UMX AUDIO TOOLS DOCUMENTATION

By Frederik Max

UMX 3DSound

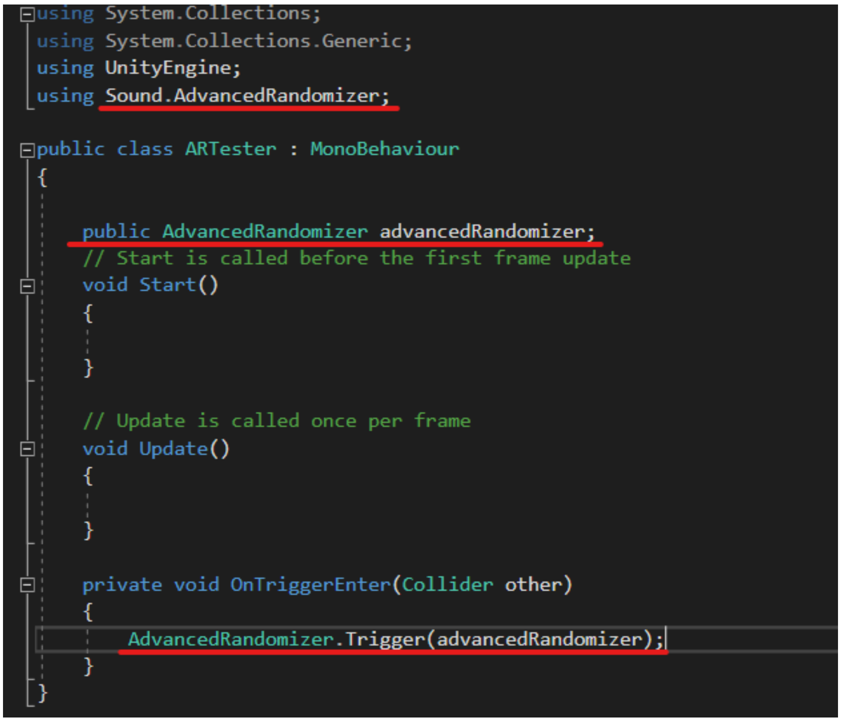
- Drag and drop this prefab into the world, wherever you want a 3D sound to play from. The volume rolloff is already set to a natural sounding curve and you can adjust the "Max Distance" to control how far away you can hear the sound.

  This tool is designed to handle looping 3D sounds like rivers, generators, engines and fire.

UMX AdvancedRandomizer

- This prefab handles audio randomization in layers (up to three). Place the prefab within the hierarchy and drag the sounds you want to use to each layer. You have the option to prevent retriggering (playing the same sound twice in a row) by enabling retrigger prevention on each layer. You also have the option to randomize both pitch and volume.

  You can call the AdvancedRandomizer gameObject from a script using these three simple lines: (Remember to assign it in the inspector as well).



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Sound.AdvancedRandomizer;

