

Melodreams



Índice

1. Descripción.	2
2. Colliders y tags	2
3. Generación de la música.....	3
4. Scripts.....	3
5. Forma de uso.....	5

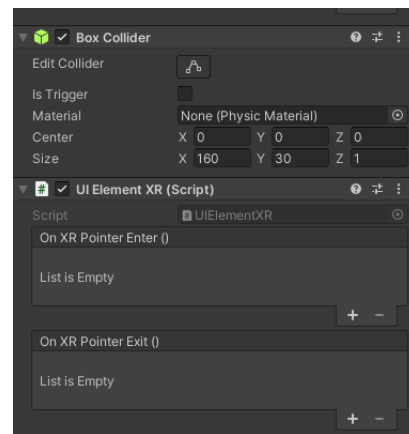
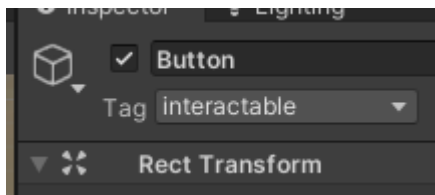
1. Descripción

En este juego, el jugador recibirá una partitura que deberá interpretar utilizando las teclas del piano para generar una variedad de notas musicales. Este paquete incluye lo siguiente:

- Un script encargado de controlar la pulsación de las teclas y la gestión de las notas.
- Otro script que contiene las diferentes escalas, es decir, las notas y su duración, y se encarga de generar las notas en la pantalla.
- Lo mejor de todo es que la implementación es fácil y altamente personalizable, lo que te permite cambiar el piano, las notas, el fondo y la melodía según tus preferencias.

2. Colliders y tags

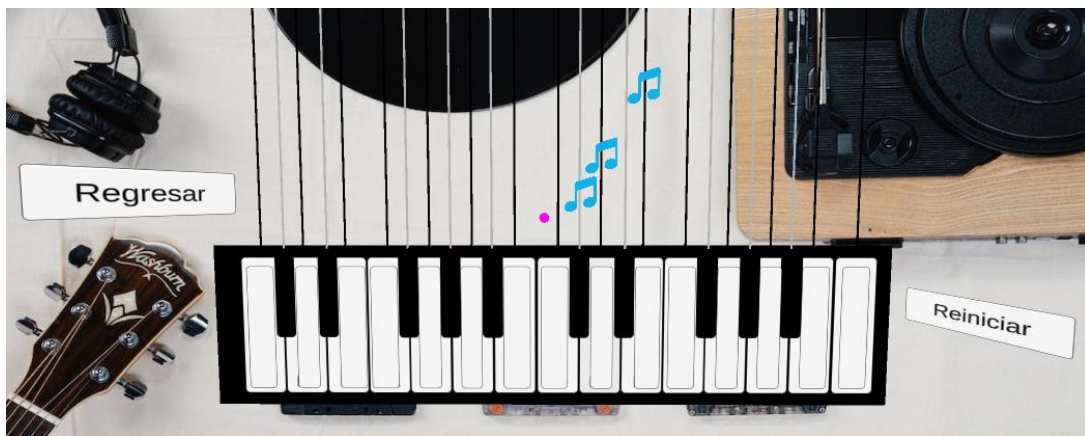
El proyecto hace uso de box colliders para las teclas del piano y los diferentes botones que permiten interactuar con las diferentes escenas. De igual modo estos elementos cuentan con el tag “interactable”, para que la implementación de la cámara que se hizo con un GazeManager pueda identificar estos elementos.



3. Generación de la música

El juego se basa en la idea de generar una melodía o escalas musicales mayores que el usuario debe interpretar. Para lograrlo, se utilizan los siguientes elementos:

- **Nota:** es el componente musical fundamental que se desplaza de la parte superior a la inferior del juego y siempre está asociado a una tecla.
- **Tecla del piano:** esta es la parte interactiva del juego. El jugador debe presionarla para generar el sonido de una nota y "eliminar" la nota correspondiente que se encuentra en la parte superior.
- **Línea de referencia:** es el punto en el que el usuario debe presionar una tecla específica para eliminar la nota y continuar con la melodía. Cuando esto sucede, se crean estrellas en la pantalla.



