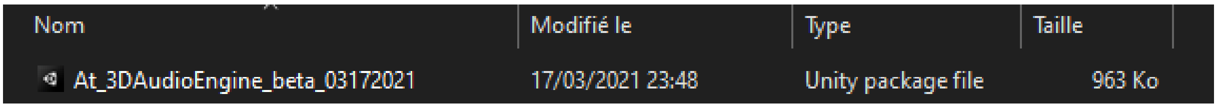
At_3DAudioEngine - Unity

Use case – version.1.0 Beta – 03/26/2021

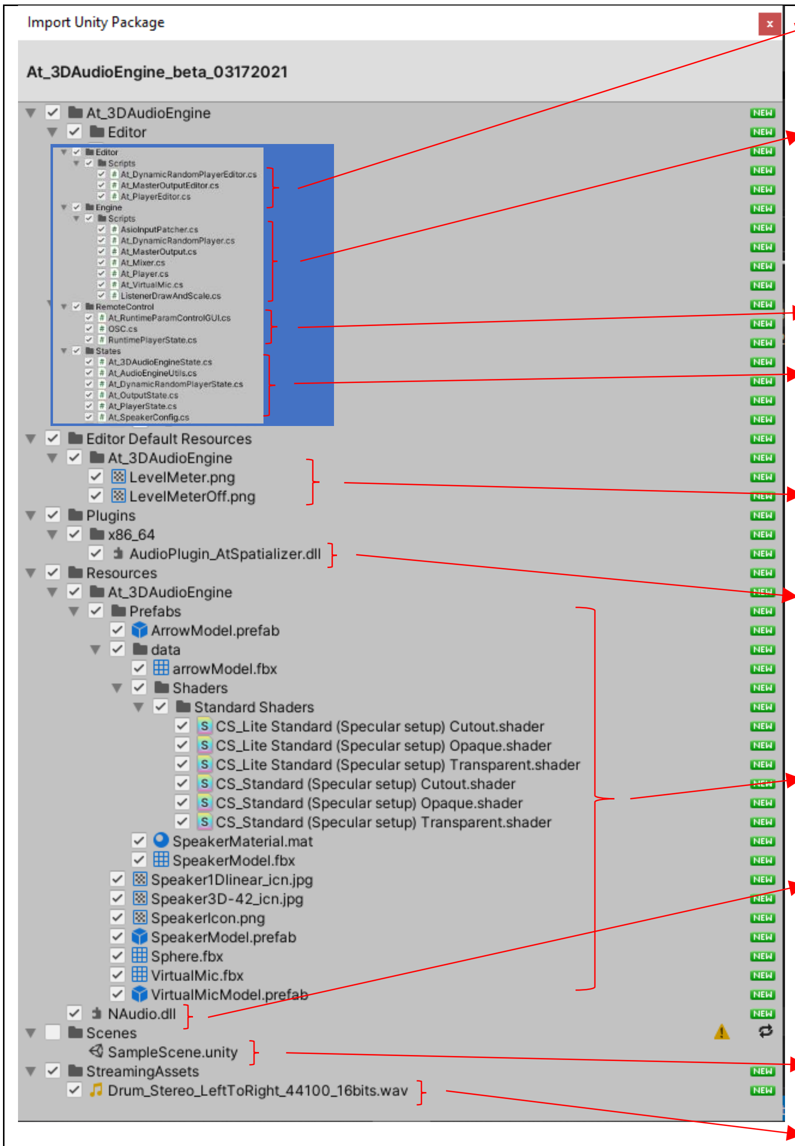
1 – Create an empty Unity 2020.X project

2 - Double-click on the provided file “At_3DAudioEngine_beta_MMDDYYYY.unitypackage” to import the audio engine



Nom	Modifié le	Type	Taille
At_3DAudioEngine_beta_03172021	17/03/2021 23:48	Unity package file	963 Ko

3 – the following window will show you all the imported files :



