

UTECA  
ISAAC LÓPEZ  
NOLASCO  
PROGRAMACION  
PARA  
VIDEOJUEGOS II  
SISTEMA DE  
SONIDO

## INDICE

### HERRAMIENTA CONTROL DE

MOVIMIENTO.....	1-3
SCRIPT SONIDO.....	4
SCRIPT AUDIO MANAGER.....	4-6
SCRIPT MENUPAUSA.....	7
SCRIPT UICONTROLLER.....	8
COMO USARLO EN UNITY.....	9-10

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Control : MonoBehaviour
6  {
7
8      public float velocidad;
9      private Rigidbody2D rigidBody;
10     private bool mirandoDerecha = true;
11     private Animator animator;
12     public float salto;
13     private BoxCollider2D boxCollider;
14     public LayerMask sueloCap;
15
16

```

Este es un script de movimiento, no es muy relevante para la herramienta principal (Sistema de sonido), así que si te interesa solo la herramienta principal, puedes ir a la página 4.

Este es el script de control que usaremos para nuestro sistema de sonido, mas que nada yo lo utilice para demostrar un sonido de salto

El float de velocidad, como su propio nombre lo dice sirve para la velocidad con la que el personaje se moverá

El private bool “mirandoDerecha” va a funcionar para nuestro personaje no camine de reversa y se pueda voltear con su respectivo sprite

Private animator funciona para las animaciones del personaje

Public float salto, será la fuerza con la que hará el salto el personaje

Public layer mask, sirve para que se pueda detectar cuando el personaje este tocando el suelo, esto nos servirá mas adelante

```

Mensaje de Unity / 0 referencias
private void Start()
{
    rigidBody = GetComponent<Rigidbody2D>();
    boxCollider = GetComponent<BoxCollider2D>();
    animator = GetComponent<Animator>();
}

Mensaje de Unity / 0 referencias
void Update()
{
    ProcesarMovimiento();
    ProcesarSalto();
}

1 referencia
bool EstaSuelo()
{
    RaycastHit2D raycastHit = Physics2D.BoxCast(boxCollider.bounds.center, new Vector2(boxCollider.bounds.size.x, boxCollider.bounds.size.y), 0f, Vector2.down, 0.2f, sueloCap );
    return raycastHit.collider != null;
}

```

RaycastHit2D funciona como un laser, este determinara y nos ayudara a saber si nuestro objeto esta tocando algo, en este caso el suelo, y nos dira si es true o false

```

1 referencia
void ProcesarSalto()
{
    if(Input.GetKeyDown(KeyCode.Space) && EstaSuelo())
    {
        rigidBody.AddForce(Vector2.up * salto, ForceMode2D.Impulse);
        AudioManager.Instance.PlaySFX("Salto");
    }
}

```

Void ProcesarSalto()

Nos permite que nuestro personaje pueda saltar con la barra espaciadora, añadiendo fuerza hacia arriba, pero aquí interviene el bool, ya que este solo puede saltar si el personaje esta tocando el suelo, ósea que no puede saltar indefinidamente de veces en el aire, también agregue un Sonido a la acción, pero mas adelante explicare como funciona el sistema de sonido.

```

1 referencia
void ProcesarMovimiento()
{
    float inputMovimiento = Input.GetAxis("Horizontal");

    rigidBody.velocity = new Vector2(inputMovimiento * velocidad, rigidBody.velocity.y);

    GestionarOrientacion(inputMovimiento);

    if (inputMovimiento != 0f)
    {
        animator.SetBool("EstaCorriendo", true);
    }
    else
    {
        animator.SetBool("EstaCorriendo", false);
    }
}

```

### Void ProcesarMovimiento()

Este nos funciona al movimiento horizontal del personaje, ya que este utiliza el float de velocidad, utilizando Vector2 osea (x, y), y multiplica el movimiento \* las velocidad que nosotros le pongamos.

También los if nos ayudan a saber si nuestro personaje se esta moviendo o no, y así ejecutar una animación, por ejemplo,

Si nuestro inputMovimiento no es igual a 0, el animator ejecuta la animación de correr, ya que esta moviéndose

Pero si este no se esta moviendo o el valor es igual a 0, este ejecuta una animación de idle, ósea que esta quieto.

