﻿using UnityEngine;

using System.Collections;

//This script is the base script for both Player and Enemy

//Ensure that the game object this is on has a rigidbody and animator

[RequireComponent(typeof(Rigidbody2D), typeof(Animator))]

public class Spaceship : MonoBehaviour

{

public float speed; //Ship's speed

public float shotDelay; //Delay between shots

public GameObject bullet; //The prefab of this ship's bullet

public bool canShoot; //Can this ship fire?

public GameObject explosion; //The prefab of this ship's explosion

protected Transform[] shotPositions; //Fire points on the ship

protected Animator animator; //Reference to the ship's animator component

void Awake ()

{

//Get the fire points for future reference (this is for efficiency)

shotPositions = new Transform[transform.childCount];

for (int i = 0; i < transform.childCount; i++)

shotPositions[i] = transform.GetChild (i);

//Get a reference to the animator component

animator = GetComponent<Animator> ();

}

protected virtual void OnEnable()

{

//If the game is playing and the ship can shoot...

if (canShoot && Manager.current.IsPlaying())

//...Start it shooting

StartCoroutine ("Shoot");

}

void OnDisable()

{

//If the ship was able to shoot and it became disabled...

if(canShoot)

//...Stop shooting

StopCoroutine ("Shoot");

}

protected void Explode ()

{

//Get a pooled explosion object

GameObject obj = ObjectPool.current.GetObject(explosion);

//Set its position and rotation

obj.transform.position = transform.position;

obj.transform.rotation = transform.rotation;

//Activate it

obj.SetActive (true);

}

//Coroutine

IEnumerator Shoot ()

{

//Loop indefinitely

while(true)

{

//If there is an acompanying audio, play it

if (GetComponent<AudioSource>())

GetComponent<AudioSource>().Play ();

//Loop through the fire points

for(int i = 0; i < shotPositions.Length; i++)

{

//Get a pooled bullet

GameObject obj = ObjectPool.current.GetObject(bullet);

//Set its position and rotation

obj.transform.position = shotPositions[i].position;

obj.transform.rotation = shotPositions[i].rotation;

//Activate it

obj.SetActive(true);

}

//Wait for it to be time to fire another shot

yield return new WaitForSeconds(shotDelay);

}

}

}