Import Unity Package

At_3DAudioEngine_beta_03172021

```
☑ At_3DAudioEngine
  ☑ Editor
    ☑ Editor
      ☑ Scripts
        ☑ At_DynamicRandomPlayerEditor.cs
        ☑ At_MasterOutputEditor.cs
        ☑ At_PlayerEditor.cs
    ☑ Engine
      ☑ Scripts
        ☑ AsioInputPatcher.cs
        ☑ At_DynamicRandomPlayer.cs
        ☑ At_MasterOutput.cs
        ☑ At_Mixer.cs
        ☑ At_Player.cs
        ☑ At_VirtualMic.cs
        ☑ ListenerDrawAndScale.cs
    ☑ RemoteControl
      ☑ At_RuntimeParamControlGUI.cs
      ☑ OSC.cs
      ☑ RuntimePlayerState.cs
    ☑ States
      ☑ At_3DAudioEngineState.cs
      ☑ At_AudioEngineUtils.cs
      ☑ At_DynamicRandomPlayerState.cs
      ☑ At_OutputState.cs
      ☑ At_PlayerState.cs
      ☑ At_SpeakerConfig.cs
☑ Editor Default Resources
  ☑ At_3DAudioEngine
    ☑ LevelMeter.png
    ☑ LevelMeterOff.png
☑ Plugins
  ☑ x86_64
    ☑ AudioPlugin_AtSpatializer.dll
☑ Resources
  ☑ At_3DAudioEngine
    ☑ Prefabs
      ☑ ArrowModel.prefab
      ☑ data
        ☑ arrowModel.fbx
        ☑ Shaders
          ☑ Standard Shaders
            ☑ CS_Lite Standard (Specular setup) Cutout.shader
            ☑ CS_Lite Standard (Specular setup) Opaque.shader
            ☑ CS_Lite Standard (Specular setup) Transparent.shader
            ☑ CS_Standard (Specular setup) Cutout.shader
            ☑ CS_Standard (Specular setup) Opaque.shader
            ☑ CS_Standard (Specular setup) Transparent.shader
        ☑ SpeakerMaterial.mat
        ☑ SpeakerModel.fbx
      ☑ Speaker1Dlinear_icn.jpg
      ☑ Speaker3D-42_icn.jpg
      ☑ SpeakerIcon.png
      ☑ SpeakerModel.prefab
      ☑ Sphere.fbx
      ☑ VirtualMic.fbx
      ☑ VirtualMicModel.prefab
  ☑ NAudio.dll
□ Scenes
    SampleScene.unity
☑ StreamingAssets
  ☑ Drum_Stereo_LeftToRight_44100_16bits.wav
```

- Scripts for the GUI of the 3D Audio Engine

- Scripts for the core of the 3D Audio Engine in Unity (to be used with Game Object in your scene)

- Scrips for OSC remote control

- Scrips for state load/save (json files are saved here)

- Textures used for level metering in the editor

- C/C++ dynamic library used for spatialization of 3D Sources in the 3D Audio Engine (Source Code not provided)

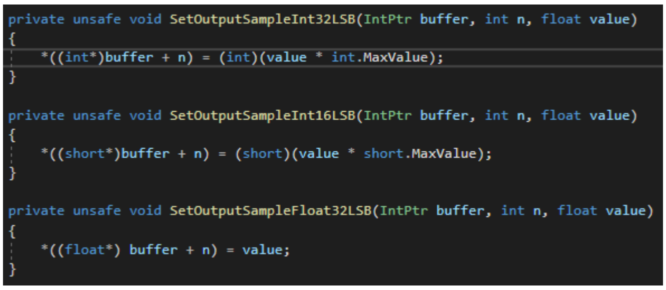
- Prefab data for displayed virtual speakers in the scene

- NAudio dynamic library built for windows x86_64, used for multichannel audio player and ASIO driver output

- Example scene

- Some audio files
NB : for now, the “StreamingAssets” folder is where you must copy all the audio files

4 – The C# part of Audio Engine uses “Unsafe code”, for the “Sample Provider” used to manage the various audio sample formats required by the output device. See in “AsioInputPatcher.cs” :

