Positional Chat

**This sample is compatible with the May 2018 Xbox One XDK.**

# Description

This sample demonstrates how to route incoming audio from GameChat2 for further audio processing such as mixing with local audio and 3D.

# Using the sample

This sample uses the controls described in the following table.

|  |  |  |
| --- | --- | --- |
| **Action** | | **Gamepad Control** |
| Rotate Emitter Counter-clockwise | Left DPad | | |
| Rotate Emitter Clockwise | Right DPad | | |
| Move Emitter | Right Thumbstick | | |
| Switch Reverb Settings | DPad Up/Down | | |

# Implementation notes

This sample is built off of the existing InGameChat sample to handle the basic connection needs for GameChat2. After setting up the UI and input, it instantiates XAudio2 and creates a default mastering voice. It then initializes X3DAudio and sets up a listener, emitter, and DSP settings. For the emitter, an attenuation curve is set (see below) and a cone is provided. As part of the setup process, it also creates two source voices, one for chat audio and one for local audio. All of the PCM data from the short, looping WAV file is read into one buffer and submitted to XAudio2 for playback on a continuous loop. Volume is set at zero and only increases when the emitter is far enough away (inner circle).

For chat, a circular buffer is populated with 16-bit, mono PCM audio for every decoded sample from GameChat2. The source voice for chat is fed muted audio until a chat buffer is ready, which is when the chat audio is fed to that voice.

The listener stays static in the center of the screen (shown in white) with the arrow pointing to the direction the listener is facing. You may change the listener’s direction with the triggers. The emitter (used only for chat) starts near the center of the screen (shown in green) and can be moved in 2D space using the right thumbstick and rotated using the shoulder buttons. The lines in front of the emitter show the cone and the ring shows the end of the attenuation curves (ie distance = 1 in the graph below).

# Notes on audio rendering and mixing

This sample uses XAudio2 for audio rendering to the default endpoint. You can also use WASAPI or any audio middleware solution you would like (such as FMOD or Wwise) just as easily. For an example on how to use other renderers and routing, see the “Middleware WASAPI Capture” sample here: <https://developer.xboxlive.com/en-us/platform/development/education/Pages/Samples.aspx#Audio>

You can also mix audio directly to the headset using GameChat2 alone. See the InGameChat sample for a basic idea of how to alter the decoded chat buffer before it is rendered to a headset. And while this sample treats all chat audio as a single source, you can also use the callback to attach audio buffers to specific users.

# Known Issues

* Sample only supports the default 24kHz from the official headset. Other input devices might have different sample rates.
* Sample does not support switching between input devices
* Sample is designed for chat between two players. Any additional players will be rendered from the same endpoint.