Unity Game Engine

Introduction to Unity – Audio

Audio

- In video games, audio is crucial, too.
- Most games play background **music** and have **sound effects**.

Audio : Sound Effects

- **Sound effects** are short clips that play along with actions in the game (such as a gunshot that plays when the player fires)

Audio : Music

- The sound clips for **music** are longer (often running into minutes) and playback isn't directly tied to events in the game.
- The sound files for music are usually much larger than the short clips used for sound effects.
- Music files are often the largest files in the game.

Audio File Formats

- A variety of audio formats supported by Unity:
  - WAV - Default audio format on Windows. Uncompressed sound file.
  - AIF - Default audio format on Mac. Uncompressed sound file.
  - MP3 - Compressed sound file;
  - OGG - Compressed sound file;
  - MOD - Music tracker file format. A specialized kind of efficient digital music.
  - XM - Music tracker file format. A specialized kind of efficient digital music.

Audio File Formats

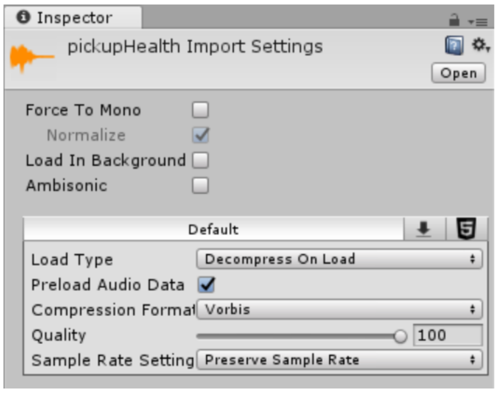
- General recommendations:
  - Choose uncompressed audio when the sound clip is short.
  - Longer sound clips (especially music) should use compressed audio.

How to get audio files?

- Create sound files using tools like Audacity (http://www.audacityteam.org/.)
- Download some sounds from one of the many free sound websites (e.g. www.freesound.org) and get the clips in WAV file format.
  - “Free” sounds are offered under a variety of licensing schemes.
  - Always make sure that you’re allowed to use the sound clip in the way you intend.
  - Many free sounds are for noncommercial use only.

Import an Audio Clip

- Drag audio files to the Project view.
- Check “Audio Clip Settings” in the Inspector



Inspector
pickupHealth Import Settings
Open

Force To Mono ☐
Normalize ✓
Load In Background ☐
Ambisonic ☐

Default

Load Type: Decompress On Load
Preload Audio Data ✓
Compression Format: Vorbis
Quality: 100
Sample Rate Setting: Preserve Sample Rate

Audio Clip Settings

- **Force To Mono -** If enabled, the audio clip will be down-mixed to a single channel sound.
- **Normalize** - When this option is enabled, audio will be normalized during the "Force To Mono" mixing down process.
- **Load In Background** – If enabled, the audio clip will be loading in the background without causing stalls on the main thread. This is **off by default**. Note that play requests on AudioClips that are still loading in the background will be deferred until the clip is done loading.
  - generally "on" for long music clips
  - "off" for short sound clips like sound effects
- **Ambisonic -** represents a soundfield that can be rotated based on the listener's orientation. It is useful for 360-degree videos. Enable this option if your audio file contains Ambisonic-encoded audio.

Audio Clip Settings

- **Load Type** - controls how the data from the file will be loaded by the computer at runtime.
  - **Decompress On Load -** Audio files will be decompressed as soon as they are loaded. <u>Use this option for smaller compressed sounds</u> to avoid the performance overhead of decompressing on the fly.
  - **Compressed In Memory -** Keep sounds compressed in memory and decompress while playing. <u>Only use it for bigger files</u> where decompression on load would use a prohibitive amount of memory.
  - **Streaming -** Decode sounds on the fly. Incrementally reads the data from the disk and decoded on the fly.

Audio Clip Settings

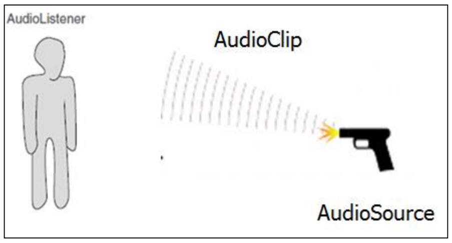
- **Preload Audio Data** – If enabled, the audio clip will be pre-loaded when the scene is loaded. This is **on by default** to reflect standard Unity behavior where all AudioClips have finished loading when the scene starts playing. It will consume memory while the sound waits to be used but.