Las notas están definidas mediante un script (melody.cs) que establece su duración y valor.

```
// define the melody with the notes and durations
melodyNotes = new string[] { "SOL4", "FA4", "MI4", "DO4", "DO4", "SOL3", "DO4", "DO4", "MI4", "DO4", "MI4", "SOL4", "FA4"
melodyDuration = new float[] { 0.5f, 0.25f, 0.5f, 0.5f, 0.5f, 1.0f, 0.5f, 0.5f, 1.0f, 0.5f, 1.0f, 1.0f, 0.5f, 0
```

4. Scripts

Los scripts relevantes para el funcionamiento son como ya se mencionó donde se crean las melodías o escalas, cada uno de estos tendrá su propio script para la generación de la música.

```
Mensaje de Unity | 0 referencias
void Start()
{
    // define the melody with the notes and durations
    melodyNotes = new string[] { "LA3", "SI3", "DOs4", "RE4", "MI4", "FAs4", "SOLs4", "LA"
    melodyDuration = new float[] { 3.0f, 3.0f, 3.0f, 3.0f, 3.0f, 3.0f, 3.0f, 3.0f, 3.0f,
```

```

Mensaje de Unity | 0 referencias
void Start ()
{
    // define the melody with the notes and durations
    melodyNotes = new string[] { "SOL4", "FA4", "MI4", "DO4", "DO4", "SOL3", "DO4", "DO4",
    melodyDuration = new float[] { 0.5f, 0.25f, 0.5f, 0.5f, 0.5f, 1.0f, 0.5f, 0.5f,

```

Para manejar las diferentes escenas se usa el script “GameManager”, en este se crean las funciones para llamar a las diferentes escenas del juego.

```

0 referencias
public void EscenaJuego()
{
    SceneManager.LoadScene("HouseMusic");
}

```

Para poder interactuar con los elementos se implementó un GazeManager, este se le adjunta a la cámara y mas un script UIElementXR para identificar a los elementos UI como botones. A estos mismos se les debe agregar el tag “interactable” para ser reconocidos. El Gazemanager determina el tiempo que se tardará en interactuar con los elementos.

```

[SerializeField] private GameObject gazeBarCanvas;
[SerializeField] Image fillIndicator;
[Tooltip("Time in seg")]
[SerializeField] public float timeForSelection = 2.5f;

```

```

private const float _maxDistance = 10;
private GameObject _gazedAtObject = null;

private readonly string interactableTag = "interactable";
private float _scaleSize = 0.030f;

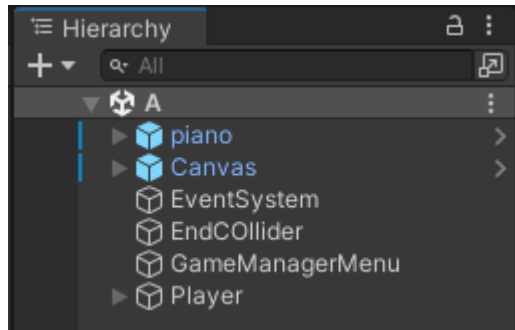
```

```

public class UIElementXR : MonoBehaviour
{
    public UnityEvent OnXRPointerEnter;
    public UnityEvent OnXRPointerExit;
    private Camera xRCamera;
    // Start is called before the first frame update
    void Start()
    {
        xRCamera = CameraPointerManager.Instance.gameObject.GetComponent<Camera>();
    }
}

```

En la jerarquía cada escala cuenta con los siguientes elementos:



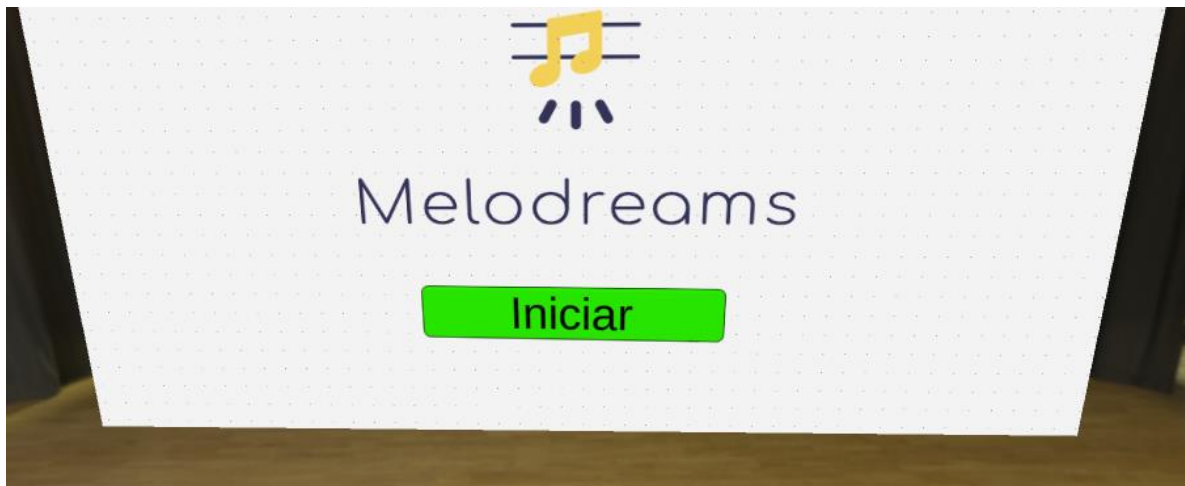
Estos prefabs contienen las notas y su AudioSource para poder escuchar cada una de ellas al interactuar con los botones que se encuentran en Canvas.

