﻿using UnityEngine;

using System.Collections;

//This script manages the player object

public class Player : Spaceship

{

void Update ()

{

//Get our raw inputs

float x = Input.GetAxisRaw ("Horizontal");

float y = Input.GetAxisRaw ("Vertical");

//Normalize the inputs

Vector2 direction = new Vector2 (x, y).normalized;

//Move the player

Move (direction);

}

void Move (Vector2 direction)

{

//Find the screen limits to the player's movement

Vector2 min = Camera.main.ViewportToWorldPoint(new Vector2(0, 0));

Vector2 max = Camera.main.ViewportToWorldPoint(new Vector2(1, 1));

//Get the player's current position

Vector2 pos = transform.position;

//Calculate the proposed position

pos += direction \* speed \* Time.deltaTime;

//Ensure that the proposed position isn't outside of the limits

pos.x = Mathf.Clamp (pos.x, min.x, max.x);

pos.y = Mathf.Clamp (pos.y, min.y, max.y);

//Update the player's position

transform.position = pos;

}

void OnTriggerEnter2D (Collider2D c)

{

//Get the layer of the collided object

string layerName = LayerMask.LayerToName(c.gameObject.layer);

//If the player hit an enemy bullet or ship...

if( layerName == "Bullet (Enemy)" || layerName == "Enemy")

{

//...and the object was a bullet...

if(layerName == "Bullet (Enemy)" )

//...return the bullet to the pool...

ObjectPool.current.PoolObject(c.gameObject) ;

//...otherwise...

else

//...deactivate the enemy ship

c.gameObject.SetActive(false);

//Tell the manager that we crashed

Manager.current.GameOver();

//Trigger an explosion

Explode();

//Deactivate the player

gameObject.SetActive(false);

}

}

}