Audio Clip Settings

- **Compression Format**
  - Music - Choose Vorbis compressed audio format.
  - Short sound effect clips don’t need to be compressed, so choose PCM (Pulse Code Modulation, the technical term for the raw, sampled sound wave).
- **Sample Rate Setting:**
  - Preserve Sample Rate (default)
  - Optimize Sample Rate
  - Override Sample Rate

Play Audio

- You must define **three different parts** in order to play sounds in Unity: **AudioClip**, **AudioSource**, and **AudioListener**.
  - **AudioClip** : Actual sound file. Should be imported.
  - **AudioSource**: the object that originates/plays audio clips
  - **AudioListener**: the object (the player) that hears sounds projected from audio sources.
    - An AudioListener component is already *on the default camera* when you create a new scene.

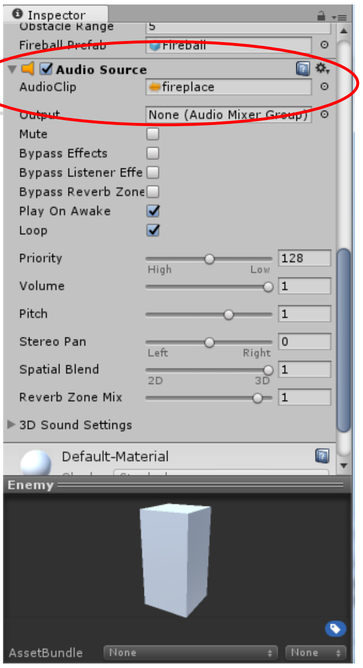


Play Sound Effect - Scenario 1

- Let's loop to play a sound automatically by putting a crackling fire sound on the enemy that wanders around.
  - Audio Clip: Crackling fire sound file
  - Audio Source: Enemy
  - Audio Listener: Camera

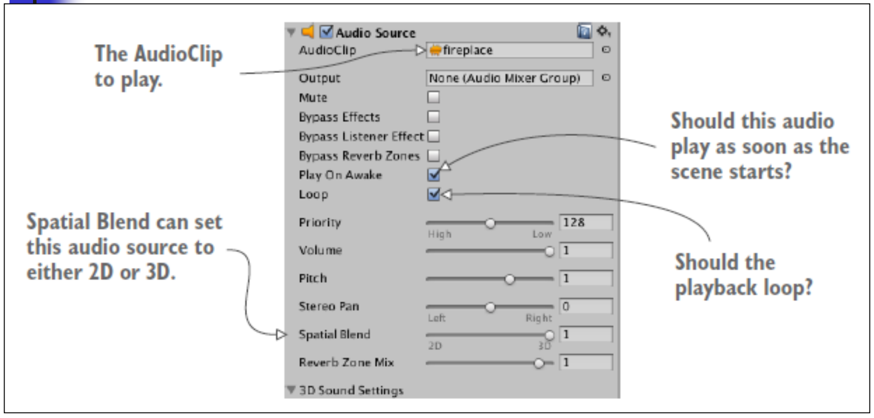
Play Sound Effect - Scenario 1

- Select the Enemy prefab and add a new **Audio Source** component (Navigation: Audio > Audio Source.)
  - Select Audio Source component in the Inspector.
  - Assign a sound clip to play. (Drag an audio file up to the **`AudioClip`** slot.)



