using System;

using UnityEngine;

namespace UnityStandardAssets.Vehicles.Ball

{

public class Ball : MonoBehaviour

{

[SerializeField] private float m\_MovePower = 5; // The force added to the ball to move it.

[SerializeField] private bool m\_UseTorque = true; // Whether or not to use torque to move the ball.

[SerializeField] private float m\_MaxAngularVelocity = 25; // The maximum velocity the ball can rotate at.

[SerializeField] private float m\_JumpPower = 2; // The force added to the ball when it jumps.

private const float k\_GroundRayLength = 1f; // The length of the ray to check if the ball is grounded.

private Rigidbody m\_Rigidbody;

private void Start()

{

m\_Rigidbody = GetComponent<Rigidbody>();

// Set the maximum angular velocity.

GetComponent<Rigidbody>().maxAngularVelocity = m\_MaxAngularVelocity;

}

public void Move(Vector3 moveDirection, bool jump)

{

// If using torque to rotate the ball...

if (m\_UseTorque)

{

// ... add torque around the axis defined by the move direction.

m\_Rigidbody.AddTorque(new Vector3(moveDirection.z, 0, -moveDirection.x)\*m\_MovePower);

}

else

{

// Otherwise add force in the move direction.

m\_Rigidbody.AddForce(moveDirection\*m\_MovePower);

}

// If on the ground and jump is pressed...

if (Physics.Raycast(transform.position, -Vector3.up, k\_GroundRayLength) && jump)

{

// ... add force in upwards.

m\_Rigidbody.AddForce(Vector3.up\*m\_JumpPower, ForceMode.Impulse);

}

}

}

}