LAMPIRAN

CameraFocusControler.cs

```
using UnityEngine;
using System.Collections;
using Vuforia;

public class CameraFocusController : MonoBehaviour {

	private bool mVuforiaStarted = false;

	void Start ()
	{
		VuforiaARController vuforia = VuforiaARController.Instance;

		if (vuforia != null)

vuforia.RegisterVuforiaStartedCallback(StartAfterVuforia);
		}

	private void StartAfterVuforia()
	{
		mVuforiaStarted = true;
		SetAutofocus();
		}

	void OnApplicationPause(bool pause)
	{
		if (!pause)
			{
			// App resumed
			if (mVuforiaStarted)
				{
				// App resumed and vuforia already started
				// but lets start it again...
				SetAutofocus(); // This is done because some
android devices lose the auto focus after resume
				// this was a bug in vuforia 4 and 5. I haven't
checked 6, but the code is harmless anyway
				}
			}
		}
```

```
        private void SetAutofocus()
        {
                if
(CameraDevice.Instance.SetFocusMode(CameraDevice.FocusMode.FOCUS_MODE
_CONTINUOUSAUTO))
                        {
                           Debug.Log("Autofocus set");
                        }
                else
                        {
                           // never actually seen a device that doesn't support this,
but just in case
                           Debug.Log("this device doesn't support auto focus");
                        }
            }
}
```

Interaksi.cs

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

public class Interaksi : MonoBehaviour {
   public AudioSource suara;
   public GameObject informasi;
        // Use this for initialization
        void Start () {

        }

        // Update is called once per frame
        void Update () {

        }
   public void GantiScene(string a)
   {
      Application.LoadLevel(a);
   }
   public void Keluar()
   {
      Application.Quit();
   }
```

```
    public void Mulai()
    {
        suara.Play();
    }
    public void ShowInformasi()
    {
        informasi.SetActive(true);
    }
    public void CloseInformasi()
    {
        informasi.SetActive(false);
    }
}
```

RotasiObjek.cs

```
using UnityEngine;
using System.Collections;
using System.IO;

public class RotasiObjek : MonoBehaviour {

        public float scalingSpeed = 0.05f;
        public float rotationSpeed = 70.0f;
        public float translationSpeed = 5.0f;
        public GameObject objek;
        //public float minScale = 0.25f;
        //public float maxScale = 0.7f;
        //        public GameObject Model;
        bool repeatScaleUp = false;
        bool repeatScaleDown = false;
        bool repeatRotateLeft = false;
        bool repeatRotateRight = false;

        void Update ()
        {


    if (repeatRotateRight) {
                        RotationRightButton();
                }

                if (repeatRotateLeft) {
                        RotationLeftButton();
```

```
		}


	}

	public void CloseAppButton ()
	{
		Application.Quit ();
	}

	public void RotationRightButton ()
	{

		objek.gameObject.transform.Rotate (0, -rotationSpeed *
Time.deltaTime, 0);
		//tanaman.gameObject.transform.Rotate (0, -rotationSpeed *
Time.deltaTime, 0);



	}

	public void RotationLeftButton ()
	{

		objek.gameObject.transform.Rotate (0, rotationSpeed *
Time.deltaTime, 0 );
		//tanaman.gameObject.transform.Rotate (0, rotationSpeed *
Time.deltaTime, 0);


	}

	public void RotationRightButtonRepeat ()
	{

		repeatRotateRight=true;
	}

	public void RotationLeftButtonRepeat ()
	{

		repeatRotateLeft=true;
	}
```

```
	public void RotateLeftButtonOff ()
	{
		repeatRotateLeft = false;
		Debug.Log ("Off");
	}

	public void RotateRightButtonOff ()
	{
		repeatRotateRight = false;
		Debug.Log ("Off");
	}




}
```

ZoomSlider.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ZoomSlider : MonoBehaviour {
   public GameObject objek;


   public void Slider_Changed(float newValue)
   {
      Vector3 pos = objek.transform.localScale;
      pos.x = newValue;
      pos.y = newValue;
      pos.z = newValue;

      objek.transform.localScale = pos;
   }

   // Use this for initialization
   void Start () {
```

```
    }

        // Update is called once per frame
        void Update () {

    }
}
```

