

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
[RequireComponent(typeof(AudioSource))]
public class dikteA : MonoBehaviour, ITrackableEventHandler
{
    #region PROTECTED_MEMBER_VARIABLES

    protected TrackableBehaviour mTrackableBehaviour;
    protected TrackableBehaviour.Status m_PreviousStatus;
    protected TrackableBehaviour.Status m_NewStatus;

    #endregion // PROTECTED_MEMBER_VARIABLES
    AudioSource audioSource;

    #region UNITY_MONOBEHAVIOUR_METHODS

    protected virtual void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
            mTrackableBehaviour.RegisterTrackableEventHandler(this);

        GameObject audio = GameObject.Find("sesA");
        audioSource = audio.GetComponent<AudioSource>();
        audioSource.Pause();
    }

    protected virtual void OnDestroy()
    {
        if (mTrackableBehaviour)
            mTrackableBehaviour.UnregisterTrackableEventHandler(this);
    }

    #endregion // UNITY_MONOBEHAVIOUR_METHODS

    #region PUBLIC_METHODS

    /// <summary>
    ///     Implementation of the ITrackableEventHandler function called when the
    ///     tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
    {
        m_PreviousStatus = previousStatus;
        m_NewStatus = newStatus;

        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName +
            " " + mTrackableBehaviour.CurrentStatus +
            " -- " + mTrackableBehaviour.CurrentStatusInfo);

        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)

```

```

    {
        OnTrackingFound();
        audioSource.Play();
    }
    else if (previousStatus == TrackableBehaviour.Status.TRACKED &&
            newState == TrackableBehaviour.Status.NO_POSE)
    {
        OnTrackingLost();
        audioSource.Pause();
    }
    else
    {
        // For combo of previousStatus=UNKNOWN + newState=UNKNOWN|NOT_FOUND
        // Vuforia is starting, but tracking has not been lost or found yet
        // Call OnTrackingLost() to hide the augmentations
        OnTrackingLost();
    }
}

#endregion // PUBLIC_METHODS

#region PROTECTED_METHODS

protected virtual void OnTrackingFound()
{
    if (mTrackableBehaviour)
    {
        var rendererComponents =
mTrackableBehaviour.GetComponentsInChildren<Renderer>(true);
        var colliderComponents =
mTrackableBehaviour.GetComponentsInChildren<Collider>(true);
        var canvasComponents =
mTrackableBehaviour.GetComponentsInChildren<Canvas>(true);

        // Enable rendering:
        foreach (var component in rendererComponents)
            component.enabled = true;

        // Enable colliders:
        foreach (var component in colliderComponents)
            component.enabled = true;

        // Enable canvas':
        foreach (var component in canvasComponents)
            component.enabled = true;
    }
}

protected virtual void OnTrackingLost()
{
    if (mTrackableBehaviour)
    {
        var rendererComponents =
mTrackableBehaviour.GetComponentsInChildren<Renderer>(true);
        var colliderComponents =
mTrackableBehaviour.GetComponentsInChildren<Collider>(true);
        var canvasComponents =
mTrackableBehaviour.GetComponentsInChildren<Canvas>(true);

```

```
    // Disable rendering:
    foreach (var component in rendererComponents)
        component.enabled = false;

    // Disable colliders:
    foreach (var component in colliderComponents)
        component.enabled = false;

    // Disable canvas':
    foreach (var component in canvasComponents)
        component.enabled = false;
}
}

#endregion // PROTECTED_METHODS
}
```