

Sistema de misiles teledirigidos para videojuegos 2d en Unity.

En esta documentación se explicará cómo funciona el código interno del juego, al igual de cómo se pueden manipular sus configuraciones dentro de la interfaz de Unity.

Explicación del código (sistema de misiles)

Este código hace funcionar al misil y que este siga al jugador.

Se puede modificar desde el propio código o desde la interfaz de unity.

Este código hace que una vez el misil toque al jugador desaparezca el misil.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(Rigidbody2D))]
public class sistema_de_misil : MonoBehaviour
{
    public Transform target;
    private Rigidbody2D rb;
    public float speed = 5f;
    public float rotateSpeed = 200f;
    public GameObject explosionEffect;

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    void Update()
    {
    }

    void FixedUpdate()
    {
        Vector2 direction = (Vector2)target.position - rb.position;
        direction.Normalize();
        float rotateAmount = Vector3.Cross(direction, transform.up).z;
        rb.angularVelocity = -rotateAmount * rotateSpeed;
        rb.velocity = transform.up * speed;
    }

    void OnTriggerEnter2D()
    {
        Instantiate(explosionEffect, transform.position, transform.rotation);
        Destroy(gameObject);
    }
}
```

Movimiento del personaje

Este código hace que el jugador pueda moverse, al igual que se puede modificar la velocidad del jugador.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    public Rigidbody2D rb;
    public float speed;
    Vector2 movement;

    // Update is called once per frame
    void Update()
    {
        movement.x = Input.GetAxisRaw("Horizontal");
        movement.y = Input.GetAxisRaw("Vertical");
    }

    private void FixedUpdate()
    {
        rb.MovePosition(rb.position + movement * speed * Time.deltaTime);
    }
}
```

Lanzador de misiles

Este código se encarga de hacer que un objeto lance los misiles periódicamente.

Los misiles salen cada cierto tiempo y se puede modificar el tiempo de reaparición de los misiles.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

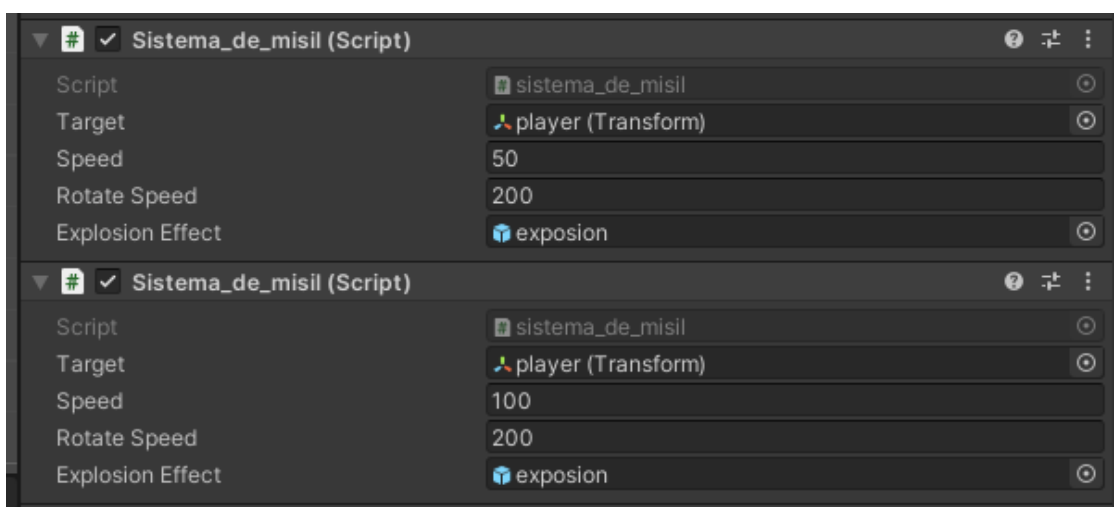
public class lanzador : MonoBehaviour
{
    public Transform punto_intancia;
    public GameObject misil;
    private float tiempo;

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        tiempo += Time.deltaTime;
        if(tiempo >=5)
        {
            Instantiate(misil, punto_intancia.position, Quaternion.identity);
            tiempo = 0;
        }
    }
}
```

Modificación de los códigos desde la interfaz de unity.

Sistema de misiles.



Se explicará cada parte de su configuración.

Target.

aquí se coloca lo que el misil va a perseguir, se puede poner cualquier cosa que el usuario quiera, en este caso, el jugador.

Speed.

se modifica la velocidad con la que se moverá el misil.

Rotate Speed.

se modifica la velocidad con la que el misil dará vueltas por si falla en tocar al jugador para intentar tocarlo de nuevo.

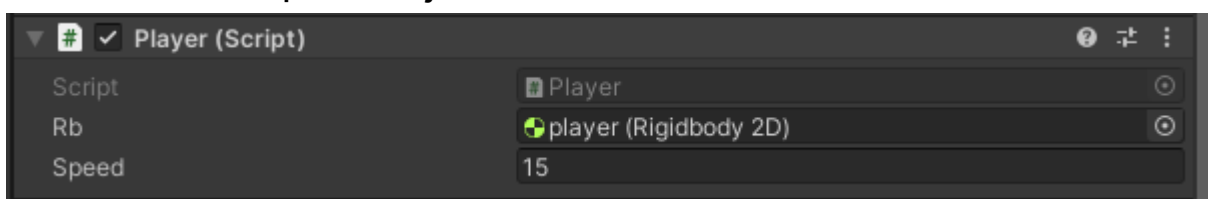
Explosion Effect.

Se coloca en esta parte un efecto especial para que cuando el misil toque al jugador este explote.

IMPORTANTE.

Si no te diste cuenta, el misil funciona con dos códigos iguales, con la diferencia que uno tiene la mitad de velocidad del otro, si se quiere modificar la velocidad se tiene que modificar en ambos y de manera en la que uno tenga la mitad del otro.

Movimiento del personaje.

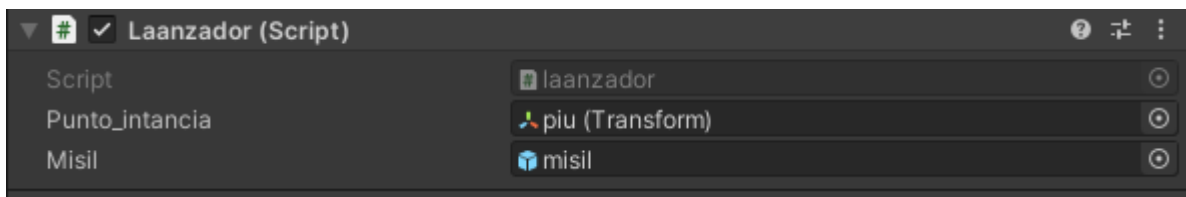


Rb. se coloca un colisionador para que el jugador tenga físicas.

Speed.

Se modifica la velocidad con la que se moverá el jugador.

Lanzador de misiles.



Punto instancia.

se coloca el objeto de donde saldrán los misiles.

Misil.

se coloca el proyectil que lanzará el lanzador de misiles.