

**MP\_0489 Programación  
multimedia y dispositivos móviles  
UF4. Análisis de motores de juegos**

**4.7. Audio**

# Índice

---

☰	Objetivos	3
☰	Añadir efectos de sonido y música	4
☰	Usar código	9
☰	Resumen	13

# Objetivos

---

Con esta unidad perseguimos los siguientes objetivos:

1 Conocer el sistema de audio en Unity.

2 Aprender cómo insertar sonidos en *GameObjects*.

3 Usar código para manipular el sonido.

---

¡Ánimo y adelante!

# Añadir efectos de sonido y música

---

Agregar sonido a través del código no es difícil. Unity, que tiene un editor visual fácil de usar y funciona con una amplia gama de formatos de archivos de audio, lo hace aún más sencillo.

---

Vamos a utilizar el ejemplo del capítulo de partículas para **trabajar con el sonido**. Comenzaremos por descargar el siguiente archivo.

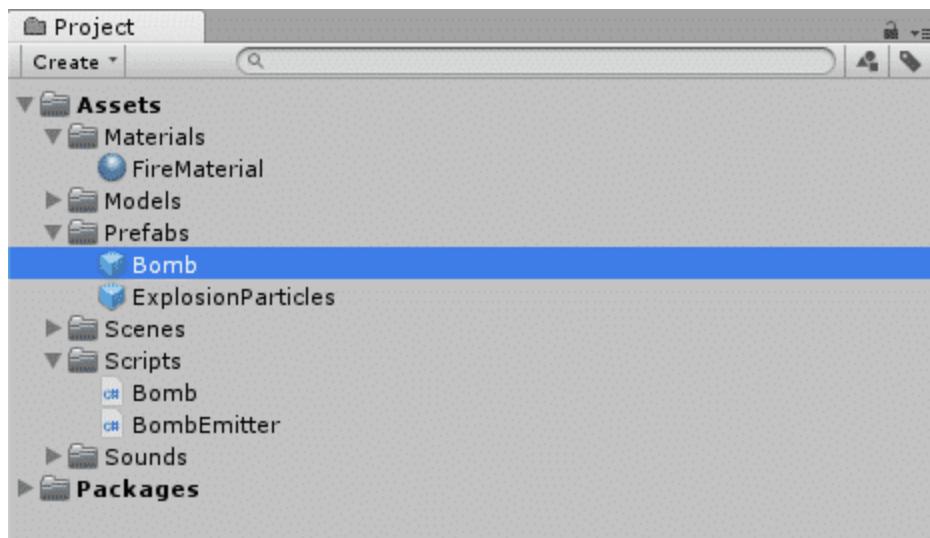


## Añadir efectos de sonido

El proyecto está configurado para dejar caer una bomba de manera repetida en un intervalo de tiempo. Y lo primero que haremos será **agregar el efecto de sonido a la bomba**, para que suene.

1

El proyecto ya se ha configurado con un *prefab* para la bomba. Vamos a seleccionar ese *Prefab* en el panel de *Proyecto*.



Lo primero que necesita este *prefab* es un componente  **AudioSource** para reproducir sonido, que es lo que realmente reproducirá el sonido en un espacio 2D o 3D.

En el espacio 3D, el volumen del sonido puede variar según la distancia a la que se encuentre el  *AudioSource* del objeto que lo escucha (el  *AudioListener* que, en este caso, está en la cámara).

También podemos configurar un  *AudioSource* para reproducir sonido en el espacio 2D, lo que significa que se reproducirá a un volumen constante, independientemente de la distancia desde el  *AudioListener*.

Una cosa importante a tener en cuenta es que el  *AudioSource* es un componente. Eso significa que es un objeto que se hereda de la clase  *MonoBehaviour* de Unity y se puede adjuntar directamente a cualquier  *GameObject*.

2

En el *Inspector*, pulsamos en el botón *Add component* y localizamos *Audio Source*.



El *prefab* de la bomba ya **puede reproducir audio**, pero necesita un *AudioClip* para hacerlo.

