using System;

using UnityEngine;

using UnityStandardAssets.CrossPlatformInput;

namespace UnityStandardAssets.Vehicles.Ball

{

public class BallUserControl : MonoBehaviour

{

private Ball ball; // Reference to the ball controller.

private Vector3 move;

// the world-relative desired move direction, calculated from the camForward and user input.

private Transform cam; // A reference to the main camera in the scenes transform

private Vector3 camForward; // The current forward direction of the camera

private bool jump; // whether the jump button is currently pressed

private void Awake()

{

// Set up the reference.

ball = GetComponent<Ball>();

// get the transform of the main camera

if (Camera.main != null)

{

cam = Camera.main.transform;

}

else

{

Debug.LogWarning(

"Warning: no main camera found. Ball needs a Camera tagged \"MainCamera\", for camera-relative controls.");

// we use world-relative controls in this case, which may not be what the user wants, but hey, we warned them!

}

}

private void Update()

{

// Get the axis and jump input.

float h = CrossPlatformInputManager.GetAxis("Horizontal");

float v = CrossPlatformInputManager.GetAxis("Vertical");

jump = CrossPlatformInputManager.GetButton("Jump");

// calculate move direction

if (cam != null)

{

// calculate camera relative direction to move:

camForward = Vector3.Scale(cam.forward, new Vector3(1, 0, 1)).normalized;

move = (v\*camForward + h\*cam.right).normalized;

}

else

{

// we use world-relative directions in the case of no main camera

move = (v\*Vector3.forward + h\*Vector3.right).normalized;

}

}

private void FixedUpdate()

{

// Call the Move function of the ball controller

ball.Move(move, jump);

jump = false;

}

}

}