using System;

using UnityEngine;

using UnityStandardAssets.Utility;

namespace UnityStandardAssets.Characters.FirstPerson

{

public class HeadBob : MonoBehaviour

{

public Camera Camera;

public CurveControlledBob motionBob = new CurveControlledBob();

public LerpControlledBob jumpAndLandingBob = new LerpControlledBob();

public RigidbodyFirstPersonController rigidbodyFirstPersonController;

public float StrideInterval;

[Range(0f, 1f)] public float RunningStrideLengthen;

// private CameraRefocus m\_CameraRefocus;

private bool m\_PreviouslyGrounded;

private Vector3 m\_OriginalCameraPosition;

private void Start()

{

motionBob.Setup(Camera, StrideInterval);

m\_OriginalCameraPosition = Camera.transform.localPosition;

// m\_CameraRefocus = new CameraRefocus(Camera, transform.root.transform, Camera.transform.localPosition);

}

private void Update()

{

// m\_CameraRefocus.GetFocusPoint();

Vector3 newCameraPosition;

if (rigidbodyFirstPersonController.Velocity.magnitude > 0 && rigidbodyFirstPersonController.Grounded)

{

Camera.transform.localPosition = motionBob.DoHeadBob(rigidbodyFirstPersonController.Velocity.magnitude\*(rigidbodyFirstPersonController.Running ? RunningStrideLengthen : 1f));

newCameraPosition = Camera.transform.localPosition;

newCameraPosition.y = Camera.transform.localPosition.y - jumpAndLandingBob.Offset();

}

else

{

newCameraPosition = Camera.transform.localPosition;

newCameraPosition.y = m\_OriginalCameraPosition.y - jumpAndLandingBob.Offset();

}

Camera.transform.localPosition = newCameraPosition;

if (!m\_PreviouslyGrounded && rigidbodyFirstPersonController.Grounded)

{

StartCoroutine(jumpAndLandingBob.DoBobCycle());

}

m\_PreviouslyGrounded = rigidbodyFirstPersonController.Grounded;

// m\_CameraRefocus.SetFocusPoint();

}

}

}