using System;

using System.Collections;

using UnityEngine;

namespace UnityStandardAssets.Utility

{

public class DragRigidbody : MonoBehaviour

{

const float k\_Spring = 50.0f;

const float k\_Damper = 5.0f;

const float k\_Drag = 10.0f;

const float k\_AngularDrag = 5.0f;

const float k\_Distance = 0.2f;

const bool k\_AttachToCenterOfMass = false;

private SpringJoint m\_SpringJoint;

private void Update()

{

// Make sure the user pressed the mouse down

if (!Input.GetMouseButtonDown(0))

{

return;

}

var mainCamera = FindCamera();

// We need to actually hit an object

RaycastHit hit = new RaycastHit();

if (

!Physics.Raycast(mainCamera.ScreenPointToRay(Input.mousePosition).origin,

mainCamera.ScreenPointToRay(Input.mousePosition).direction, out hit, 100,

Physics.DefaultRaycastLayers))

{

return;

}

// We need to hit a rigidbody that is not kinematic

if (!hit.rigidbody || hit.rigidbody.isKinematic)

{

return;

}

if (!m\_SpringJoint)

{

var go = new GameObject("Rigidbody dragger");

Rigidbody body = go.AddComponent<Rigidbody>();

m\_SpringJoint = go.AddComponent<SpringJoint>();

body.isKinematic = true;

}

m\_SpringJoint.transform.position = hit.point;

m\_SpringJoint.anchor = Vector3.zero;

m\_SpringJoint.spring = k\_Spring;

m\_SpringJoint.damper = k\_Damper;

m\_SpringJoint.maxDistance = k\_Distance;

m\_SpringJoint.connectedBody = hit.rigidbody;

StartCoroutine("DragObject", hit.distance);

}

private IEnumerator DragObject(float distance)

{

var oldDrag = m\_SpringJoint.connectedBody.drag;

var oldAngularDrag = m\_SpringJoint.connectedBody.angularDrag;

m\_SpringJoint.connectedBody.drag = k\_Drag;

m\_SpringJoint.connectedBody.angularDrag = k\_AngularDrag;

var mainCamera = FindCamera();

while (Input.GetMouseButton(0))

{

var ray = mainCamera.ScreenPointToRay(Input.mousePosition);

m\_SpringJoint.transform.position = ray.GetPoint(distance);

yield return null;

}

if (m\_SpringJoint.connectedBody)

{

m\_SpringJoint.connectedBody.drag = oldDrag;

m\_SpringJoint.connectedBody.angularDrag = oldAngularDrag;

m\_SpringJoint.connectedBody = null;

}

}

private Camera FindCamera()

{

if (GetComponent<Camera>())

{

return GetComponent<Camera>();

}

return Camera.main;

}

}

}