Inspector
Obstacle Range 5
Fireball Prefab Fireball
Audio Source
AudioClip fireplace
Output None (Audio Mixer Group)
Mute ☐
Bypass Effects ☐
Bypass Listener Effe ☐
Bypass Reverb Zone ☐
Play On Awake ✓
Loop ✓
Priority 128 (High – Low)
Volume 1
Pitch 1
Stereo Pan 0 (Left – Right)
Spatial Blend 1 (2D – 3D)
Reverb Zone Mix 1
3D Sound Settings
Default-Material
Enemy
AssetBundle None None

Audio Source Setting



Audio Source
AudioClip fireplace
Output None (Audio Mixer Group)
Mute ☐
Bypass Effects ☐
Bypass Listener Effect ☐
Bypass Reverb Zones ☐
Play On Awake ✓
Loop ✓
Priority 128 (High – Low)
Volume 1
Pitch 1
Stereo Pan 0 (Left – Right)
Spatial Blend 1 (2D – 3D)
Reverb Zone Mix 1
3D Sound Settings

The AudioClip to play.

Should this audio play as soon as the scene starts?

Spatial Blend can set this audio source to either 2D or 3D.

Should the playback loop?

Spatial Blend (2D vs 3D Sounds): In 3D sounds, their volume and pitch are influenced by the movement of the listener. For example, a sound effect triggered in the distance will sound very faint.

Play Sound Effect - Scenario 2

- For the majority of sound effects you'll want to trigger the sound with code commands.
- Play sound when the player shoots and hits something.
  - Add an AudioSource component to the player object.
  - In this example, don't link in a specific audio clip. Specific audio clips will be defined in code.
  - Turn **off** "Play On Awake".
  - Modify your script (e.g `RayShooter.cs)` to play sounds when the bullet collides anything.

RayShooter.cs

```
public class RayShooter : MonoBehaviour {
  [SerializeField] private AudioSource soundSource;  //player object
  [SerializeField] private AudioClip hitWallSound;   //audio clip
  [SerializeField] private AudioClip hitEnemySound;  //audio clip

  void Update() {
    ...
    if (target != null) {
      target.ReactToHit();
      soundSource.PlayOneShot(hitEnemySound);
      // soundSource.clip = hitEnemySound;
      // soundSouce.Play();
    }
    else {
      StartCoroutine(SphereIndicator(hit.point));
      soundSource.PlayOneShot(hitWallSound);
    }
    ...
```

AudioSource Class

- See https://docs.unity3d.com/ScriptReference/AudioSource.html
- Variable:
  - `clip` : the default AudioClip to play.
  - `loop` : boolean for looping
  - `volume` : the volume of the audio source (0.0 to 1.0).
- Methods:
  - `Pause()`
  - `Play()`
  - `PlayOneShot (audioclip)`
  - `Stop()`

Background Music

- Music tracks tend to consume a large amount of memory on the computer. **Optimize** two things:
  - **When to load** - Avoid to have the music loaded into memory before it's needed
  - **How to load and store** - Avoid to consume too much memory when loaded. Streaming vs Loading at once.

Background Music – Lazy Load

- Optimizing **when** music loads is done using the `Resources.Load()` command.
  - This command allows you to load assets by name.
  - More importantly, assets from Resources aren't loaded until the code manually fetches them.
- We want to lazy-load the audio clips for music. Otherwise, the music could consume a lot of memory while it isn't even being used.

Background Music - Streaming

- Streaming music off the disc - saves the computer from ever needing to have the entire file loaded at once. The style of loading was a setting in the Inspector of the imported audio clip.

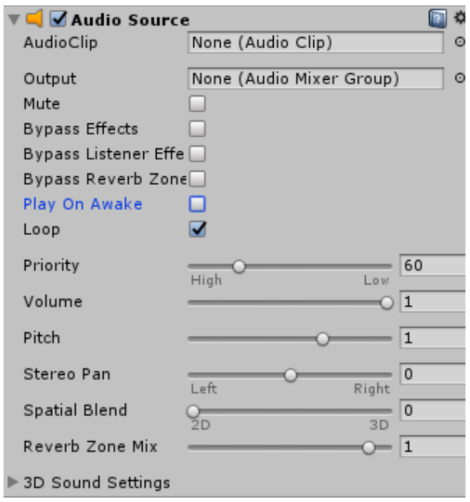
Background Music - Import

- IMPORT AUDIO CLIPS
  - Load Type - Choose Streaming
  - Set the Compression Format to Vorbis, for compressed audio.
  - Load In Background - Turn on so that the game won’t pause or slow down while music is loading.

