/\*==============================================================================

Copyright (c) 2017 PTC Inc. All Rights Reserved.

Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.

All Rights Reserved.

Confidential and Proprietary - Protected under copyright and other laws.

==============================================================================\*/

using UnityEngine;

using Vuforia;

/// <summary>

/// A custom handler that implements the ITrackableEventHandler interface.

/// </summary>

public class DefaultTrackableEventHandler : MonoBehaviour, ITrackableEventHandler

{

#region PRIVATE\_MEMBER\_VARIABLES

protected TrackableBehaviour mTrackableBehaviour;

#endregion // PRIVATE\_MEMBER\_VARIABLES

#region UNTIY\_MONOBEHAVIOUR\_METHODS

protected virtual void Start()

{

mTrackableBehaviour = GetComponent<TrackableBehaviour>();

if (mTrackableBehaviour)

mTrackableBehaviour.RegisterTrackableEventHandler(this);

}

#endregion // UNTIY\_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)

{

if (newStatus == TrackableBehaviour.Status.DETECTED ||

newStatus == TrackableBehaviour.Status.TRACKED ||

newStatus == TrackableBehaviour.Status.EXTENDED\_TRACKED)

{

Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");

OnTrackingFound();

}

else if (previousStatus == TrackableBehaviour.Status.TRACKED &&

newStatus == TrackableBehaviour.Status.NOT\_FOUND)

{

Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");

OnTrackingLost();

}

else

{

// For combo of previousStatus=UNKNOWN + newStatus=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 PRIVATE\_METHODS

protected virtual void OnTrackingFound()

{

var rendererComponents = GetComponentsInChildren<Renderer>(true);

var colliderComponents = GetComponentsInChildren<Collider>(true);

var canvasComponents = 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()

{

var rendererComponents = GetComponentsInChildren<Renderer>(true);

var colliderComponents = GetComponentsInChildren<Collider>(true);

var canvasComponents = 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 // PRIVATE\_METHODS

}