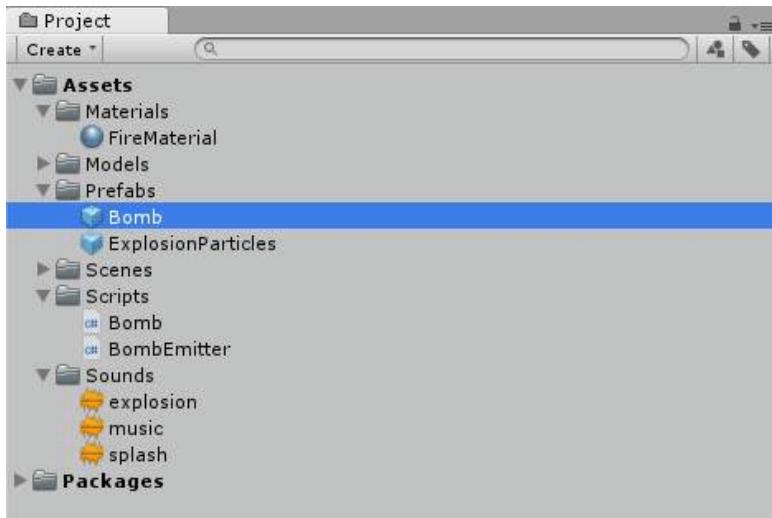


Un *AudioClip* es el archivo de audio real que reproducirá  *AudioSource*, una variable que se puede configurar en  *AudioSource* (es decir, cada  *AudioSource* tendrá un  *AudioClip*).

Podemos ver que, en el componente que hemos agregado, el  *AudioClip* está vacío.

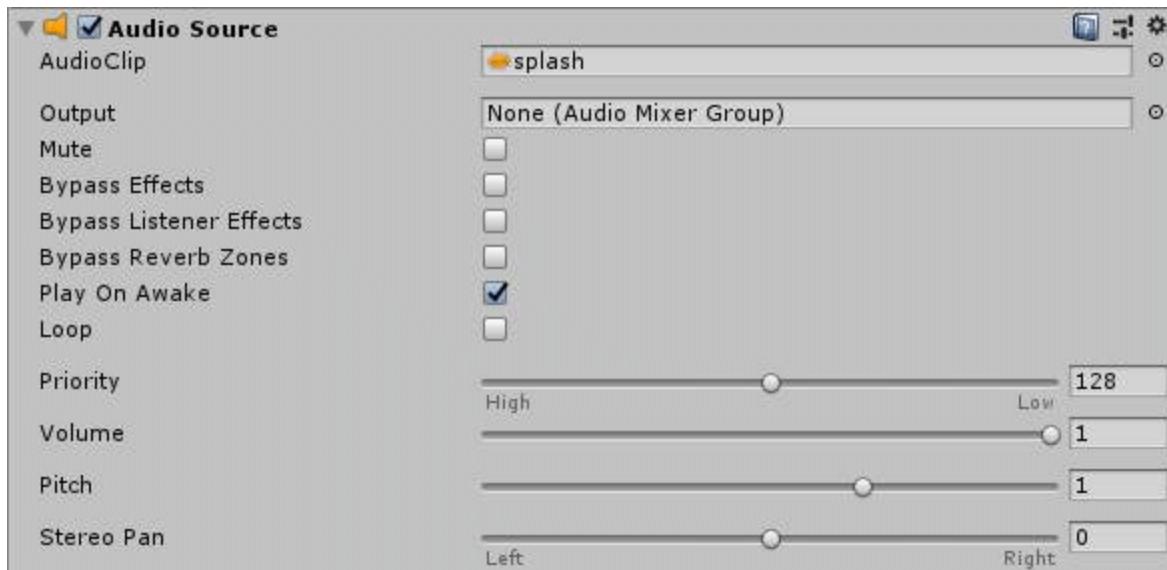
3

Dentro de la carpeta *Assets* del panel de *Proyecto*, encontramos una subcarpeta llamada *Sounds*.



4

Arrastramos el sonido "splash" al hueco de *AudioClip*, que tenemos en el componente *Audio Source* de nuestro *Prefab Bomb*.



