



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN**

# **PROGRAMACIÓN DE VIDEOJUEGOS**

## **2**

**PROFESOR:**

**JOSÉ FRANCISCO SALGADO RODRIGUEZ**

**INTEGRANTES:**

- SALGADO MORA ARTURO**

**PROYECTO 2. Realidad Aumentada**

**9 DE NOVIEMBRE DEL 2025**

### Clases programadas

#### CarruselDeFiltros

```
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using System.Collections.Generic;
using TMPro;

public class CarruselDeFiltrosUI : MonoBehaviour
{
    [Header("Referencias de Escena")]
    public ARFaceManager faceManager;
    public TextMeshProUGUI textoDisplay;

    [Header("Datos de Filtros")]
    // (facemap1, facemap2...)
    public List<Sprite> iconosFiltro;

    // Texto ("Hombre 1", "Niño"...)
    public List<string> nombresFiltro;
    // -----
    private Material materialCara;
    private int indiceActual = 0;

    void Start()
    {
        // Chequeo
        if (iconosFiltro.Count == 0)
        {
            Debug.LogError("¡No se han asignado iconos de filtro en el script CarruselDeFiltrosUI!");
            return;
        }

        faceManager.trackablesChanged.AddListener(OnCarasCambiadas);

        // Estado inicial con el primer filtro
        ActualizarTexto();
    }

    void OnDestroy()
    {
        faceManager.trackablesChanged.RemoveListener(OnCarasCambiadas);
    }
}
```

```
// Se llama cuando ARCore detecta o actualiza una cara
void OnCarasCambiadas(ARTrackablesChangedEventArgs<ARFace> args)
{
    // Cuando se detecta una cara por primera vez
    foreach (ARFace face in args.added)
    {
        materialCara = face.GetComponent<MeshRenderer>().material;
        AplicarTexturaActual();
    }

    // Si la cara se actualiza (por si se pierde y recupera)
    foreach (ARFace face in args.updated)
    {
        if (materialCara == null)
        {
            materialCara =
face.GetComponent<MeshRenderer>().material;
        }
    }
}

public void SeleccionarFiltro(int indice)
{
    // Otro chequeo
    if (indice < 0 || indice >= iconosFiltro.Count)
    {
        Debug.LogWarning("Índice de filtro no válido: " + indice);
        return;
    }

    // Actualiza el índice
    indiceActual = indice;

    // Aplica la textura al material de la cara
    AplicarTexturaActual();

    // Actualiza el texto en la UI
    ActualizarTexto();
}
```

```
private void AplicarTexturaActual()
{
    if (materialCara != null && iconosFiltro.Count > indiceActual)
    {
        materialCara.SetTexture("_BaseMap",
iconosFiltro[indiceActual].texture);
    }
}

private void ActualizarTexto()
{
    if (textoDisplay != null && nombresFiltro.Count > indiceActual)
    {
        textoDisplay.text = nombresFiltro[indiceActual];
    }
}
```