SISTEMA DE SONIDO

Índice

este proyecto contiene

Instalación

- [ ] tienes que importar la carpeta de assets y utilizar la carpeta preestablecida de sistema de dialogo ya que esta tiene todos las carpetas necesarias para que pueda funcionar el sistema.

Contenido

primero veremos lo básico

referente al jugador te mueves con las flechas direccionales y las teclas WASD

y el menú se activa con p de papá y desaparece igual

Este proyecto contiene 7 scripts diferentes en referencia a

- [ ] la sentencia de audios

```
[System.Serializable]
1 referencia
public class AudioTracks
{
    public string trackTitle;
    public AudioClip trackClip;

} //la sentencia con el nombre de los audios
```

- [ ] la música y los SFX

Para los SFX

```
[RequireComponent(typeof(AudioSource))]
Script de Unity (2 referencias de recurso) | 0 referencias
public class SFXAudioSource : MonoBehaviour
{
    AudioSource audioSource;
    AudioSettings audioSettings;

    Mensaje de Unity | 0 referencias
    void Start()
    {
        audioSettings = AudioSettings.audioSettings;
        audioSource = GetComponent<AudioSource>();
        audioSource.volume = audioSettings.GetSFXVolume();
        audioSettings.AddMeToSFXAudioSources(audioSource);
    }

    Mensaje de Unity | 0 referencias
    void OnDestroy()
    {
        audioSettings.RemoveMeFromSFXAudioSources(audioSource);
    }
}//este audio es la sentencia para modificar los SFX
```

Para los Track

```
[RequireComponent(typeof(AudioSource))]
Script de Unity (2 referencias de recurso) | 0 referencias
public class MusicAudioSource : MonoBehaviour
{
    AudioSource audioSource;
    AudioSettings audioSettings;

    Mensaje de Unity | 0 referencias
    void Start()
    {
        audioSettings = AudioSettings.audioSettings;
        audioSource = GetComponent<AudioSource>();
        audioSource.volume = audioSettings.GetMusicVolume();
        audioSettings.AddMeToMusicAudioSources(audioSource);
    }

    Mensaje de Unity | 0 referencias
    void OnDestroy()
    {
        audioSettings.RemoveMeFromMusicAudioSources(audioSource);
    }
}// este audio es la sentencia para modificar la musica
```

- [ ] como aparecen en el project los tracks;

```
Script de Unity (1 referencia de recurso) | 0 referencias
public class AudioSetup : MonoBehaviour
{
    [SerializeField] private AudioTracks[] tracks;
    [SerializeField] private AudioSource mainSource;
    public string trackname;

    Mensaje de Unity | 0 referencias
    void Start()
    {
        AudioUtilities.SetAudioSource(mainSource);

        foreach(var item in tracks)
        {
            AudioUtilities.AddTrack(item.trackTitle, item.trackClip);

        }
        AudioUtilities.PlayTrack(trackname); // este codigo cambia el Track desde la pestaña de project
    }
}
```

- [ ] Para añadir mas tracks e iniciarlos por su nombre

```
public class AudioUtilities
{
    public static AudioSource audioSource;
    public static Dictionary<string, AudioClip> tracks = new Dictionary<string, AudioClip>();
1 referencia
public static void SetAudioSource(AudioSource source)
    {
        audioSource = source;
    }

    1 referencia
    public static void AddTrack(string trackTitle,AudioClip trackClip)
    {
        tracks.Add(trackTitle, trackClip); //sirve parab añadir mas Tracks al project
    }
    1 referencia
    public static void PlayTrack(string trackTitle)
    {
        if (tracks.ContainsKey(trackTitle))
        {
            audioSource.clip = tracks[trackTitle];
            audioSource.Play();
        }
        else
        {
            Debug.Log($"Traack:{trackTitle} does not");
        }
    }// inicia el track dependiendo de cual hayas elegido en el project
}
```

