﻿using UnityEngine;

public class ProjectileDragging : MonoBehaviour {

public float maxStretch = 3.0f;

public LineRenderer catapultLineFront;

public LineRenderer catapultLineBack;

private SpringJoint2D spring;

private Transform catapult;

private Ray rayToMouse;

private Ray leftCatapultToProjectile;

private float maxStretchSqr;

private float circleRadius;

private bool clickedOn;

private Vector2 prevVelocity;

void Awake () {

spring = GetComponent <SpringJoint2D> ();

catapult = spring.connectedBody.transform;

}

void Start () {

LineRendererSetup ();

rayToMouse = new Ray(catapult.position, Vector3.zero);

leftCatapultToProjectile = new Ray(catapultLineFront.transform.position, Vector3.zero);

maxStretchSqr = maxStretch \* maxStretch;

CircleCollider2D circle = GetComponent<Collider2D>() as CircleCollider2D;

circleRadius = circle.radius;

}

void Update () {

if (clickedOn)

Dragging ();

if (spring != null) {

if (!GetComponent<Rigidbody2D>().isKinematic && prevVelocity.sqrMagnitude > GetComponent<Rigidbody2D>().velocity.sqrMagnitude) {

Destroy (spring);

GetComponent<Rigidbody2D>().velocity = prevVelocity;

}

if (!clickedOn)

prevVelocity = GetComponent<Rigidbody2D>().velocity;

LineRendererUpdate ();

} else {

catapultLineFront.enabled = false;

catapultLineBack.enabled = false;

}

}

void LineRendererSetup () {

catapultLineFront.SetPosition(0, catapultLineFront.transform.position);

catapultLineBack.SetPosition(0, catapultLineBack.transform.position);

catapultLineFront.sortingLayerName = "Foreground";

catapultLineBack.sortingLayerName = "Foreground";

catapultLineFront.sortingOrder = 3;

catapultLineBack.sortingOrder = 1;

}

void OnMouseDown () {

spring.enabled = false;

clickedOn = true;

}

void OnMouseUp () {

spring.enabled = true;

GetComponent<Rigidbody2D>().isKinematic = false;

clickedOn = false;

}

void Dragging () {

Vector3 mouseWorldPoint = Camera.main.ScreenToWorldPoint(Input.mousePosition);

Vector2 catapultToMouse = mouseWorldPoint - catapult.position;

if (catapultToMouse.sqrMagnitude > maxStretchSqr) {

rayToMouse.direction = catapultToMouse;

mouseWorldPoint = rayToMouse.GetPoint(maxStretch);

}

mouseWorldPoint.z = 0f;

transform.position = mouseWorldPoint;

}

void LineRendererUpdate () {

Vector2 catapultToProjectile = transform.position - catapultLineFront.transform.position;

leftCatapultToProjectile.direction = catapultToProjectile;

Vector3 holdPoint = leftCatapultToProjectile.GetPoint(catapultToProjectile.magnitude + circleRadius);

catapultLineFront.SetPosition(1, holdPoint);

catapultLineBack.SetPosition(1, holdPoint);

}

}