using System;

using UnityEngine;

namespace UnityStandardAssets.Cameras

{

public abstract class AbstractTargetFollower : MonoBehaviour

{

public enum UpdateType // The available methods of updating are:

{

FixedUpdate, // Update in FixedUpdate (for tracking rigidbodies).

LateUpdate, // Update in LateUpdate. (for tracking objects that are moved in Update)

ManualUpdate, // user must call to update camera

}

[SerializeField] protected Transform m\_Target; // The target object to follow

[SerializeField] private bool m\_AutoTargetPlayer = true; // Whether the rig should automatically target the player.

[SerializeField] private UpdateType m\_UpdateType; // stores the selected update type

protected Rigidbody targetRigidbody;

protected virtual void Start()

{

// if auto targeting is used, find the object tagged "Player"

// any class inheriting from this should call base.Start() to perform this action!

if (m\_AutoTargetPlayer)

{

FindAndTargetPlayer();

}

if (m\_Target == null) return;

targetRigidbody = m\_Target.GetComponent<Rigidbody>();

}

private void FixedUpdate()

{

// we update from here if updatetype is set to Fixed, or in auto mode,

// if the target has a rigidbody, and isn't kinematic.

if (m\_AutoTargetPlayer && (m\_Target == null || !m\_Target.gameObject.activeSelf))

{

FindAndTargetPlayer();

}

if (m\_UpdateType == UpdateType.FixedUpdate)

{

FollowTarget(Time.deltaTime);

}

}

private void LateUpdate()

{

// we update from here if updatetype is set to Late, or in auto mode,

// if the target does not have a rigidbody, or - does have a rigidbody but is set to kinematic.

if (m\_AutoTargetPlayer && (m\_Target == null || !m\_Target.gameObject.activeSelf))

{

FindAndTargetPlayer();

}

if (m\_UpdateType == UpdateType.LateUpdate)

{

FollowTarget(Time.deltaTime);

}

}

public void ManualUpdate()

{

// we update from here if updatetype is set to Late, or in auto mode,

// if the target does not have a rigidbody, or - does have a rigidbody but is set to kinematic.

if (m\_AutoTargetPlayer && (m\_Target == null || !m\_Target.gameObject.activeSelf))

{

FindAndTargetPlayer();

}

if (m\_UpdateType == UpdateType.ManualUpdate)

{

FollowTarget(Time.deltaTime);

}

}

protected abstract void FollowTarget(float deltaTime);

public void FindAndTargetPlayer()

{

// auto target an object tagged player, if no target has been assigned

var targetObj = GameObject.FindGameObjectWithTag("Player");

if (targetObj)

{

SetTarget(targetObj.transform);

}

}

public virtual void SetTarget(Transform newTransform)

{

m\_Target = newTransform;

}

public Transform Target

{

get { return m\_Target; }

}

}

}