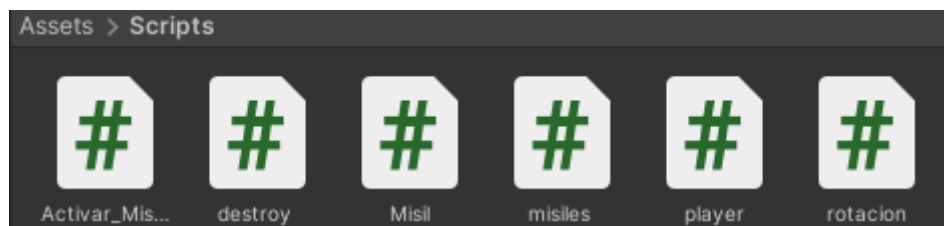


# Sistema de Misiles Teledirigidos

Creador: Oskar Dimas Carbajal

Este programa Esta echo para ser fácil de implementar y abierto para cualquier modificación posible

Estos códigos serán los encargados de hacer todo eso posible



Player se encargará de hacer que el objetivo del misil pueda ir cualquier dirección para poder esquivar los misiles lanzados siendo W,A,S,D los controles para poder moverse, además de tener las sus variables (velocidad y velocidad de rotación) en público por si gustas modificar los valores de ambos.

```
public class player : MonoBehaviour
{
    //variables

    public float speed = 0.0f; //velocidad a la que se va a mover el personaje

    public float rotationSpeed = 0.0f; //velocidad a la que va a rotar el personaje

    //ambos son publicos para poder modificarlos desde la consola
}
```

Activar Misil se encargará de que cada que se puse la tecla correspondiente ponga al objeto que se dese (en este caso el misil) como activo u inactivo, siendo la tecla para activarlo la R y para desactivarlo la E.

```
if (Input.GetKeyDown(KeyCode.E))//cada que se precione la tecla e se desactivara el objeto
{
    objectToActiveAndDesactivate.SetActive(false); // se pondra en falso para desactivarlo
}
if (Input.GetKeyDown(KeyCode.R))//cada que se precione la letra R se activara el ojeeto
{
    objectToActiveAndDesactivate.SetActive(true); // se pondra en verdadero el objeto piniendolo a funcionar
}
```

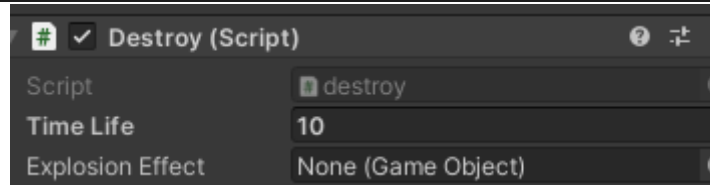
Rotación será el encargado de manejar la torreta, esta lo que hará será siempre apuntar al jugador sin importa a donde se mueva, además de señalar la posición del jugador con un láser de color rojo (personalizable desde el código más en específico en la imagen se especifica dónde).

```
void Update()
{
    Vector3 targetOrientation = _objetivo.position - transform.position;
    Debug.DrawRay(transform.position, targetOrientation, Color.red); //esta linea se encarga de generar un "laser" que apunte al ojeitivo todo el tiempo
}
```

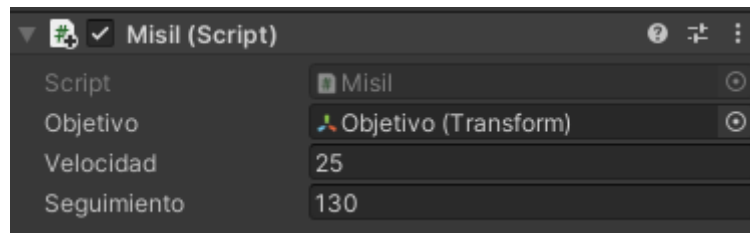


Destroy será el encargado de manejar el tiempo que los misiles estén en pantalla, así como el que cuando toquen a su objetivo se destruyan, además de contar con una “rama” que, al agregar algún efecto de explosión, se ejecute dicha animación, así como también se encarga de eliminar el objeto para coordinar ambas cosas.

```
//proceso encargado de activar la explosion al momento de colicionar con el objetivo
@ MonoBehaviour
private void OnTriggerEnter(Collider other)
{
    if (other.tag == "Player") //cuando se cumpla con la condicional de que el misil colicione con el objetivo entonces se activara este proceso
    {
        Instantiate(explosionEffect, transform.position, Quaternion.identity); //cuando se llegue a impactar con el objetivo se activara la explosion y se eliminara el objetivo
        Destroy(gameObject);
    }
}
```



Misil se encargará de hacer que el misil persiga al jugador sin importar a donde se dirija, corrigiendo su curso cada que el jugador cambia de posición, dejando sus variables en público (velocidad y rango de seguimiento (entre más grande sea el número del rango más cerca estará del jugador)) por si se gusta cambiar dichos valores, así como dejando un espacio para colocar al objeto u objetivo que el misil seguirá.



Misiles este script será el encargado de que el prefab misil sea clonado durante cierto tiempo y deteniendo el proceso cuando se cumpla dicho rango además de dejar un campo para colocar el objeto a duplicar, así como el tiempo entre disparo y el contador de misiles.