DefaultTrackableEventHandler.cs

```
/*==============================================================
==================
Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
==============================================================
================*/

using UnityEngine;

namespace Vuforia
{
    /// <summary>
    /// A custom handler that implements the ITrackableEventHandler interface.
    /// </summary>
    public class DefaultTrackableEventHandler : MonoBehaviour,
                                                ITrackableEventHandler
    {
        #region PRIVATE_MEMBER_VARIABLES
        public AudioSource suara;
        private TrackableBehaviour mTrackableBehaviour;

        #endregion // PRIVATE_MEMBER_VARIABLES



        #region UNTIY_MONOBEHAVIOUR_METHODS

        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)
        {
            OnTrackingFound();
        }
        else
        {
            OnTrackingLost();
        }
    }

    #endregion // PUBLIC_METHODS



    #region PRIVATE_METHODS


    private void OnTrackingFound()
    {
        Renderer[] rendererComponents =
GetComponentsInChildren<Renderer>(true);
        Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);

        // Enable rendering:
```

```
            foreach (Renderer component in rendererComponents)
            {
                component.enabled = true;
            }

            // Enable colliders:
            foreach (Collider component in colliderComponents)
            {
                component.enabled = true;
            }
            suara.Play();

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


        private void OnTrackingLost()
        {
            Renderer[] rendererComponents =
GetComponentsInChildren<Renderer>(true);
            Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);

            // Disable rendering:
            foreach (Renderer component in rendererComponents)
            {
                component.enabled = false;
            }

            // Disable colliders:
            foreach (Collider component in colliderComponents)
            {
                component.enabled = false;
            }

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

        #endregion // PRIVATE_METHODS
    }
}
```

VuforiaBehaviour.cs

```
/*============================================================================
==========
Copyright (c) 2016 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;

namespace Vuforia
{
    /// <summary>
    /// The VuforiaBehaviour class handles tracking and triggers native video
    /// background rendering. The class updates all Trackables in the scene.
    /// </summary>
    public class VuforiaBehaviour : VuforiaAbstractBehaviour
    {
        protected override void Awake()
        {
            AddOSSpecificExternalDatasetSearchDirs();

            gameObject.AddComponent<ComponentFactoryStarterBehaviour>();

            base.Awake();
        }

        private static VuforiaBehaviour mVuforiaBehaviour= null;

        /// <summary>
        /// A simple static singleton getter to the VuforiaBehaviour (if present in the
scene)
        /// Will return null if no VuforiaBehaviour has been instantiated in the scene.
        /// </summary>
        public static VuforiaBehaviour Instance
        {
            get
            {
                if (mVuforiaBehaviour == null)
                    mVuforiaBehaviour = FindObjectOfType<VuforiaBehaviour>();

                return mVuforiaBehaviour;
            }
```

```
    }


    /// <summary>
    /// This method inserts new dataset search roots for datasets defined in
StreamingAssets/QCAR.  This may
    /// be used to streamline the "Split Application Binary" Unity feature under the
Android plugin.  This method is
    /// called before the datasets are loaded in the Start()-method.
    /// </summary>
    private void AddOSSpecificExternalDatasetSearchDirs()
    {
#if UNITY_ANDROID
        if (Application.platform == RuntimePlatform.Android)
        {
            var databaseLoader = DatabaseLoadARController.Instance;

            // Get the external storage directory
            AndroidJavaClass jclassEnvironment = new
AndroidJavaClass("android.os.Environment");
            AndroidJavaObject jobjFile =
jclassEnvironment.CallStatic<AndroidJavaObject>("getExternalStorageDirectory");
            string externalStorageDirectory =
jobjFile.Call<string>("getAbsolutePath");

            // Get the package name
            AndroidJavaObject jobjActivity = new
AndroidJavaClass("com.unity3d.player.UnityPlayer").GetStatic<AndroidJavaObject
>("currentActivity");
            string packageName = jobjActivity.Call<string>("getPackageName");

            // Add some best practice search directories
            //
            // Assumes just Vufroria datasets extracted to the files directory
            databaseLoader.AddExternalDatasetSearchDir(externalStorageDirectory +
"/Android/data/" + packageName + "/files/");

            // Assume entire StreamingAssets dir is extracted here and our datasets are
in the "Vuforia" directory
            databaseLoader.AddExternalDatasetSearchDir(externalStorageDirectory +
"/Android/data/" + packageName + "/files/Vuforia/");

            // Assume entire StreamingAssets dir is extracted here and our datasets are
in the "QCAR" directory
```

```
            databaseLoader.AddExternalDatasetSearchDir(externalStorageDirectory +
"/Android/data/" + packageName + "/files/QCAR/");
        }
#endif //UNITY_ANDROID
      }
   }
}
```

ImageTargetBehaviour.cs

```
/*==============================================================
==================
Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and other laws.
==============================================================
================*/

using System.Collections.Generic;
using UnityEngine;

namespace Vuforia
{
   /// <summary>
   /// This class serves both as an augmentation definition for an ImageTarget in the
editor
   /// as well as a tracked image target result at runtime
   /// </summary>
   public class ImageTargetBehaviour : ImageTargetAbstractBehaviour
   {
   }
}
```