public class ARTester : MonoBehaviour
{

    public AdvancedRandomizer advancedRandomizer;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter(Collider other)
    {
        AdvancedRandomizer.Trigger(advancedRandomizer);
    }
}
```

This tool is designed to handle sounds with multiple layers like gunshots, footsteps, spells and impacts.

UMX AmbientZone

- Drag the prefab into your scene and re-size the box trigger to fit the area where you want your ambient sound to play. You have the option to control fade-in and fade-out time on your sound, as well as the volume targets for the fades. Whenever your character enters the trigger the volume of the sound will fade up and whenever the characters exits, it will fade down.

  You can replace the box trigger with another shape if needed.
  This tool is designed to handle ambiences and potentially music. Also exists as a 2D tool.

UMX AudioGate

- You can drag this tool into your scene if want to transition from one mixer snapshot to another. Under the parent gameObject you will see both the "GateOne" and "GateTwo" object. On each of these you can assign the snapshot you want to transition to whenever you enter the trigger, as well as the transition time to that snapshot.
  This tool is designed to handle vertical remixing and horizontal remixing of both ambience and music.

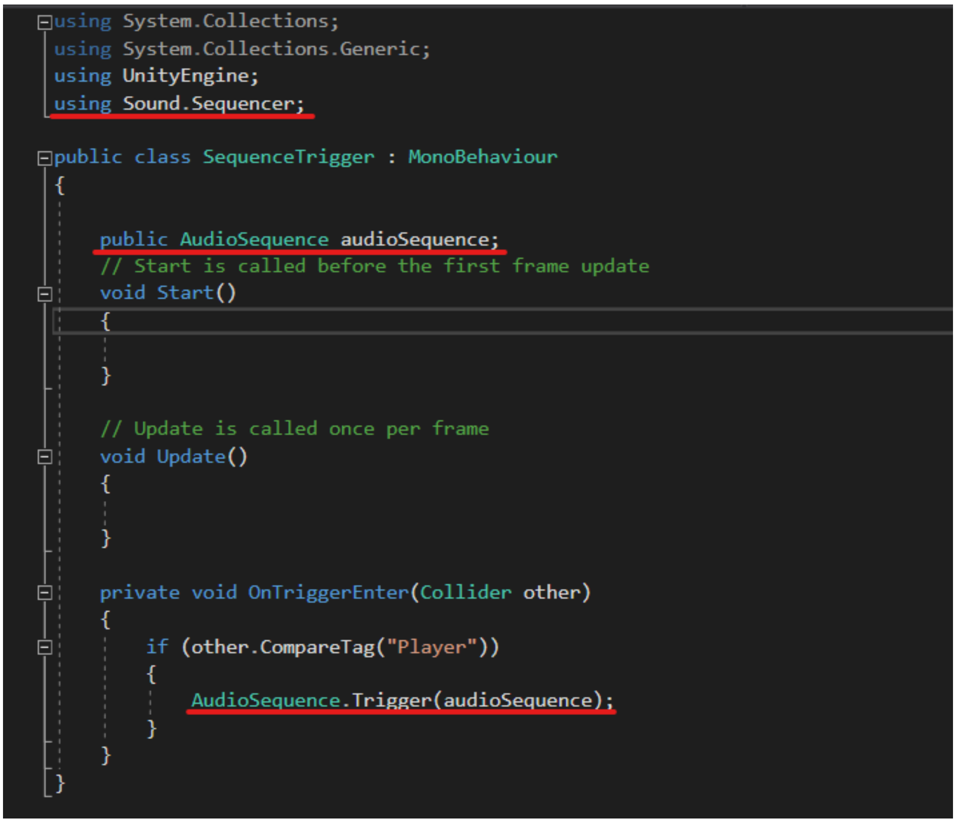
UMX AudioRequestSystem

- Place this prefab within your hierarchy to enable the other developers to quickly make audio request. This tool generates a text file within the asset folder which updates whenever someone adds a request. To add a request just type your request into the text box and click the "Add Request" button. The console will inform you that a request has been successfully added by writing "Request Added: ".
  This tool is designed to make your teams audio workflow quicker, quickly generating tasks and easily through ideas forward to the sound designer in a hectic environment.

UMX AudioSequence

- AudioSequence works just like the AdvancedRandomizer but instead of randomizing between the layers at the same time, you can set a delay between each layer. The sequence plays element 1 first, then element 2 and then element 3. Each set of sounds can have volume and pitch randomization.

  You can call the AudioSequence gameObject from a script using these three simple lines:



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Sound.Sequencer;

public class SequenceTrigger : MonoBehaviour
{

    public AudioSequence audioSequence;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            AudioSequence.Trigger(audioSequence);
        }
    }
}
```

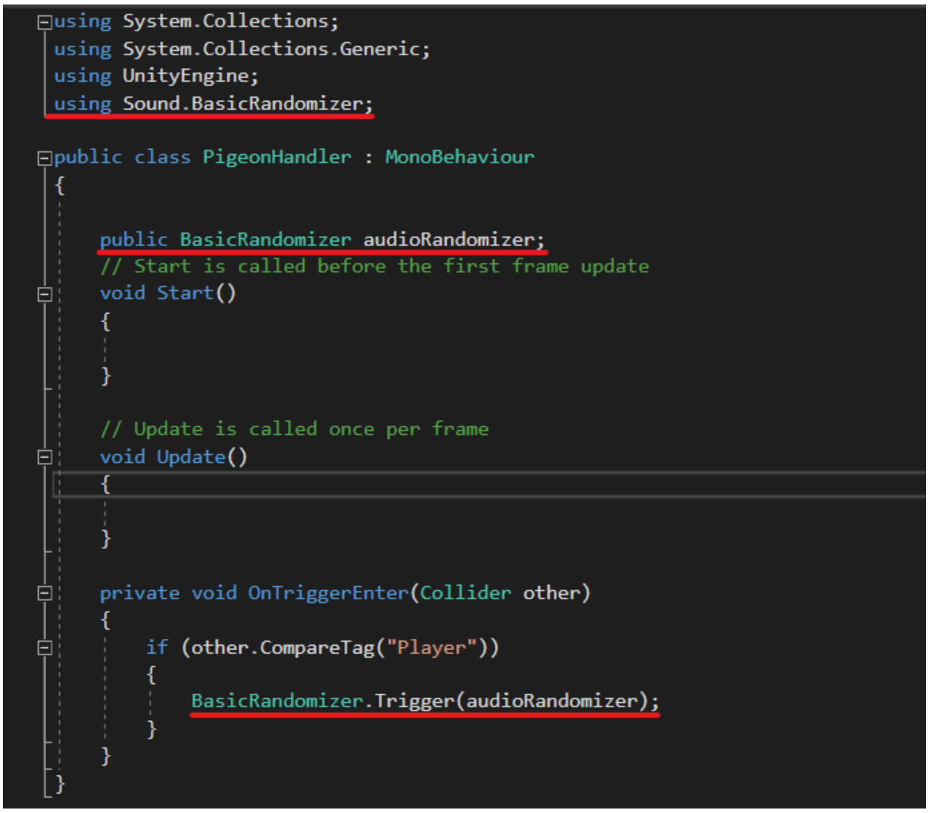
This tool is designed to handle sounds that need to be played in a sequence like impacts and musical sounds.