** Remember to place the music file in the Resources folder.*

Background Music - AudioSource

- SET UP AN AUDIOSOURCE
  - Create an empty `GameObject`, name it “BG_Music”.
  - Add an `AudioSource` component to “BG_Music” and then adjust the settings in the component.



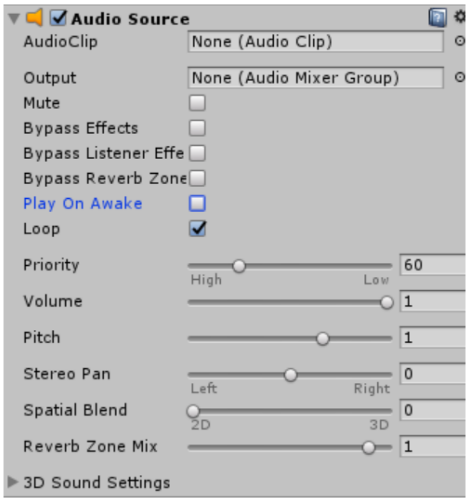
Audio Source [✓]

AudioClip	None (Audio Clip)
Output	None (Audio Mixer Group)
Mute	☐
Bypass Effects	☐
Bypass Listener Effe	☐
Bypass Reverb Zone	☐
Play On Awake	☐
Loop	✓
Priority (High – Low)	60
Volume	1
Pitch	1
Stereo Pan (Left – Right)	0
Spatial Blend (2D – 3D)	0
Reverb Zone Mix	1

3D Sound Settings

Background Music - AudioSource

- Deselect “Play On Awake” as we want to control it.
- Turn on the Loop option this time to let the background music play over and over in a loop.
- Spatial Blend – set it to 2D because background music doesn’t have any specific position in the scene.



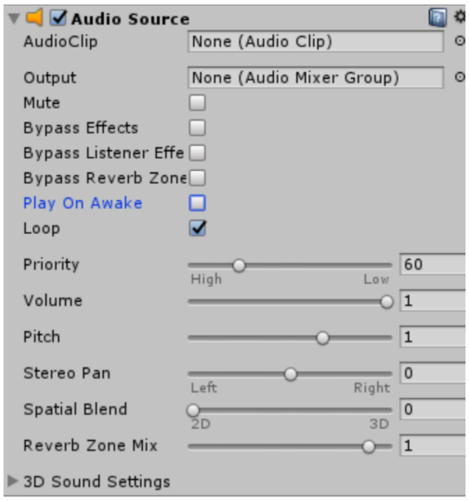
Audio Source

AudioClip	None (Audio Clip)
Output	None (Audio Mixer Group)
Mute	☐
Bypass Effects	☐
Bypass Listener Effe	☐
Bypass Reverb Zone	☐
Play On Awake	☐
Loop	✓
Priority (High – Low)	60
Volume	1
Pitch	1
Stereo Pan (Left – Right)	0
Spatial Blend (2D – 3D)	0
Reverb Zone Mix	1

3D Sound Settings

Background Music - AudioSource

- Reduce the Priority value to 60 (lower values are higher priority.)
  - When too many sounds are playing simultaneously, the audio system will start discarding sounds based on priorities.



Audio Source

AudioClip	None (Audio Clip)
Output	None (Audio Mixer Group)
Mute	☐
Bypass Effects	☐
Bypass Listener Effe	☐
Bypass Reverb Zone	☐
Play On Awake	☐
Loop	✓
Priority (High – Low)	60
Volume	1
Pitch	1
Stereo Pan (Left – Right)	0
Spatial Blend (2D – 3D)	0
Reverb Zone Mix	1

3D Sound Settings

```
...
public class BG_MusicControl : MonoBehaviour {
    [SerializeField] private AudioSource bgMusic;
...
    void Start () {
        bgMusic.ignoreListenerVolume = true;
        bgMusic.ignoreListenerPause = true;
        bgMusic.clip =
                Resources.Load("intro-synth") as AudioClip;
        bgMusic.volume = 1;
    }
...

    private void PlayMusic() {
        bgMusic.Play();
    }
    private void StopMusic() {
        bgMusic.Stop();
    }
...
}
```

Resources

- Audio files
  - SoundFX.zip (sound effect files)
  - Music.zip (background music files)