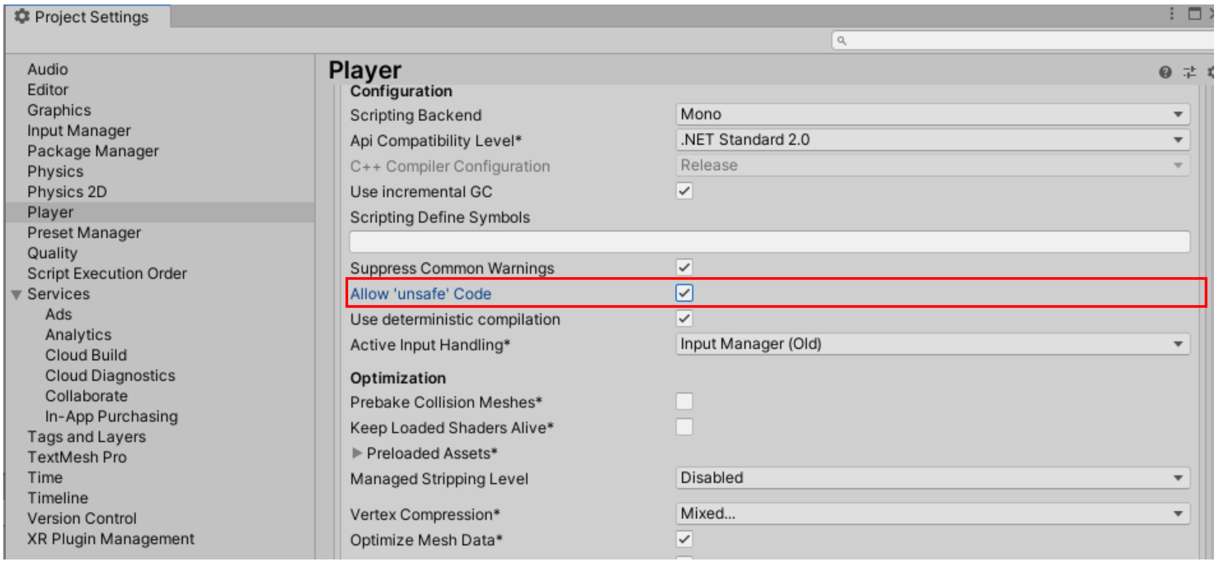


```
private unsafe void SetOutputSampleInt32LSB(IntPtr buffer, int n, float value)
{
    *((int*)buffer + n) = (int)(value * int.MaxValue);
}

private unsafe void SetOutputSampleInt16LSB(IntPtr buffer, int n, float value)
{
    *((short*)buffer + n) = (short)(value * short.MaxValue);
}

private unsafe void SetOutputSampleFloat32LSB(IntPtr buffer, int n, float value)
{
    *((float*) buffer + n) = value;
}
```

So in Edit>Project settings>Player>Other Settings>… click the option “Allow ‘unsafe’ Code :



Player

Configuration

Scripting Backend: Mono

Api Compatibility Level*: .NET Standard 2.0

C++ Compiler Configuration: Release

Use incremental GC ☑

Scripting Define Symbols

Suppress Common Warnings ☑

Allow 'unsafe' Code ☑

Use deterministic compilation ☑

Active Input Handling*: Input Manager (Old)

