

Introduction into Game Programming (CSC329)

– Sound –

Ubbo Visser

Department of Computer Science
University of Miami

Content taken from <http://docs.unity3d.com/Manual/>

March 19, 2024

Outline

- 1 Audio overview
 - Basic Theory
 - Working with Audio Assets
 - Audio Recording
 - Audio Files
 - Tracker Modules
- 2 Audio Reference
 - Audio Clip
 - Audio Listener
 - Create a new Audio Source
 - Reverb Zones
 - Microphone
- 3 Coroutines

Audio overview

Audio overview

- Background music or sound effects necessary for games.
- Unity's audio system is flexible and powerful.
- Most standard audio file formats can be used.
- Filters can be used as well, e.g. effects like echo.



Basic Theory

Basic Theory

- Sounds are emitted by objects and heard by listeners in the real world.
- Unity requires sounds to originate from Audio Sources attached to objects.
- Sounds then picked up by an Audio Listener attached to another object (e.g. main camera).
- Unity then simulates effects of a sources distance and position from the listener object and play them.
- Doppler Effect for added realism.
- Adding Audio Filters to objects for echos.
- Audio Source, Audio Listener, the audio effects and Reverb Zones give more information.

Working with Audio Assets

Audio Assets

- Audio files in AIFF, WAV, MP3 and Ogg.
- Importing an audio file creates an Audio Clip.
- Clip on Audio Source or used from a script.
- Tracker modules for music supported.
- Use short audio samples as instruments that are then arranged to play tunes.

Audio Recording

Audio Recording

- Access computers microphone from a script and create Audio Clips by direct recording.
- `Microphone` class: API to find available microphones, query capabilities and start and end a recording session.

Audio Files

Audio Files

- Audio in Unity is either *Native* or *Compressed*
- Native: Use Native (WAV, AIFF) audio for short sound effects. The audio data will be larger but sounds wont need to be decoded at runtime.
- Compressed: The audio data will be small but will need to be decompressed at runtime.
- Clips must be used in conjunction with Audio Sources and an Audio Listener in order to actually generate sound.
- Audio Source component has Volume, Pitch and numerous other properties.
- Only **one** Audio Listener in scene allowed (e.g. Main Camera).

Tracker Modules

Tracker Modules

- Packages of audio samples that have been modeled, arranged and sequenced programatically.
- Similar to MIDI files.
- Tracks contain information about when to play the instruments.
- Pro: Melody and rhythm of the original tune can be recreated.
- Cons: sounds are dependent on the sound bank available in the audio hardware.
- Can sound different on different computers.
- 4 most common module file formats supported: Impulse Tracker (.it), Scream Tracker (.s3m), Extended Module File Format (.xm), and the original Module File Format (.mod).
- Most popular tools to create and edit Tracker Modules are MilkyTracker for OSX and OpenMPT for Windows.

Audio Components

Audio Clip

- Audio Clip: mono, stereo and multichannel audio assets (up to eight channels).
- Tracker module assets behave the same way as any other audio assets.

Audio Listener

- Acts as a microphone-like device.
- Receives input from given Audio Source and plays sounds through the computer speakers.
- Each scene can only have one Audio Listener.
- Access the project-wide audio settings using the Audio Manager.
- View the Audio Clip Component page (Manual) for more information about Mono vs Stereo sounds.

Audio Components

Audio Source

- Audio Source plays back an Audio Clip in the scene.
- If Clip is 3D, source is played back at a given position and will attenuate over distance.
- Audio can be spread between speakers (stereo to 7.1) (Spread) and morphed between 3D and 2D (PanLevel). This can be controlled over distance with falloff curves.
- Individual filters can be applied to each audio source (need to program due to PRO feature only).

Audio Components

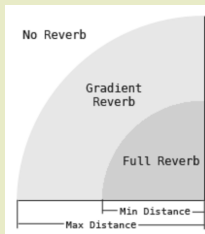
Create a new Audio Source

- 1 Import your audio files into your Unity Project. These are now Audio Clips.
- 2 Go to GameObject→Create Empty from the menubar.
- 3 With the new GameObject selected, select Component→Audio→Audio Source.
- 4 Assign the Audio Clip property of the Audio Source Component in the Inspector.

Audio Components

Reverb Zones

- Reverb Zones take an Audio Clip and distorts it depending where the audio listener is located inside the reverb zone.
- Gradually change from a point where there is no ambient effect to a place where there is one, for example when you are entering a cavern.



Audio Components

Microphone

- Useful for capturing input from a built-in (physical) microphone on your PC or mobile device.
- Start and end a recording from a built-in microphone, get a listing of available audio input devices (microphones), and find out the status of each such input device.
- No component for the Microphone class but you can access it from a script.

Coroutines

Why coroutines?

- Allows you to spread tasks across several frames.
- Method that can pause execution and return control; continues where it left off in the following frame.

When coroutines?

- In situations where you would like to use a method call to contain a procedural animation or a sequence of events over time.
- It is best to use coroutines for longer asynchronous operations, such as waiting for HTTP transfers, asset loads, or file I/O to complete.

Multithreading?

- Coroutines aren't threads. Synchronous operations that run within a coroutine still execute on the main thread.
- Multi-threaded code within Unity, consider the C# Job System