5

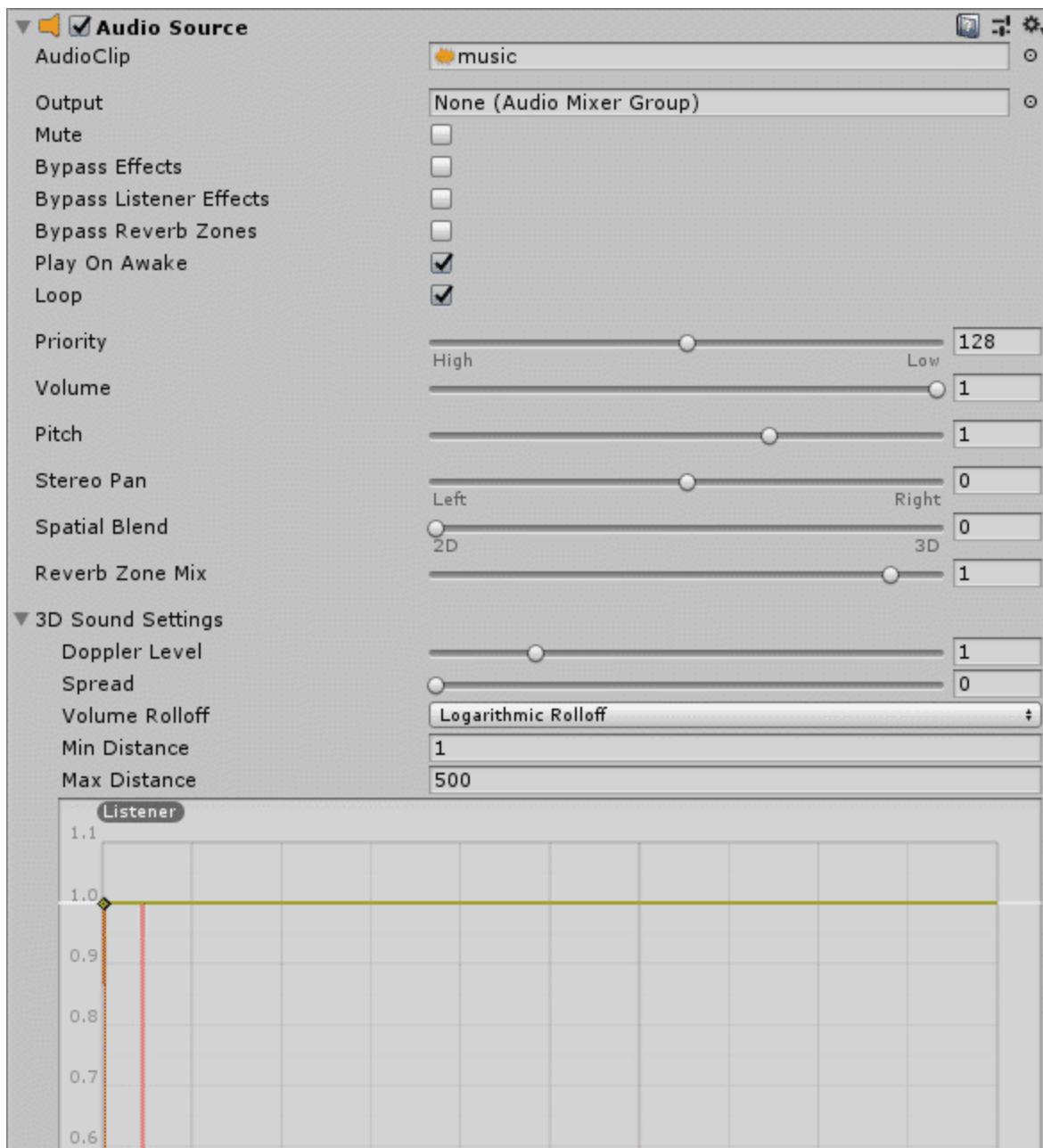
Nos aseguramos que *Play On Awake* esté marcado. Esto significa que el archivo de sonido se reproducirá tan pronto como se cree el objeto, que es exactamente lo que queremos.

## Añadir música de fondo

Vamos a ver cómo podríamos añadir una música de fondo a nuestra escena.

1

Seleccionamos el suelo (*Floor*) en el panel de *Jerarquía* y repetimos la operación que hemos realizado con la bomba, es decir, añadimos un componente *Audio Source* y arrastramos desde *Sounds* en el panel de *Proyecto* el audio "music" al hueco de *AudioClip*.



2

En este caso, nos aseguramos de marcar ***Loop***, de manera que el audio esté repitiéndose constantemente.

Además, podemos **modificar *Volume***, poniéndolo a la mitad para que la música se oiga más suave, dado que estará constantemente sonando.

# Usar código

---

---

En este apartado vamos a mostrarte cómo trabajar con sonidos desde el código.

---

Para ello, haremos que suene una explosión cuando la bomba toque el suelo y explote.

## Añadir eventos y sonidos

En esta ocasión **trabajaremos el sonido mediante código**, indicando que cuando se reciba el evento de colisión suene ese audio. Tendremos que hacerlo en el suelo (*Floor*), puesto que si lo hicieramos en la propia bomba, al colisionar y desaparecer no sonaría la explosión.