Optimization

Prebake Collision Meshes* □

Keep Loaded Shaders Alive* □

Preloaded Assets*

Managed Stripping Level: Disabled

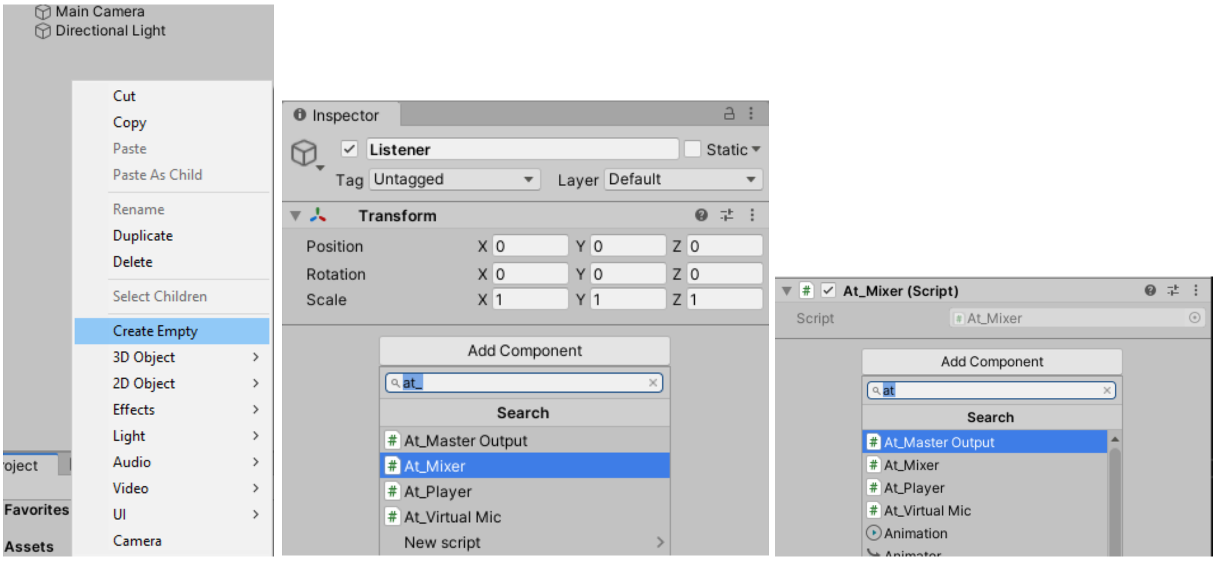
Vertex Compression*: Mixed...

Optimize Mesh Data* ☑

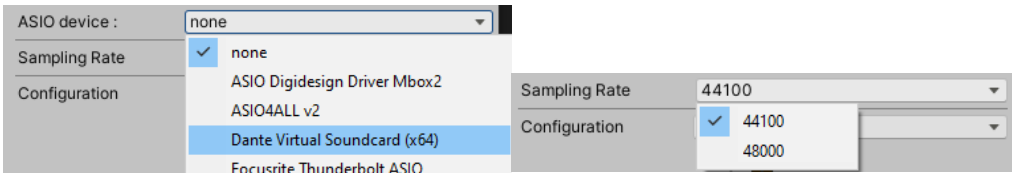
5 – To use the engine you need to create two *GameObjects* in the *Hierarchy*, a *Listener* and a *Source* :

Step 1 : create an empty GameObject in your scene, called “Listener” (or whatever…) :

Step 2 : Add the Scripts “At_Mixer.cs” and “At_MasterOutput.cs” to this object



Step 3 : using the GUI of the “At_MasterOutput.cs” script, select the ASIO device and the sampling rate (depending on the audio format of the audio files) :



ASIO device : none

Sampling Rate

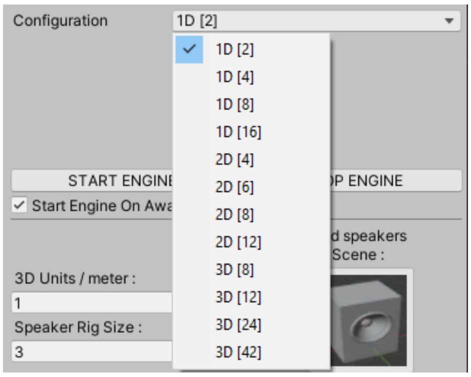
Configuration

none
ASIO Digidesign Driver Mbox2
ASIO4ALL v2
Dante Virtual Soundcard (x64)
Focusrite Thunderbolt ASIO

Sampling Rate 44100

Configuration

Step 4 : using the GUI of the “At_MasterOutput.cs” script, select you basic speakers configuration among the twelves proposed :

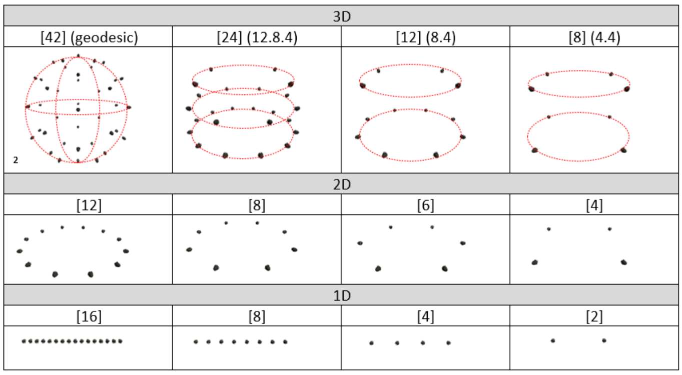


Configuration 1D [2]

