationSwitchDelay — anti-flicker

Auto-management

- bAutoStartConversation — open WS on select
- bAutoManageListening — mic open/close
- bAutoManageGaze — set agent gaze to player
- bAutoPassiveGaze — out-of-range passive tracking

Events: OnAgentSelected • OnAgentDeselected • OnNoAgentInRange

Methods: GetSelectedAgent() • ForceSelectAgent() • StartConversationWithSelectedAgent() • SendTextToSelectedAgent() • ClearSelection()

Data Assets

Reusable configuration assets for agent identity, animations, and tools.

AgentConfig

Agent identity, voice, LLM model, tools

EmotionPoseMap

Emotion → facial AnimSequences (3 levels)

LipSyncPoseMap

15 OVR visemes → mouth poses

BodyPoseMap

Emotion → body gesture anim lists

ActionSet

Actions the agent can trigger

Tool

Custom LLM tools with parameters

AgentConfig (ElevenLabs)

Complete agent definition — editable in Content Browser with API buttons.

Identity

- AgentID
- AgentName

Voice

- VoiceID / VoiceName
- TTSMModelID
- Stability • Similarity • Speed

LLM

- LLMModel (gemini-2.5-flash default)
- Language (en default)

Behavior

- CharacterPrompt (personality)
- FirstMessage (greeting)
- TurnTimeout, MaxTurns

Latency

- TurnEagerness (Eager/Normal/Patient)
- bSpeculativeTurn

Tools

- Tools[] array of Tool assets
- ActionSet integration

Editor: Voice picker • Model picker • LLM picker • Create / Update / Fetch Agent buttons

Pose Maps

Three pose-map data assets map abstract states → AnimSequences.

EmotionPoseMap

*Emotion → facial AnimSequences
(3 intensity levels per emotion)*

- Joy (N / M / E)
- Sadness (N / M / E)
- Anger (N / M / E)
- Fear (N / M / E)
- Surprise (N / M / E)
- Disgust (N / M / E)
- Neutral (N / M / E)

LipSyncPoseMap

*15 OVR visemes →
phoneme AnimSequences*

- sil • PP • FF • TH • DD
- kk • CH • SS • nn • RR
- aa • E • ih • oh • ou
- MetaHuman MHF_* recommended
- Shareable across characters

BodyPoseMap

*Emotion → body anim lists
(Idle / N / M / E per emotion)*

- Idle: listening animations
- Normal: light speaking
- Medium: expressive speaking
- Extreme: emphatic speaking
- Random selection per loop
- Continuous variety

Tools & ActionSets

LLM-triggered actions that bridge conversation → gameplay.

Tool asset

- ToolName — snake_case identifier
- ToolDescription — when LLM should use it
- PromptFragment — appended system prompt
- Parameters[] — typed LLM arguments
- ToolID auto-populated on Create

Built-in Tools

- set_emotion(emotion, intensity)
- perform_action(action) — from ActionSet
- Custom tools — any parameters

Action flow

1. LLM decides to call a tool (e.g. `perform_action("flee")`)
2. Plugin fires `OnAgentActionRequested("flee")`
3. Game calls `ForceDisableConversation("flee")` → animations blend to neutral
4. `OnReadyForAction("flee")` fires → play gameplay montage
5. Game calls `ForceEnableConversation()` when done

AnimGraph Integration

Four AnimNodes auto-discover components — no manual wiring.

Face AnimBP



Body AnimBP



Why this order?

Emotion curves set mood (brows, eyes, cheeks). Lip sync overrides mouth during speech. Gaze head rotation runs before body expression so it doesn't fight arms.

Project Settings

Project Settings → Plugins → PS AI ConvAgent - ElevenLabs

API_Key	ElevenLabs API key — required for authentication
ServerRegion	Global / US / EU / India
CustomWebSocketURL	Override WebSocket URL (advanced)
bVerboseLogging	Verbose WebSocket logging (development)
GlobalContextPrompt	Shared context injected as dynamic variables for all agents

Setup Workflow

From zero to a talking MetaHuman in 5 steps.

1

Project Settings

Enter API_Key, pick ServerRegion.

2

Data Assets

Create AgentConfig, EmotionPoseMap, LipSyncPoseMap, BodyPoseMap.

3

NPC Components

Add ElevenLabs + Mic + LipSync + FacialExpr + BodyExpr + Gaze on the MetaHuman.

4

AnimGraphs

Wire AnimNodes in Face + Body AnimBPs. Components auto-discover.

5

Player Pawn

Add InteractionComponent. Bind selection events to UI. Play.

Networking & Multiplayer

Listen Server model — server owns WebSocket, clients relay via RPCs.

Audio Pipeline

- Client mic → Server RPC (Opus-encoded)
- Server forwards to ElevenLabs WebSocket
- Agent audio → Multicast RPC to all clients
- Clients decode Opus → procedural sound wave

Level of Detail (LOD)

- AudioLODCullDistance — skip audio at distance
- LipSyncLODDistance — skip lip sync at distance
- Active speaker always full quality

Speaker Arbitration

- SpeakerSwitchHysteresis — min silence before switching
- SpeakerIdleTimeout — clear active speaker
- NetActiveSpeakerPawn replicated for gaze
- Only active speaker's audio forwarded to LLM

Replicated State

- bNetIsConversing
- NetConnectedPawns[]
- CurrentEmotion + CurrentEmotionIntensity

ForceDisableConversation Flow

Clean handoff between conversation and gameplay actions.

LLM Tool Call

→ *perform_action("flee")*

OnAgentActionRequested

→ *Event fires on ElevenLabs component*

ForceDisableConversation

→ *Blend out face/body/gaze animations*

OnReadyForAction

→ *All components have reached neutral*

Play Gameplay

→ *Montage / behavior switch*

ForceEnableConversation

→ *Resume conversation when done*

Events Cheat Sheet

Blueprint-assignable delegates on ElevenLabs component.

Connection

- `OnAgentConnected`
- `OnAgentDisconnected`
- `OnAgentError`

Dialogue

- `OnAgentTranscript`
- `OnAgentTextResponse`
- `OnAgentPartialResponse`
- `OnAgentStartedGenerating`

Audio

- `OnAgentStartedSpeaking`
- `OnAudioPlaybackStarted`
- `OnAgentStoppedSpeaking`
- `OnAgentInterrupted`

State / Actions

- `OnAgentEmotionChanged`
- `OnAgentActionRequested`
- `OnAgentClientToolCall`
- `OnActiveSpeakerChanged`
- `OnReadyForAction`
- `OnAgentResponseTimeout`

Summary

A complete conversational AI stack for Unreal Engine.

Real-time

Spectral lip sync, speculative TTS, adaptive pre-buffering

Expressive

Emotion-driven face/body animation via AnimSequences

Multiplayer

Listen Server + Opus codec + distance LOD + speaker arbitration

Data-driven

Reusable PoseMaps, AgentConfigs, Tools, ActionSets

Bridgeable

Behavior plugin gaze bridge, ForceDisable handoff for gameplay

BP-friendly

All components blueprintable, rich event system