UMX BackgroundFade

- Drag this prefab into your scene if you want a sound to slowly fade up to a given volume target. This is useful for when wanting to avoid an abrupt start to some ambience or a piece of music. Use “Fade Time” to determine how long you want the fade to be. Use “Fade Target” to set the target volume.
  This tool is designed to handle fading in loops like ambiences, music and alarms.

UMX BasicRandomizer

- BasicRandomizer is a version of the AdvancedRandomizer that only contains a single layer of sounds to randomize between. You can still randomize volume and pitch, while also being able to toggle retriggering.
  You can call the BasicRandomizer gameObject from a script using these three simple lines:



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Sound.BasicRandomizer;

public class PigeonHandler : MonoBehaviour
{

    public BasicRandomizer audioRandomizer;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            BasicRandomizer.Trigger(audioRandomizer);
        }
    }
}
```

This tool is designed to handle randomization between sounds that don’t necessarily need layers like abilities, footsteps and gunshots.

UMX BasicSpline

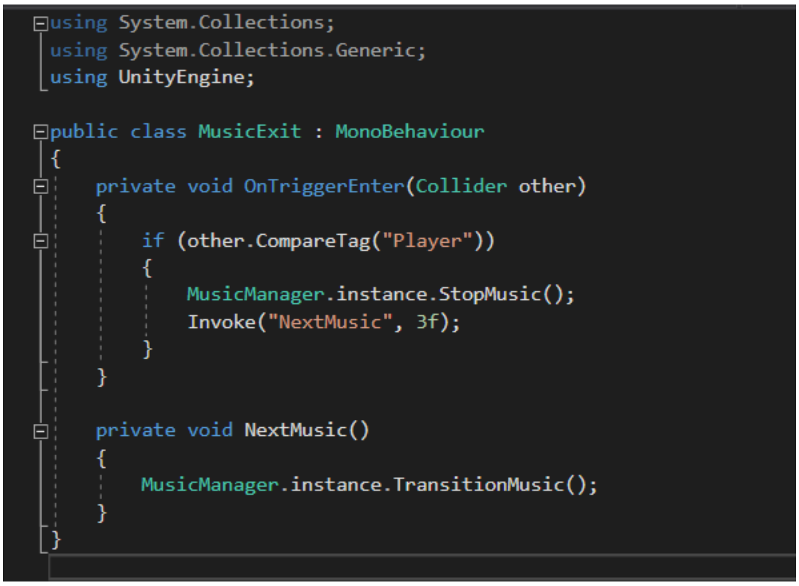
- Drag this prefab into your scene if want a sound to follow a specific path. When dragged into the scene you are presented with the gameObject "UMX BasicSpline", which has two children. "UMX SplineAudioSource" holds the audioSource that will follow along the path. You need to assign what you want the audioSource to follow (Character Follow Object) and which path the audioSource should move along (Spline). The "UMX SplineHolder" object is where the spline points should be located, it is between these points that the spline is drawn. You can make your own points, delete the ones I have provided and move them around. Remember to assign the "UMX SplineHolder" object in the "UMX SpineAudioSource" object.
  This tool is designed to handle sounds that you want to follow a path like rivers, traffic and various other 3D sounds.

UMX MultiLoop3D

- MultiLoop3D works just like "3D Sound" but has three different 3D sounds assigned to it, with different rolloffs. This way you can simulate more detail when moving closer to the sound. If you want, you can also make your own rolloffs to have different sounds playing depending on how close you are instead of them being stacked.
  This tool is useful for giving 3D loops more detail like car engines and machines.

UMX MusicManager

- MusicManager requires a lot of scripting on your own part. The basic setup is there for you to follow and expand on depending on your music needs. You can call for a piece of music to start (and stopping another) from the manager like this;



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MusicExit : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            MusicManager.instance.StopMusic();
            Invoke("NextMusic", 3f);
        }
    }

    private void NextMusic()
    {
        MusicManager.instance.TransitionMusic();
    }
}
```

