

```

using System;
using System.Collections.Generic;
using UnityEngine;

/// <summary>
/// This is the main behaviour class that encapsulates cloud recognition behaviour.
/// It just has to be added to a Vuforia-enabled Unity scene and will initialize the
target finder and wait for new results.
/// State changes and new results will be sent to registered ICloudRecoEventHandlers
/// </summary>
public class CloudRecoBehaviour : MonoBehaviour, ITrackerEventHandler
{

    #region PRIVATE_MEMBER_VARIABLES

    // ImageTracker reference to avoid lookups
    private ImageTracker mImageTracker;
    // if the TargetFinder is currently initializing
    private bool mCurrentlyInitializing = false;
    // if the TargetFinder was successfully initialized
    private bool mInitSuccessSimpleCloudRecoEventHandler = false;
    // if cloud reco has been started
    private bool mCloudRecoStarted = false;
    // if the OnInitialized callback has been called
    private bool mOnInitializedCalled = false;
    // a list of registered handlers that will be notified of new cloud reco events
    private readonly List<ICloudRecoEventHandler> mHandlers = new
List<ICloudRecoEventHandler>();

    #endregion // PRIVATE_MEMBER_VARIABLES

    #region EXPOSED_PUBLIC_VARIABLES

    // access keys to the online cloud reco database
    public string AccessKey = "";
    public string SecretKey = "";

    // Colors used in the Scanning UI
    public Color ScanlineColor = new Color(1f, 1f, 1f);
    public Color FeaturePointColor = new Color(0.427f, 0.988f, 0.286f);

    #endregion

    SimpleCloudRecoEventHandler

    #region PROPERTIES

    /// <summary>
    /// If cloud has been enabled in the menu
    /// </summary>
    public bool CloudRecoEnabled
    {
        get { return mCloudRecoStarted; }
        set
        {
            if (value) StartCloudReco();
            else StopCloudReco();
        }
    }

    /// <summary>

```

```

    /// If cloud reco has been initialized yet
    /// </summary>
    public bool CloudRecoInitialized
    {
        get { return mInitSuccess; }
    }

#endregion // PROPERTIES

#region PRIVATE_METHODS

    /// <summary>
    /// Initializes the TargetFinder
    /// </summary>
    private void Initialize()
    {
        // start initializing the TargetFinder
        mCurrentlyInitializing = mImageTracker.TargetFinder.StartInit(AccessKey,
SecretKey);

        if (!mCurrentlyInitializing)
            Debug.LogError("CloudRecoBehaviour: TargetFinder initialization failed!");
    }

    // Deinitializes the TargetFinder
    private void Deinitialize()
    {
        mCurrentlyInitializing = !mImageTracker.TargetFinder.Deinit();

        if (mCurrentlyInitializing)
            Debug.LogError("CloudRecoBehaviour: TargetFinder deinitialization
failed!");
        else
            // reset the init success flag
            mInitSuccess = false;
    }

    /// <summary>
    /// Checks the initialization state of the TargetFinder.
    /// If the initialization was successful, the recognition can be started.
    /// </summary>
    private void CheckInitialization()
    {
        TargetFinder.InitState initState = mImageTracker.TargetFinder.GetInitState();
        if (initState == TargetFinder.InitState.INIT_SUCCESS)
        {
            // notify the event handlers that initialization was successful
            foreach (ICloudRecoEventHandler cloudRecoEventHandler in mHandlers)
                cloudRecoEventHandler.OnInitialized();

            // set colors:
            mImageTracker.TargetFinder.SetUIScanlineColor(ScanlineColor);
            mImageTracker.TargetFinder.SetUIPointColor(FeaturePointColor);

            mCurrentlyInitializing = false;
            mInitSuccess = true;

            // init was successfull, start reco:
            StartCloudReco();
        }
    }