```

1 referencia
void GestionarOrientacion(float inputMovimiento)
{
    if ((mirandoDerecha == true && inputMovimiento < 0) || (mirandoDerecha == false && inputMovimiento > 0))
    {
        mirandoDerecha = !mirandoDerecha;
        transform.localScale = new Vector2(-transform.localScale.x, transform.localScale.y);
    }
}

```

Aquí gestionamos la orientacion, ya que nuestro personaje se mueve en un eje horizontal(x) si el personaje mira hacia la derecha es verdadero y el movimiento es menor a 0, significa que se va a rotar hacia la derecha, y de lo contrario a la izquierda

Todo esto transformando la escala, con "transform.localScale" y transformando los valores hacia negativos automaticamente

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5
6 [System.Serializable]
7 public class Sonido
8 {
9     public string nombre;
10    public AudioClip clip;
11 }
12

```

Aquí ya empezamos con el sistema de sonido, este primer script funciona para diferenciar y agregar mas rápido sonidos como saltos, caminado, ataques etc. O agregar música de fondo o de alguna batalla final.

String nombre funciona para que le pongamos nombre a nuestro sonido e identificarlo Public

Audioclip es para agregar sonido a unity

```

1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5
6 public class AudioManager : MonoBehaviour
7 {
8     public Sonido[] Sonidomusica, sfxSonido;
9     public AudioSource musicSource, sfxSource;
10

```

Este es el script de el AudioManager

Como su nombre lo dice, public sonido, nos ayudara para crear funciones para la musica y los sonidos

Public audiosource, serán los clips o audios que descarguemos y usarlos en el audiomanager y asi se puedan ejecutar por separado

```

public static AudioManager Instance;

⌵ Mensaje de Unity | 0 referencias
private void Awake()
{
    if (Instance == null)
    {
        Instance = this;
        DontDestroyOnLoad(gameObject);
    }
    else
    {
        Destroy(gameObject);
    }
}

⌵ Mensaje de Unity | 0 referencias
private void Start()
{
    PlayMusica("Musica de fondo");
}

```

Aquí usamos public static AudioManager Instance, para así modificar cualquier propiedad del mismo

Nuestro programa iniciará reproduciendo la música de fondo que elijamos, ya que el void start, empieza con PlayMusica, más adelante explicare que es

```

1 referencia
public void PlayMusica(string name)
{
    Sonido a = Array.Find(Sonidomusica, x => x.nombre == name);
    if (a == null)
    {
        Debug.Log("No se encontro el sonido");
    }
    else
    {
        musicSource.clip = a.clip;
        musicSource.Play();
    }
}

1 referencia
public void PlaySFX(string name)
{
    Sonido a = Array.Find(sfxSonido, x => x.nombre == name);
    if (a == null)
    {
        Debug.Log("No se encontro el sonido");
    }
    else
    {
        sfxSource.PlayOneShot(a.clip);
    }
}

```

Public void Musica, funciona para que nuestro sonido se pueda reproducir ya que este empezara a buscar en la variable Sonidomusica la musica que hayamos seleccionado, en caso de que este no haya sido encontrado, en la consola se mostrara un mensaje de que no se encontro el sonido, de lo contrario, este empezara a reproducirla.

Lo mismo pasa con los sfx, solo que en su respectiva variable, y estos se reproducen 1 vez según la acción, ya que no son sonidos infinitos

Para eso usamos el PlayOneShot()

```

1 referencia
public void MusicVolume(float volume)
{
    musicSource.volume = volume;
}

1 referencia
public void SfxVolume(float volume)
{
    sfxSource.volume = volume;
}