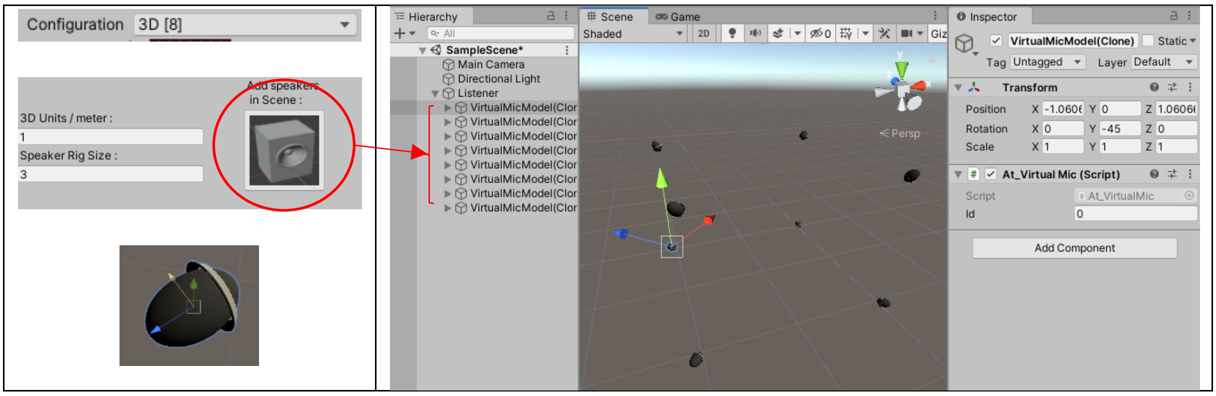
☑ Start Engine On Awa

3D Units / meter : 1

Speaker Rig Size : 3



Step 5 : click the “Add speakers in Scene” to add the virtual microphones” used to spatialized 3D source :



Configuration 3D [8]

3D Units / meter : 1

Speaker Rig Size : 3

VirtualMicModel(Clone) Static

Tag Untagged Layer Default

Transform

Position X -1.0606 Y 0 Z 1.0606

Rotation X 0 Y -45 Z 0

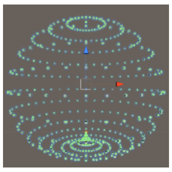
Scale X 1 Y 1 Z 1

☑ At_Virtual Mic (Script)

Script At_VirtualMic

Id 0

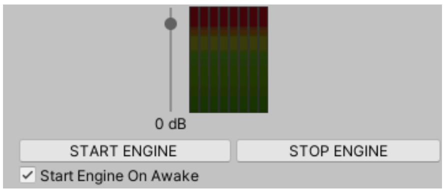
In this example, this will add 8 *GameObjets* with a mesh representing a virtual cardioid microphone pointing toward the opposite direction from the *Listener* position. As illustrated bellow, a cardioid directivity means a sensitivity equal to 1 in the forward direction and a sensitivity of 0 in the backward direction :



Each of these *GameObjects* is a child of the *Listener* GameObject and has the script “At_VirtualMic.cs” attached to it with an unique ID, given the physical output channel of this microphone in the ASIO device. You can change the position and rotation properties of these

GameObjects ("manually" in the hierarchy) to fulfill your needs, depending on your physical speakers configuration.

A last, you can check the option "Start Engine On Awake" to initialize ASIO output when starting the game simulation. This can be done at runtime in the editor buy clicking the button "START ENGINE" or buy calling the public method "`void StartEngine()`" of the "`At_MasterOutput`" `class attached to the Listener GameObject.`



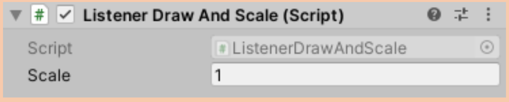
☑ Start Engine On Awake

The engine must be started before playing an audio source (see below).

NB : a multichannel meter displayed the short-term RMS value of an output buffer in dBFs for each output channel. The master level can be adjusted with a slider, from + 10dBFs to -80dBFs.

