ARHitTestExample.cs

Old Script (In the video)

```
using System;
using System.Collections.Generic;

namespace UnityEngine.XR.iOS
{
    public class UnityARHitTestExample : MonoBehaviour
    {
        public Transform m_HitTransform;

        bool HitTestWithResultType (ARPoint point, ARHitTestResultType resultTypes)
        {
            List<ARHitTestResult> hitResults = UnityARSessionNativeInterface.GetARSessionNativeInterface ().HitTest (point, resultTypes);
            if (hitResults.Count > 0) {
                foreach (var hitResult in hitResults) {
                    Debug.Log ("Got hit!");
                    m_HitTransform.position = UnityARMatrixOps.GetPosition (hitResult.worldTransform);
                    m_HitTransform.rotation = UnityARMatrixOps.GetRotation (hitResult.worldTransform);
                    Debug.Log (string.Format ("x:{0:0.######} y:{1:0.######} z:{2:0.######}", m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));
                    return true;
                }
            }
            return false;
        }

        // Update is called once per frame
        void Update () {
            if (Input.touchCount > 0 && m_HitTransform != null)
            {
                var touch = Input.GetTouch(0);
                if (touch.phase == TouchPhase.Began || touch.phase == TouchPhase.Moved)
                {
                    var screenPosition = Camera.main.ScreenToViewportPoint(touch.position);
                    ARPoint point = new ARPoint {
                        x = screenPosition.x,
                        y = screenPosition.y
                    };

                    // prioritize reults types
                    ARHitTestResultType[] resultTypes = {
                        ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent,
                        // if you want to use infinite planes use this:
                        //ARHitTestResultType.ARHitTestResultTypeExistingPlane,
                        ARHitTestResultType.ARHitTestResultTypeHorizontalPlane,
                        ARHitTestResultType.ARHitTestResultTypeFeaturePoint

                    };

                    foreach (ARHitTestResultType resultType in resultTypes)
                    {
                        if (HitTestWithResultType (point, resultType))
                        {
                            return;
                        }
                    }
                }
            }
        }
```

No change

New Script (currently there in your unity project)

```
using System;
using System.Collections.Generic;

namespace UnityEngine.XR.iOS
{
    public class UnityARHitTestExample : MonoBehaviour
    {
        public Transform m_HitTransform;
        public float maxRayDistance = 30.0f;
        public LayerMask collisionLayer = 1 << 10;  //ARKitPlane layer

        bool HitTestWithResultType (ARPoint point, ARHitTestResultType resultTypes)
        {
            List<ARHitTestResult> hitResults = UnityARSessionNativeInterface.GetARSessionNativeInterface ().HitTest (point, resultTypes);
            if (hitResults.Count > 0) {
                foreach (var hitResult in hitResults) {
                    Debug.Log ("Got hit!");
                    m_HitTransform.position = UnityARMatrixOps.GetPosition (hitResult.worldTransform);
                    m_HitTransform.rotation = UnityARMatrixOps.GetRotation (hitResult.worldTransform);
                    Debug.Log (string.Format ("x:{0:0.######} y:{1:0.######} z:{2:0.######}", m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));
                    return true;
                }
            }
            return false;
        }

        // Update is called once per frame
        void Update () {
            #if UNITY_EDITOR   //we will only use this script on the editor side, though there is nothing that would prevent it from working on device
            if (Input.GetMouseButtonDown (0)) {
                Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);
                RaycastHit hit;

                //we'll try to hit one of the plane collider gameobjects that were generated by the plugin
                //effectively similar to calling HitTest with ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent
                if (Physics.Raycast (ray, out hit, maxRayDistance, collisionLayer)) {
                    //we're going to get the position from the contact point
                    m_HitTransform.position = hit.point;
                    Debug.Log (string.Format ("x:{0:0.######} y:{1:0.######} z:{2:0.######}", m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));

                    //and the rotation from the transform of the plane collider
                    m_HitTransform.rotation = hit.transform.rotation;
                }
            }
            #else
            if (Input.touchCount > 0 && m_HitTransform != null)
            {
                var touch = Input.GetTouch(0);
                if (touch.phase == TouchPhase.Began || touch.phase == TouchPhase.Moved)
                {
                    var screenPosition = Camera.main.ScreenToViewportPoint(touch.position);
                    ARPoint point = new ARPoint {
                        x = screenPosition.x,
                        y = screenPosition.y
                    };

                    // prioritize reults types
                    ARHitTestResultType[] resultTypes = {
                        ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent,
                        // if you want to use infinite planes use this:
                        //ARHitTestResultType.ARHitTestResultTypeExistingPlane,
                        ARHitTestResultType.ARHitTestResultTypeHorizontalPlane,
                        ARHitTestResultType.ARHitTestResultTypeFeaturePoint
                    };

                    foreach (ARHitTestResultType resultType in resultTypes)
                    {
                        if (HitTestWithResultType (point, resultType))
                        {
                            return;
                        }
                    }
                }
            }
            #endif
        }
```

Don’t worry about this block

No change

There has been a minor update in the “ARHitTestExample” script since the video was recorded.

The update is very minor and does not effect the functionality of the script in any way.

As shown in the image above there are no changes in the parts denoted by red and blue blocks.

Only change in the new script is the addition of a piece of code which is shown inside the black block.

The code inside the black block does not add any extra functionality to the project, it is just used for debugging purpose.