```

Esto nos ayudara para nuestro menú de sonido, lo explicare más adelante



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class MenuPausa : MonoBehaviour
6  {
7      [SerializeField] private GameObject botonPausa;
8      [SerializeField] private GameObject menuPausa;
9
10     public void Pausa()
11     {
12         Time.timeScale = 0f;
13         botonPausa.SetActive(false);
14         menuPausa.SetActive(true);
15     }
16
17     public void Reanudar()
18     {
19         Time.timeScale = 1f;
20         botonPausa.SetActive(true);
21         menuPausa.SetActive(false);
22     }
23 }

```

Este es el script de nuestro menu, es corto pero funciona para los botones del menu

Public void pausa, congela el tiempo de nuestro programa

Time.timescale = 0f; sirve para congelarlo

Y se activa mientras el boton haya sido presionado, de lo contrario todo seguira normal

El boton Reanudar hace lo mismo, solo que una vez que hayamos presionado pausa, y apretamos el boton reanudar nuestro tiempo volvera a la normalidad con el valor de 1

Time.timescale = 1f; sirve volver a la normalidad

```

Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class UIController : MonoBehaviour
7  {
8      public Slider _musicSlider, _sfxSlider;
9
10
11      public void MusicVolume()
12      {
13          AudioManager.Instance.MusicVolume(_musicSlider.value);
14      }
15
16      public void SFXVolume()
17      {
18          AudioManager.Instance.SfxVolume(_sfxSlider.value);
19      }
20

```

Este es el ultimo script, y funciona con sliders, osea barras de desplazamiento para el sonido,