NEW !

You add the `ListenerDrawAndScale.cs` to adjust the scale of the listener

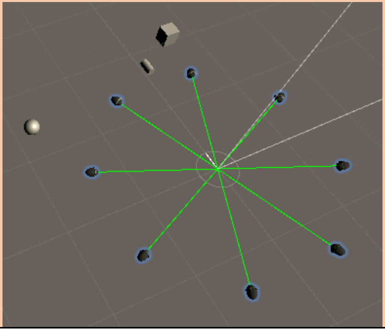


☑ Listener Draw And Scale (Script)

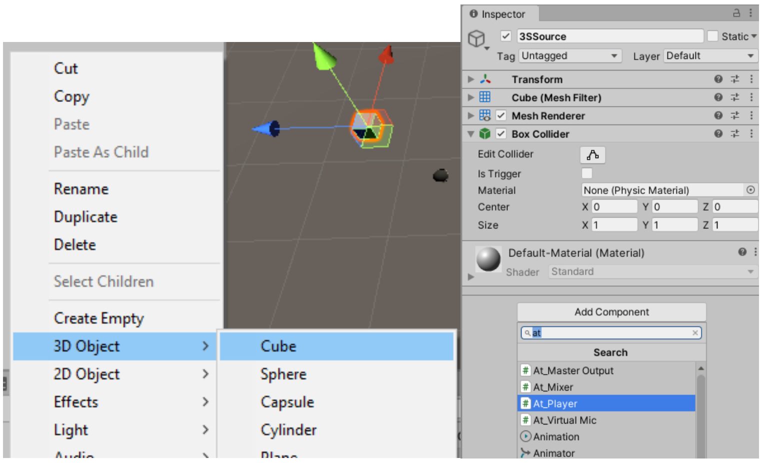
Script ListenerDrawAndScale

Scale 1

This script also display green lines (only on the "Scene" window when editing) to see the limit of the Virtual Microphone Rig of the listener



Step 6.a (Use Case #1) : adding 3D Audio Source :



☑ 3SSource □ Static

Tag Untagged Layer Default

Transform

Cube (Mesh Filter)

☑ Mesh Renderer

☑ Box Collider

Edit Collider

Is Trigger □

Material None (Physic Material)

Center X 0 Y 0 Z 0

Size X 1 Y 1 Z 1

Default-Material (Material)

Shader Standard

at

At_Master Output

At_Mixer

At_Player

At_Virtual Mic

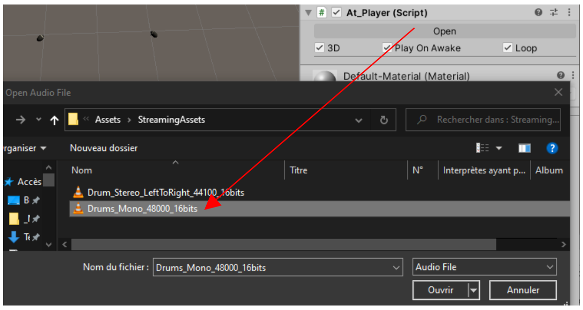
Animation

Animator

The audio file is supposed to be monophonic. Otherwise, only the first channel of the audio file feeds the audio graph.

Add a 3D Object in the hierarchy, called “3DSource”, and add the script “At_Player.cs” :

Click the “Open” button and select an audio file in the “streamingAssets” folder :

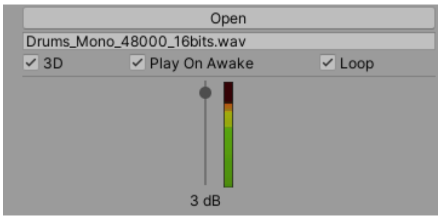


☑ 3D ☑ Play On Awake ☑ Loop

Drum_Stereo_LeftToRight_44100_16bits
Drums_Mono_48000_16bits

Nom du fichier : Drums_Mono_48000_16bits
Audio File

Because this audio source is supposed to be spatialized by the 3D Audio Engine, check the “3D” option. You can also check the option “Play On Awake” if you want the audio file to be played automatically when starting the game simulation. The audio source must be play/stop at runtime by calling the public method “`void StartPlaying()`”/”`void StopPlaying()`” of the “`At_Player`” class attached to the GameObject. Check the “Loop” option if you want the audio file to be looped.



