using System;

using UnityEngine;

using UnityEngine.EventSystems;

namespace UnityStandardAssets.CrossPlatformInput

{

public class AxisTouchButton : MonoBehaviour, IPointerDownHandler, IPointerUpHandler

{

// designed to work in a pair with another axis touch button

// (typically with one having -1 and one having 1 axisValues)

public string axisName = "Horizontal"; // The name of the axis

public float axisValue = 1; // The axis that the value has

public float responseSpeed = 3; // The speed at which the axis touch button responds

public float returnToCentreSpeed = 3; // The speed at which the button will return to its centre

AxisTouchButton m\_PairedWith; // Which button this one is paired with

CrossPlatformInputManager.VirtualAxis m\_Axis; // A reference to the virtual axis as it is in the cross platform input

void OnEnable()

{

if (!CrossPlatformInputManager.AxisExists(axisName))

{

// if the axis doesnt exist create a new one in cross platform input

m\_Axis = new CrossPlatformInputManager.VirtualAxis(axisName);

CrossPlatformInputManager.RegisterVirtualAxis(m\_Axis);

}

else

{

m\_Axis = CrossPlatformInputManager.VirtualAxisReference(axisName);

}

FindPairedButton();

}

void FindPairedButton()

{

// find the other button witch which this button should be paired

// (it should have the same axisName)

var otherAxisButtons = FindObjectsOfType(typeof(AxisTouchButton)) as AxisTouchButton[];

if (otherAxisButtons != null)

{

for (int i = 0; i < otherAxisButtons.Length; i++)

{

if (otherAxisButtons[i].axisName == axisName && otherAxisButtons[i] != this)

{

m\_PairedWith = otherAxisButtons[i];

}

}

}

}

void OnDisable()

{

// The object is disabled so remove it from the cross platform input system

m\_Axis.Remove();

}

public void OnPointerDown(PointerEventData data)

{

if (m\_PairedWith == null)

{

FindPairedButton();

}

// update the axis and record that the button has been pressed this frame

m\_Axis.Update(Mathf.MoveTowards(m\_Axis.GetValue, axisValue, responseSpeed \* Time.deltaTime));

}

public void OnPointerUp(PointerEventData data)

{

m\_Axis.Update(Mathf.MoveTowards(m\_Axis.GetValue, 0, responseSpeed \* Time.deltaTime));

}

}

}