UMX MusicTriggerGate

- MusicTriggerGate works almost the same way as “AudioGate” but also has the ability to play stingers when moving from one snapshot to another. On the “GateOne” object you can see the “Music Gate Stinger Handler” script, where you can assign two audio clips. The enter stinger will play when entering an area (depending on which way your gate is turned), and the exit stinger will play when exiting.
  This tool is designed to handle smooth transitions between different pieces of music with a stinger.

UMX PlaySoundAtLocation

- If you want to enter a trigger and have a certain sound play to the right, this is the tool to use. As a child to the PlaySoundAtLocation gameObject, you have the object "Audio Location". This is the object you need to move around to have a sound play from that location. This object also contains a basic randomizer to randomize between sounds.
  This tool is designed to handle positioned audio when entering a trigger. Useful for horror games and the like.

UMX Random3D

- Drag and drop this prefab into your scene wherever you want random 3D sounds to play from on a loop. You can again randomize between volume and pitch, and also toggle retriggering.
  This tool is useful for when you want to have a more organic landscape with sounds seemingly happening randomly around you.

UMX ReverbZone

- If you need a reverb zone that is not based on a circular emitter, this is a solution for that. Place and resize the box sized trigger however you want. Whenever you enter the trigger it will activate a reverb zone in that trigger. You can resize both the trigger and the reverb zone. The "Reverb Off Delay" time determines how long after you exit the trigger it will take for the reverb zone to deactivate. This ensures that you won't hear any clicks when the reverb is suddenly turned off.
  This tool is designed to handle reverb zones for rooms that are not circle shaped.

UMX ThirdPersonFootsteps

- Drag and drop this into your hierarchy if you have a third person character that needs footsteps. This prefab has the ability to change footstep sounds depending on which surface you are walking on. The sounds change depending on how the collider you are walking on is tagged. If you want to add more footstep surfaces or delete some, you can just copy-paste what is written in the script and rename appropriately.
  This tool is designed to handle footsteps for third person characters.

UMX TriggerEnterExitAudio

- Drag and drop this prefab into your scene if you want to play a sound when you enter a trigger and play another sound when exiting it. You have the option to randomize sounds, volume and pitch for both the entering and exiting events. You can scale the box trigger to your liking, to cover rooms and hallways. The box collider can also be replaced by another shaped collider if needed.
  This tool is designed to handle sounds when entering or exiting an area like sounds that inform players of what is to come and musical stingers. This tool also exists in a 2D format.

UMX VelocityPitch

- This tool handles pitch change based on velocity. The prefab already has a sphere loaded for testing, but the script can be transferred to other gameObject. If you load the script on another gameObject, remember to add an audioSource component as well. “Smooth Time” determines how smooth you want the transition from one pitch to another. “Multiplier” determines how fast this transition happens and “Use Smoothening” determines whether “Smooth Time” should be applied at all. Use smoothening when handling object like cars that don’t suddenly drop in pitch. On the other hand, don’t use smoothing when handling objects like glass bottles rolling on the floor, since their pitch drops immediately when stopping.

This tool is designed to handle and simulate sounds from rolling/moving objects like cars, rolling bottles and balls.

UMX Volumetric3D

- Just like the standard "3DSound" tool, Volumetric3D handles 3D sounds that you want to surround the player when standing close enough to them. In other words, the sounds transform into 2D sounds. Just drag and drop this into your scene and adjust the "Max Distance" if needed. Depending on your needs, you can also change the "Spatial Blend" curve to better suit transition from 3D to 2D that you are going for.
  This tool is designed to handle 3D sounds that need to smoothly blend into 2D like ambiences, sounds coming from a doorway, water and machines.

UMX FlipFlop

- Drag FlipFlop into your scene if you want to alternate between two different layers of randomized sounds every time you walk through a trigger. You again have the option to randomize pitch, volume and toggle retriggering.
  This tool is designed to handle alternating sounds like wood creaks.