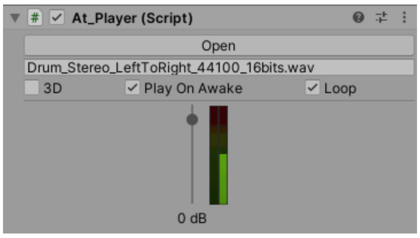
Drums_Mono_48000_16bits.wav
☑ 3D ☑ Play On Awake ☑ Loop

NB : As for the master output, a multichannel meter displayed the short-term RMS value in dBFs for each channel of the audio file (only one channel in the previous example). The level of this source can be adjusted with a slider, from + 10dBFs to -80dBFs. Remember : because this is an 3D audio source, only the first channel is feeding the audio graph.

Step 6.b (Use Case #2) : adding 2D Audio Source :

The audio file is supposed to be polyphonic. Channel #N is directly feeding the Nth output of ASIO device. If the audio file has more channels than the output devices, these channels are ignored.

Add a 3D Object in the hierarchy, called “2DSource” and, as for a 3D audio source, add the script “At_Player.cs”. However, you must let the option “3D” uncheck.

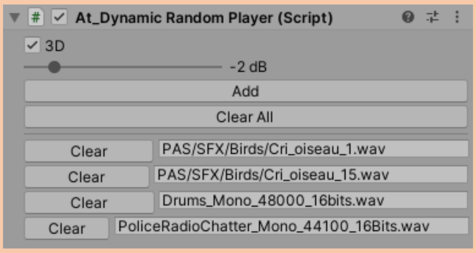


Drum_Stereo_LeftToRight_44100_16bits.wav
☐ 3D ☑ Play On Awake ☑ Loop

Step 6.c (Use Case #3) : adding a source instance (2D/3D) at runtime, with a random playlist of audio file :

NEW !

Create a 3D Object in the hierarchy called “DynamicSource” and Add the script `At_DynamicRandomPlayer.cs` to it.



At_Dynamic Random Player (Script)

☑ 3D

-2 dB

Add

Clear All

Clear	PAS/SFX/Birds/Cri_oiseau_1.wav
Clear	PAS/SFX/Birds/Cri_oiseau_15.wav
Clear	Drums_Mono_48000_16bits.wav
Clear	PoliceRadioChatter_Mono_44100_16Bits.wav

Use the “Add” button to add an audio file in the list. Use the “Clear” button beside the file name to remove this file from the list

Use the “Clear All” button to remove all the audio file of the list

Check “3D” if you want a 3D sound to be generated or let uncheck in 2D case

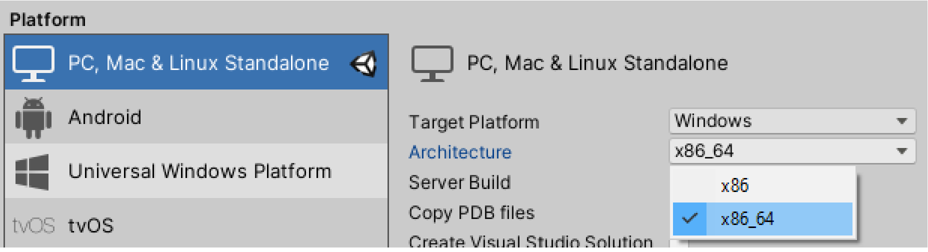
Set the gain of the

You can now call the public method “`void AddOneShotInstanceAndRandomPlay()” of the script At_DynamicRandomPlayer.cs` to trigger the creation of a new GameObject with the corresponding At_Player.cs script on it.

/!\ For now any of this GameObject are destroyed when playing is ending. However, the player do not loop, so “almost” no CPU are consumed.

Step 7 BUILD

Select the Windows target platform (no Mac OS version available) and x86_64 architecture (no x86 version available).



Platform

PC, Mac & Linux Standalone

Android

Universal Windows Platform

tvOS

PC, Mac & Linux Standalone

Target Platform: Windows

Architecture: x86_64

Server Build

Copy PDB files

Create Visual Studio Solution

And build…