Utilizamos unityEngineUI, para utilizar y modificar los objetos UI de nuestra escena

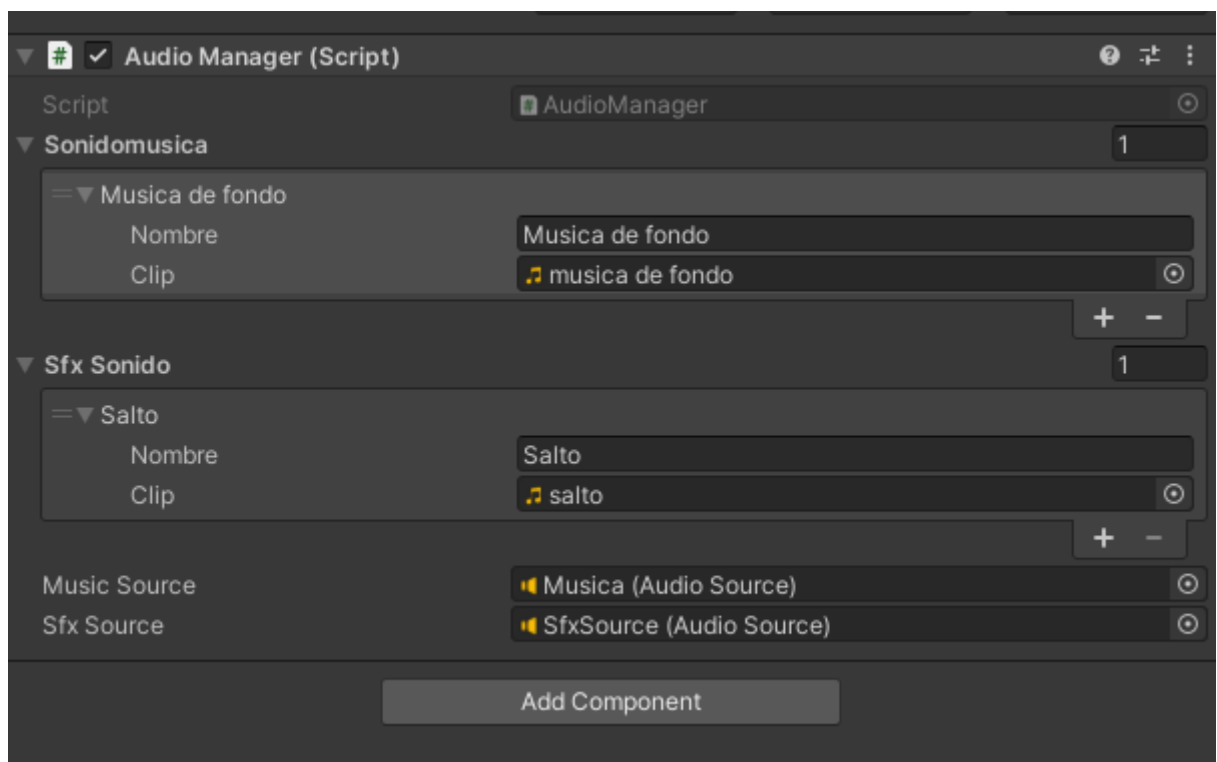
Public slider, seran nuestras barras de desplazamiento

Public void MusicVolume, hara que se modifique el volumen de la musica del programa moviendo las barras, modificando su valor hasta 0

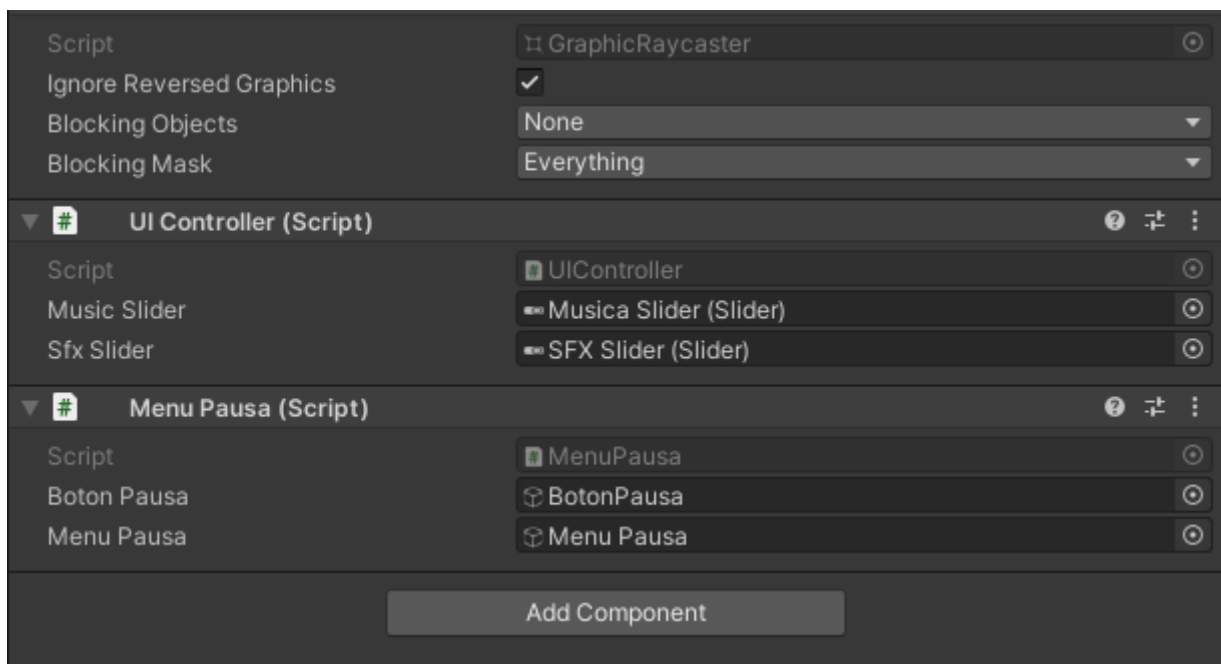
Public void SFXVolume, hara que se modifique el volumen de los efectos del programa moviendo las barras, modificando su valor hasta 0



De esta manera, en unity, podemos ir a nuestro AudioManager



Le agregamos al audio manager su respectivo script de Audio Manager, y nos permitirá agregar sonidos, en el clip, ira el sonido descargado , y le pones su nombre.



Así quedaría El menú, agregando los scripts y botones que deseemos, en este caso, Resume, pausa y los slider del volumen.

```
void ProcesarSalto()
{
    if(Input.GetKeyDown(KeyCode.Space) && EstaSuelo())
    {
        rigidBody.AddForce(Vector2.up * salto, ForceMode2D.Impulse);
        AudioManager.Instance.PlaySFX("Salto");
    }
}
```

De esta manera, vuelvo a recalcar, podemos agregarle sonido a cualquier acción con el código `AudioManager.Instance.PlaySFX("nombre de nuestro sonido o clip");`

## Recomendaciones

Te puedes mover con wasd o las flechas de desplazamiento, saltas con la barra espaciadora

El sonido al inicio esta en un nivel medio, pero aun así ten cuidado si suena fuerte al inicio

Utilice solo 1 musica de fondo y 1 efecto especial (salto) para demostrar la utilidad de la herramienta