

SoundInstance Bases

!

This describes some functions and introduces you to SoundInstance, to see exemples of all functions and possibilities, please go on [Functions](#).

Introduction

This is not mandatory
Firstly, you may want to add some musics and sounds to the game library. If you didn't do at the setup, just open "Tools > SoundInstance > Game Musics Window" or "Tools > SoundInstance > Sounds Library". Sounds and musics must have a unique name (a music can have the same name as a sound since they are not stored in the same place).

Playing a sound

You can play any sound from any scripts, even statics one if you use them. To play a sound, there are 2 methodes you can use depending on your needs. You can either use `SoundInstance.InstantiateOnTransform()` or `SoundInstance.InstantiateOnPos()`.

Example:

csharp

```
public void Fire()
{
    AudioClip Clip_Fire;

    //SoundInstance.InstantiateOnTransform(AudioClip, Transorm, [Volume = -1], [Is3D = true], [Randomization = SoundInstance.Ran
    SoundInstance.InstantiateOnTransform(Clip_Fire, transform);
}
```

This will play the 'Clip_Fire' clip on the transform this script is, with the default volume, in a 3D environment and with no randomization. This is the really basic way to use this asset. But most of the time, you will need to adjust some settings.

Supposing you are using the same fire clip each times you fire with a gun, it will sounds repeated and unrealistic. However, you can add some randomization to the sound so it sounds a little different all the times and it's already much more realistic. Lets do this then!

csharp

```
public void Fire()
{
    AudioClip Clip_Fire;

    //SoundInstance.InstantiateOnTransform(AudioClip, Transorm, [Volume = -1], [Is3D = true], [Randomization = SoundInstance.Ran
    SoundInstance.InstantiateOnTransform(Clip_Fire, transform, -1, false, SoundInstance.Randomization.Medium);
}
```

This script will do the exact same thing, but it will add medium randomization when you play the fire clip. You can choose between No Randomization, Low Randomization, Medium Randomization, and High Randomization. In this case, the Medium one sounds good to me, the high one would sounds too much unrealistic because we are still shooting with the same gun.

Now, supposing you don't have an access to the audio clip or you are too lazy to make a reference to this audio clip and binding it in the inspector. Just take the sound from the library! !> Before playing a clip from the sounds library, make sure to insert it in first place "Tools > SoundInstance > Sounds Library". Once the clip is in the library, you can access it from anywhere with less than a signe line of code.

csharp

```
public void Fire()
{
    SoundInstance.InstantiateOnTransform(SoundInstance.GetClipFromLibrary("Fire"), transform, 1.0f, false, SoundInstance.Randi
}
```

Assuming you named the sound 'Fire' in the library, this line will do the same thing as the previous code, but without any reference to the audio clip in the inspector. This is usefull when you want to access to 1 audio clip from multiples scripts.

Now, if you want to instantiate the sound on a specific position, the `SoundInstance.InstantiateOnPos()` methode which works the same way, but takes a Vector3 in argument instead of a Transform.

```
public void Fire()
{
    SoundInstance.InstantiateOnPos(SoundInstance.GetClipFromLibrary("Fire"), new Vector3(0, 0, 0), 1.0f, false, SoundInstance.Ri
}
```

Playing music

Playing with music is easy, you barely have some methodes to know the it's done!
Firstly, make sure to insert your music in the musics library, to make this, just open "Tools > SoundInstance > Game Musics Window". Once your music is inserted in the library, you can start playing it in your game with a single line of code:

```
public void StartMusic()
{
    //                                     Name      fade
    SoundInstance.StartMusic("happy", 0.7f);
}
```

With the code above, you can also choose a fading speed. If you don't want any fading, you can just set it to 100 and you shouldn't hear the fading effect.

Now, supposing the player entered a cave, you may want to change the music to something more intimidating? You can just use the same methode and the music will switch with a fading effect (depending on your fading speed).

```
public void EnterCave()
{
    //                                     Name      fade
    SoundInstance.StartMusic("suspense", 0.7f);
}
```

Maybe you want your game to pause and resume the music when the player opens up the pause menu?

```
public void OpenMenu()
{
    //                                     fade
    SoundInstance.PauseMusic(1.5f);
}

public void CloseMenu()
{
    //                                     fade
    SoundInstance.ResumeMusic(1.5f);
}
```

Setting volume

If you play a sound with the default volume argument, it will set the volume to the defaultVolume variable of SoundInstance. If you want to set the default volume, you can easily do it:

```
{
    SoundInstance.defaultVolume = 0.8f;
}
```

The music volume can be set separately, but with the same way:

```
{
    SoundInstance.musicVolume = 0.8f;
}
```

This is all you need to know to use this asset well, now if you want an advenced usage, please go on [Functions](#) to see all the possibilities.