```

```

else if (initState < 0) // there has been an initialization error
{
    // notify the event handlers of the init error
    foreach (ICloudRecoEventHandler cloudRecoEventHandler in mHandlers)
        cloudRecoEventHandler.OnInitError(initState);

    mCurrentlyInitializing = false;
}
}

/// <summary>
/// Starts cloud recognition and notifies event handlers
/// </summary>
private void StartCloudReco()
{
    if (mImageTracker != null)
    {
        if (!mCloudRecoStarted)
        {
            // start cloud reco:
            mCloudRecoStarted = mImageTracker.TargetFinder.StartRecognition();

            foreach (ICloudRecoEventHandler cloudRecoEventHandler in mHandlers)
                cloudRecoEventHandler.OnStateChanged(true);
        }
    }
}

/// <summary>
/// Stops cloud recognition and notifies event handlers
/// </summary>
private void StopCloudReco()
{
    if (mCloudRecoStarted)
    {
        // stop cloud reco:
        mCloudRecoStarted = !mImageTracker.TargetFinder.Stop();

        foreach (ICloudRecoEventHandler cloudRecoEventHandler in mHandlers)
            cloudRecoEventHandler.OnStateChanged(false);
    }
}

#endregion // PRIVATE_METHODS

#region PUBLIC_METHODS

/// <summary>
/// Registers an event handler with this CloudRecoBehaviour which will be called
on events
/// </summary>
public void RegisterEventHandler(ICloudRecoEventHandler eventHandler)
{
    mHandlers.Add(eventHandler);

    // in case initialization has already happened:
    if (mOnInitializedCalled)
        eventHandler.OnInitialized();
}

```

```

    /// <summary>
    /// Unregisters an event handler
    /// </summary>
    public bool UnregisterEventHandler(ICloudRecoEventHandler eventHandler)
    {
        return mHandlers.Remove(eventHandler);
    }

#endregion // PUBLIC_METHODS

#region UNITY_MONOBEHAVIOUR_METHODS

    /// <summary>
    /// If we have been initialized before, we will initialize the TargetFinder again
    here when enabled
    /// </summary>
    void OnEnable()
    {
        if (mOnInitializedCalled)
        {
            Initialize();
        }
    }

    /// <summary>
    /// If this component is disabled, the targetfinder is deinitialized
    /// </summary>
    void OnDisable()
    {
        // only if QCAR hasn't already been deinitialized
        if (QCARManager.Instance.Initialized)
        {
            if (mOnInitializedCalled)
            {
                if (mCloudRecoStarted)
                {
                    // stop cloud reco:
                    mCloudRecoStarted = !mImageTracker.TargetFinder.Stop();
                    if (mCloudRecoStarted)
                    {
                        Debug.LogError("Cloud Reco could not be stopped at this
point!");
                    }
                    return;
                }
            }
            Deinitialize();
        }
    }

    /// <summary>
    /// Register for the OnInitialized event at the QCARBehaviour
    /// </summary>
    void Start()
    {
        QCARBehaviour qcarBehaviour =
(QCARBehaviour)FindObjectOfType(typeof(QCARBehaviour));
        if (qcarBehaviour)
        {
            qcarBehaviour.RegisterTrackerEventHandler(this);
        }
    }

```

```

}

/// <summary>
/// Update the TargetFinder each frame
/// </summary>
void Update()
{
    if (mOnInitializedCalled)
    {
        // while we are in init phase, check for the init state
        if (mCurrentlyInitializing)
        {
            CheckInitialization();
        }
        else if (mInitSuccess)
        {
            // update the Target Finder
            TargetFinder.UpdateState updateState =
mImageTracker.TargetFinder.Update();

            // if new results are available, notify the event handlers of them!
            if (updateState == TargetFinder.UpdateState.UPDATE_RESULTS_AVAILABLE)
            {
                IEnumerable<TargetFinder.TargetSearchResult> newResults =
mImageTracker.TargetFinder.GetResults();
                foreach (TargetFinder.TargetSearchResult targetSearchResult in
newResults)
                {
                    foreach (ICloudRecoEventHandler cloudRecoEventHandler in
mHandlers)
                        cloudRecoEventHandler.OnNewSearchResult(targetSearchResult
);
                }
            }
            else if (updateState < 0)
            {
                // notify the event handlers of the update error
                foreach (ICloudRecoEventHandler cloudRecoEventHandler in
mHandlers)
                    cloudRecoEventHandler.OnUpdateError(updateState);
            }
        }
    }
}

#endregion // UNTIY_MONOBEHAVIOUR_METHODS

#region ITrackerEventHandler_IMPLEMENTATION

/// <summary>
/// Initialize after QCAR has been started correctly
/// </summary>
public void OnInitialized()
{
    // get a reference to the Image Tracker, remember it
    mImageTracker = (ImageTracker)TrackerManager.Instance.GetTracker(
Tracker.Type.IMAGE_TRACKER);

    if (mImageTracker != null)
    {

```

```
        // initialized the target finder
        Initialize();
    }

    // remember that the component has been initialized
    mOnInitializedCalled = true;
}

public void OnTrackablesUpdated()
{
    // not used here
}

#endregion // ITrackerEventHandler_IMPLEMENTATION
}
```