using System;

using UnityEngine;

namespace UnityStandardAssets.\_2D

{

public class Camera2DFollow : MonoBehaviour

{

public Transform target;

public float damping = 1;

public float lookAheadFactor = 3;

public float lookAheadReturnSpeed = 0.5f;

public float lookAheadMoveThreshold = 0.1f;

private float m\_OffsetZ;

private Vector3 m\_LastTargetPosition;

private Vector3 m\_CurrentVelocity;

private Vector3 m\_LookAheadPos;

// Use this for initialization

private void Start()

{

m\_LastTargetPosition = target.position;

m\_OffsetZ = (transform.position - target.position).z;

transform.parent = null;

}

// Update is called once per frame

private void Update()

{

// only update lookahead pos if accelerating or changed direction

float xMoveDelta = (target.position - m\_LastTargetPosition).x;

bool updateLookAheadTarget = Mathf.Abs(xMoveDelta) > lookAheadMoveThreshold;

if (updateLookAheadTarget)

{

m\_LookAheadPos = lookAheadFactor\*Vector3.right\*Mathf.Sign(xMoveDelta);

}

else

{

m\_LookAheadPos = Vector3.MoveTowards(m\_LookAheadPos, Vector3.zero, Time.deltaTime\*lookAheadReturnSpeed);

}

Vector3 aheadTargetPos = target.position + m\_LookAheadPos + Vector3.forward\*m\_OffsetZ;

Vector3 newPos = Vector3.SmoothDamp(transform.position, aheadTargetPos, ref m\_CurrentVelocity, damping);

transform.position = newPos;

m\_LastTargetPosition = target.position;

}

}

}