Cada tecla cuenta con un boxcollider y cuando la reticula pasa sobre cada tecla estas se ponen de color verde.

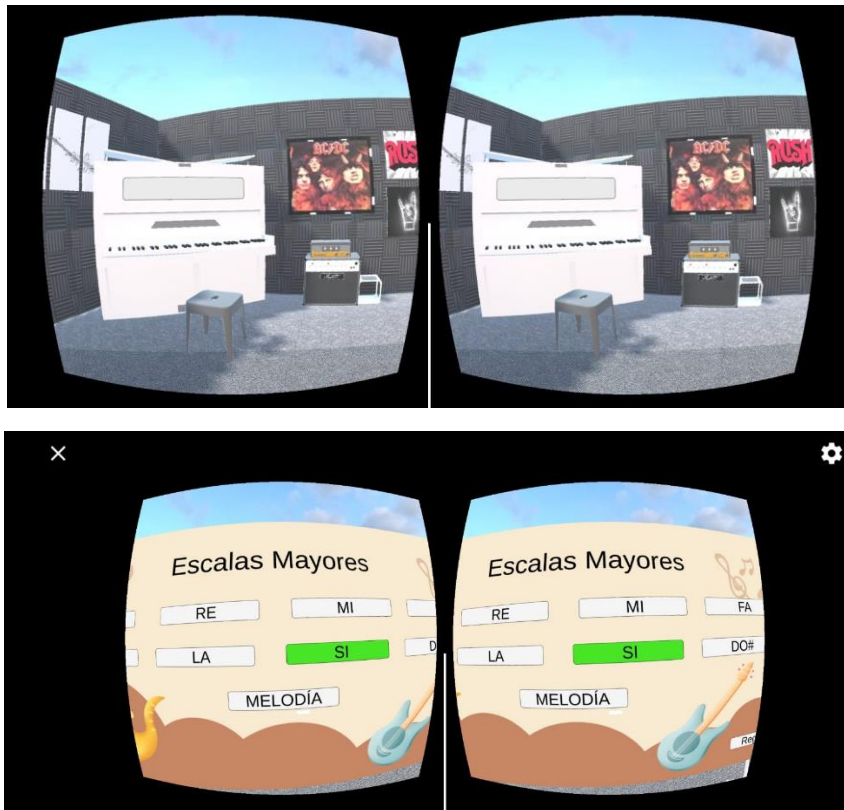


5. Forma de uso

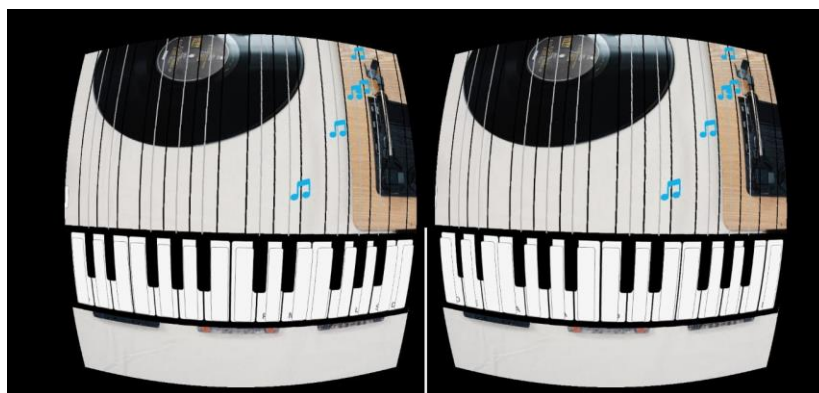
Al iniciar el juego se llega al menú principal en donde se debe centrar la vista en el botón iniciar:



Para así llegar a la HouseMusic en donde se deberá interactuar con la parte superior del piano para desplegar el menú de las diferentes escalas musicales con las que se cuenta:



Una vez seleccionada la escala en el menú se llega a un entorno donde se ven las teclas del piano y las notas van cayendo del cielo sobre líneas que indican la tecla donde se debe tocar. Una vez que dejen de caer las notas se puede reiniciar la música o se puede regresar para elegir otra melodía.



Link video: <https://youtu.be/pREkx5AddOw>

Repositorio: <https://github.com/luisreyes55/ProyectoFinalTSIC>