1

Con el suelo seleccionado en el panel de *Jerarquía*, pulsamos en **Add component** dentro del panel de *Inspector*, y elegimos **New Script**. Lo llamaremos *Floor* y dentro de él pondremos esto:

```
using UnityEngine;
using System.Collections;

public class Floor : MonoBehaviour {

    public AudioClip explosionSound;
    private AudioSource audioSource;

    // Use this for initialization
    void Start () {

    }

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

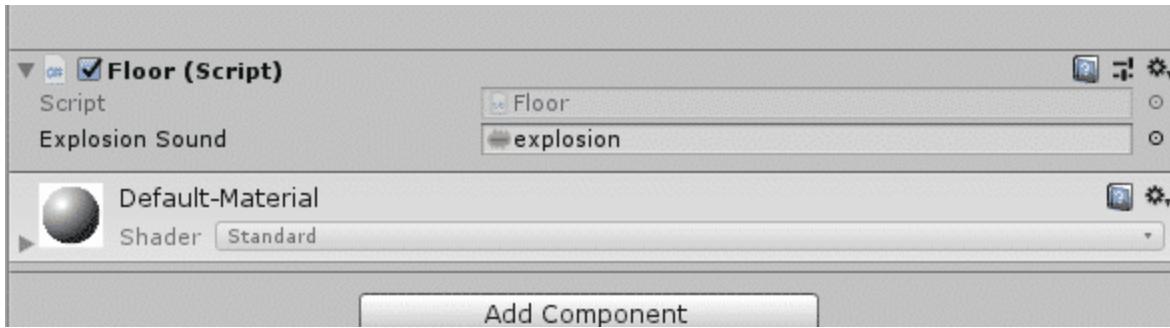
    }

    public void OnCollisionEnter(Collision collision)
    {

    }
}
```

2

Vemos que, dentro del panel de *Inspector*, en el componente de *script* aparece una casilla con el nombre **Explosion Sound**. Ahí arrastraremos el sonido de explosión desde el panel de *Proyecto*.



3

Ahora solamente tenemos que indicar en el código que suene el audio cuando la bomba colisione con el suelo. Lo haremos con la orden *PlayOneShot()*, que reproduce el sonido una vez.

4

La ventaja de usar *PlayOneShot()* es que el sonido se reproducirá hasta su finalización, incluso si se vuelve a llamar a *PlayOneShot()*. Para ello, primero inicializamos el objeto *audioSource* en el método *start()*. Y después introducimos la orden *PlayOneShot()* en el método *onCollisionEnter()*.

```
using UnityEngine;
using System.Collections;

public class Floor : MonoBehaviour {

    public AudioClip explosionSound;
    private AudioSource audioSource;

    // Use this for initialization
    void Start () {
        audioSource = GetComponent<AudioSource>();
    }

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

    }

    public void OnCollisionEnter(Collision collision)
    {
        audioSource.PlayOneShot(explosionSound);
    }
}
```

## Modificar el sonido

Podemos hacer una última mejora, ya que escuchar el mismo sonido una y otra vez es molesto. Para que los sonidos resulten más frescos y menos irritantes, **haremos alguna variación del tono**.

1

Abrimos el *script* de la bomba en el panel de *Proyecto*, y **añadimos este código** en el método *Inicio()*.

```
void Inicio ()  
{  
    AudioSource audioSource = GetComponent <AudioSource> ();  
    audioSource.pitch = UnityEngine.Random.Range(0.8f, 1.5f);  
    audioSource.Play ();  
  
    float randomX = UnityEngine.Random.Range(10f, 100f);  
    float randomY = UnityEngine.Random.Range(10f, 100f);  
    float randomZ = UnityEngine.Random.Range(10f, 100f);  
  
    Rigidbody bomba = GetComponent<Rigidbody>();  
    bomba.AddTorque(randomX, randomY, randomZ);  
}
```

2

El código toma el componente  *AudioSource* y establece la variable de tono en un valor aleatorio entre 0,8 y 1,5.

---

Ahora, cuando la bomba caiga, su sonido irá variando y será diferente cada vez.

---

## Resumen

---

---

---

**Hemos terminado la lección, repasemos los puntos más importantes que hemos tratado.**

- A lo largo de esta unidad hemos visto **cómo utilizar sonidos en Unity** para sumarlos a nuestros proyectos de juegos.
- Hemos aprendido **cómo añadir sonidos a *GameObjects*** para que suenen al entrar en escena.
- También hemos puesto **música de fondo a una escena** y **sonido a un evento de colisión**.
- Y, para finalizar, hemos aprendido **cómo modificar un sonido de manera aleatoria**.

