using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class DetectPlayer : MonoBehaviour {

public Animator attackAnimator;

public SpriteRenderer ASR;

public GameObject clapSFX;

public bool collided;

public float xPos;

public int parentNumber;

public int flipCount = 0;

public int animCount;

void Start ()

{

animCount = 0;

attackAnimator = gameObject.GetComponent<Animator> ();

ASR = gameObject.GetComponent<SpriteRenderer> ();

attackAnimator.ResetTrigger ("Fail");

attackAnimator.ResetTrigger ("Success");

parentNumber = gameObject.GetComponentInParent<Enemy> ().number;

xPos = transform.localPosition.x;

}

void Update ()

{

if (gameObject.GetComponentInParent<Enemy>().canFlip == true && flipCount < 1)

{

//xPos = transform.localPosition.x;

transform.localPosition = new Vector2(-xPos, transform.localPosition.y);

flipCount++;

}

else if (gameObject.GetComponentInParent<Enemy>().SR.flipX == false)

{

transform.localPosition = new Vector2(xPos, transform.localPosition.y);

}

if (gameObject.GetComponentInParent<Enemy>().attacking == true && parentNumber == gameObject.GetComponentInParent<Enemy>().who && animCount < 1 && GetComponentInParent<Enemy>().canHands == true)

{

animCount++;

//StartCoroutine ("grabWait");

if (collided == true)

{

GameControl.instance.player.GetComponent<GyroDuck> ().canDuck = false;

GameControl.instance.player.GetComponent<GyroDuck> ().duckingCounter = 1;

attackAnimator.ResetTrigger ("Fail");

attackAnimator.SetTrigger ("Success");

//GameControl.instance.DuckDied ();

}

else

{

attackAnimator.ResetTrigger ("Success");

attackAnimator.SetTrigger ("Fail");

}

}

}

/\*public IEnumerator grabWait()

{

//wait for length of pre-attack animation

yield return new WaitForSecondsRealtime (0.333f);

if (collided == true)

{

gameObject.GetComponent<Animator> ().Play("Success");

//attackAnimator.ResetTrigger ("Fail");

//attackAnimator.SetTrigger ("Success");

yield return new WaitForSecondsRealtime (0.5f);

}

else

{

gameObject.GetComponent<Animator> ().Play("Fail");

//attackAnimator.ResetTrigger ("Success");

//attackAnimator.SetTrigger ("Fail");

yield return new WaitForSecondsRealtime (0.5f);

}

}\*/

void OnTriggerEnter2D(Collider2D other)

{

if (other.gameObject.tag == "Player")

{

collided = true;

}

}

void OnTriggerExit2D(Collider2D other)

{

if (other.gameObject.tag == "Player")

{

collided = false;

}

}

void attackEnded()

{

GameControl.instance.player.GetComponent<SpriteRenderer>().color = Color.white;

gameObject.GetComponentInParent<Enemy>().idle = true;

gameObject.GetComponentInParent<Enemy>().attacking = false;

gameObject.GetComponentInParent<Enemy> ().canHands = false;

gameObject.GetComponentInParent<ScrollingObject> ().useAnticipationSpeed = false;

EnemyPool.instance.slowDown = false;

attackAnimator.ResetTrigger ("Fail");

attackAnimator.ResetTrigger ("Success");

clapSFX.SetActive (false);

}

void callDuckDied()

{

GameControl.instance.DuckDied ();

}

void callClapSFX()

{

if (GlobalControl.Instance.SFXOn == true)

{

clapSFX.SetActive (true);

}

}

}