☐ para que el menu de controles de sonido funcione correctamente

```
public class UIElementInitializer : MonoBehaviour
{
    3 referencias
    public enum UIElementType {
      SFX_Slider,
      MUSIC_Slider
    }

    public UIElementType type;

    Slider slider;

    Mensaje de Unity | 0 referencias
    private void Start()
    {
        switch (type)
        {
            case UIElementType.SFX_Slider:
                slider = GetComponent<Slider>();
                slider.value = AudioSettings.audioSettings.GetSFXVolume();
                break;

            case UIElementType.MUSIC_Slider:
                slider = GetComponent<Slider>();
                slider.value = AudioSettings.audioSettings.GetMusicVolume();
                break;
        }
        // este script sirve para que los sliders reacciones con los scripts de sfx  y de musica para que bajen y suban volumen
```

☐ y el script principal para modificar los audios sin afectar a la funcionalidad del juego

```
Script de Unity (2 referencias de recurso) | 7 referencias
public class AudioSettings : MonoBehaviour
{
    public static AudioSettings audioSettings;

    [Header("Information - Read Only from inspector")]
    [SerializeField]
    private float musicVolume;
    [SerializeField]
    private float sfxVolume;

    float musicDefaultVolume=0.7f;
    float sfxDefaultVolume = 0.9f;



    string musicVolumeDataName = "music-volume";
    string sfxVolumeDataName = "sfx-volume";

    List<AudioSource> musicAudioSources;
    List<AudioSource> sfxAudioSources;

    private int musicAudioSourcesCount=0;
    private int sfxAudioSourcesCount = 0;

    // todas las sentencias que vamos utilizar en el script para manegar los sliders y el volumen del juego
```

```
Mensaje de Unity | 0 referencias
private void Awake()
{
    audioSettings = this;
    musicAudioSources = new List<AudioSource>();
    sfxAudioSources = new List<AudioSource>();
    LoadSavedSettings();
}

1 referencia
void LoadSavedSettings()
{
    musicVolume = PlayerPrefs.GetFloat(musicVolumeDataName,musicDefaultVolume);
    sfxVolume = PlayerPrefs.GetFloat(sfxVolumeDataName, sfxDefaultVolume);


}
//va guardar los cambios que se realicen al volumen
```

```
public void ChangeMusicVolume(float newVolume)
{
    musicVolume = newVolume;
    PlayerPrefs.SetFloat(musicVolumeDataName, musicVolume);
    SetVolumeToAudioSources(musicAudioSources, musicVolume);
}


0 referencias
public void ChangSFXVolume(float newVolume)
{
    sfxVolume = newVolume;
    PlayerPrefs.SetFloat(sfxVolumeDataName, sfxVolume);
    SetVolumeToAudioSources(sfxAudioSources, sfxVolume);
}
//modifica el volumen de los Tracks utilizados
```

```
void SetVolumeToAudioSources(List<AudioSource> audioSources, float volume)
{
    foreach (AudioSource a in audioSources)
    {
        a.volume = volume;
    }
} //establece un volumen predeterminado para todo
```

```
2 referencias
public float GetMusicVolume()
{
    return musicVolume;
}
2 referencias
public float GetSFXVolume()
{
    return sfxVolume;
}
//les pone el volumen a los tracks
```

```
public void AddMeToMusicAudioSources(AudioSource a)
{
    musicAudioSources.Add(a);
    musicAudioSourcesCount = musicAudioSources.Count;
}

1 referencia
public void RemoveMeFromMusicAudioSources(AudioSource a)
{
    musicAudioSources.Remove(a);
    musicAudioSourcesCount = musicAudioSources.Count;
}
1 referencia
public void AddMeToSFXAudioSources(AudioSource a)
{
    sfxAudioSources.Add(a);
    sfxAudioSourcesCount = sfxAudioSources.Count;
}

1 referencia
public void RemoveMeFromSFXAudioSources(AudioSource a)
{
    sfxAudioSources.Remove(a);
    sfxAudioSourcesCount = sfxAudioSources.Count;
}
//añade y quita los audios necesarios dependiendo de cual se nesecite
```

En referente a lo visual posee:

- [ ] El Jugador
- [ ] Los Arboles
- [ ] el panel de control de sonido

En referente a lo a auditivo posee:

- [ ] Las Música
- [ ] los efectos de pasos

Recomendaciones

utilizar el audio manager para poder cambiar y añadir canciones al juego y poder variar la música de fondo