using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class GyroDuck : MonoBehaviour

{

 public bool gyroEnabled;

 private Gyroscope gyro;

 private Collider2D C2D;

 private SpriteRenderer SR;

 public bool canDuck;

 public bool Rotated;

 public bool ResetRot;

 public bool FirstRot;

 public bool InverseRot;

 public float xRotation;

 public float yRotation;

 public float zRotation;

 public float newXRotation;

 public float xRot1;

 public float xRot2;

 public float AltxRotation;

 public bool isRefRotX;

 public bool doneFirstRot;

 public int firstRotCounter;

 public int duckingCounter;

 public int testCounter;

 void Awake ()

 {

 FirstRot = true;

 }

 void Start ()

 {

 gyroEnabled = EnableGyro ();

 C2D = GetComponent<Collider2D> ();

 SR = GetComponent<SpriteRenderer> ();

 canDuck = false;

 ResetRot = false;

 FirstRot = true;

 Rotated = false;

 }

 public bool EnableGyro()

 {

 if (SystemInfo.supportsGyroscope)

 {

 gyro = (Input.gyro);

 gyro.enabled = true;

 return true;

 }

 return false;

 }

 void Update ()

 {

 gyroEnabled = EnableGyro ();

 if (gyroEnabled == true)

 {

 if (FirstRot == true && firstRotCounter < 1)

 {

 //StartCoroutine ("FirstTimer");

 isRefRotX = true;

 StopCoroutine ("FirstTimer");

 StartCoroutine ("FirstTimer");

 firstRotCounter++;

 }

 CheckXRot ();

 SetXRot ();

 if (ResetRot == true)

 {

 xRotation = Input.gyro.attitude.x;

 if (xRotation >= 0.6f)

 {

 xRotation = 0.4f;

 }

 ResetRot = false;

 }

 //xRotationMax = xRotation + 0.4f;

 if (isRefRotX == false)

 {

 newXRotation = Input.gyro.attitude.x;

 canDuck = true;

 doneFirstRot = true;

 }

 if (doneFirstRot == true)

 {

 newXRotation = Input.gyro.attitude.x;

 }

 if (InverseRot == true)

 {

 xRot1 = 0.99f - xRotation;

 xRot2 = 0.99f - newXRotation;

 AltxRotation = xRot1 + xRot2;

 }

 if (((newXRotation - xRotation) >= 0.2f) || (AltxRotation >= 0.2f))

 {

 Rotated = true;

 xRot1 = 0f;

 xRot2 = 0f;

 AltxRotation = 0f;

 InverseRot = false;

 }

 if (duckingCounter < 1 && canDuck == true && gyroEnabled == true && (Rotated == true || Input.GetKeyDown ("space")))

 {

 duckingCounter++;

 Input.gyro.enabled = false;

 Ducking ();

 }

 }

 }

 void SetXRot ()

 {

 StopCoroutine ("Timer");

 StartCoroutine ("Timer");

 }

 void Ducking ()

 {

 //StopCoroutine ("DuckCooldown");

 //StartCoroutine ("DuckCooldown");

 DuckCooldown();

 }

 void CheckXRot ()

 {

 if (Input.gyro.attitude.x >= 0.99f)

 {

 InverseRot = true;

 }

 }

 public IEnumerator FirstTimer ()

 {

 yield return new WaitForSecondsRealtime (0.5f);

 if (isRefRotX == true)

 {

 xRotation = Input.gyro.attitude.x;

 if (xRotation >= 0.6f)

 {

 xRotation = 0.4f;

 }

 isRefRotX = false;

 }

 }

 public IEnumerator Timer()

 {

 if (FirstRot == false)

 {

 yield return new WaitForSecondsRealtime (1f);

 ResetRot = true;

 yield return ResetRot;

 }

 if (doneFirstRot == true)

 {

 FirstRot = false;

 }

 yield return FirstRot;

 //xRotation = Input.gyro.attitude.x;

 //yield return xRotation;

 }

 /\*public IEnumerator DuckCooldown()

 {

 GameControl.instance.ducked = true;

 canDuck = false;

 C2D.enabled = !C2D.enabled;

 //SR.color = Color.black;

 PlayerMovement.instance.p\_Animator.SetTrigger("Duck");

 PlayerMovement.instance.fastSplash.SetActive (false);

 GameControl.instance.DuckScored ();

 yield return new WaitForSecondsRealtime (2f);

 C2D.enabled = true;

 GameControl.instance.ducked = false;

 PlayerMovement.instance.fastSplash.SetActive (true);

 //SR.color = Color.white;

 canDuck = true;

 ResetRot = true;

 Rotated = false;

 Input.gyro.enabled = true;

 yield return canDuck;

 yield return C2D;

 yield return ResetRot;

 yield return Rotated;

 }\*/

 public void DuckCooldown()

 {

 testCounter++;

 GameControl.instance.ducked = true;

 canDuck = false;

 GameControl.instance.DuckScored();

 C2D.enabled = !C2D.enabled;

 //SR.color = Color.black;

 PlayerMovement.instance.p\_Animator.SetTrigger("Duck");

 PlayerMovement.instance.fastSplash.SetActive (false);

 }

 void Duck2()

 {

 GameControl.instance.onePointSFX.SetActive (false);

 GameControl.instance.minusPointsSFX.SetActive (false);

 GameControl.instance.failureSFX.SetActive (false);

 C2D.enabled = true;

 GameControl.instance.ducked = false;

 PlayerMovement.instance.fastSplash.SetActive (true);

 //SR.color = Color.white;

 canDuck = true;

 ResetRot = true;

 Rotated = false;

 Input.gyro.enabled = true;

 //yield return canDuck;

 //yield return C2D;

 //yield return ResetRot;

 //yield return Rotated;

 }

 void resetDuckingCounter()

 {

 duckingCounter = 0;

 }

}