

# Scripts del juego.

## Sistema de movimiento del player.

Este script fue creado con el objetivo de hacer que el Player pueda moverse dentro del juego y de esa manera activar los sistemas de dialogo.

1. Primero se agregaron las referencias para poder programarlo.}

```
//Movimiento
public float moveSpeed;
//Salto
private bool canDoubleJump;
public float jumpForce;
//Componentes
public Rigidbody2D rgb;
//Animaciones+
private Animator anim;
private SpriteRenderer theSR;
//Piso
private bool isGrounded;
public Transform groundCheckPoint;
public LayerMask whatIsGround;
```

2. Aquí le decimos que el jugador va a poder moverse en los planos (x, y) a la velocidad deseada en el unity.

Le decimos que detecte el suelo todo objeto que tenga la etiqueta "isGrounded" para así hacer que solo pueda saltar cuando lo este tocando y no en cualquier área.

También se dio la opción de tener un doble salto, pero no se puede reactivar hasta que vuelva a tocar suelo.

Y se activa una animación en el jugador depende la dirección en la cual se mueva.

```
void Update()
{
    rgb.velocity = new Vector2(moveSpeed * Input.GetAxis("Horizontal"), rgb.velocity.y);

    isGrounded = Physics2D.OverlapCircle(groundCheckPoint.position, 0.2f, whatIsGround);

    if(isGrounded)
    {
        canDoubleJump = true;
    }

    if(Input.GetButtonDown("Jump"))
    {
        if (isGrounded)
        {
            rgb.velocity = new Vector2(rgb.velocity.x, jumpForce);
        }
        else
        {
            if(canDoubleJump)
            {
                rgb.velocity = new Vector2(rgb.velocity.x, jumpForce);
                canDoubleJump = false;
            }
        }
    }

    if(rgb.velocity.x < 0)
    {
        theSR.flipX = true;
    }
    else
    {
        if(rgb.velocity.x > 0)
        {

```

```

        theSR.flipX = false;
    }
    anim.SetFloat("moveSpeed", Mathf.Abs(rgb.velocity.x));
    anim.SetBool("isGrounded", isGrounded);
}
}
}

```

## Fondo movimiento.

Se creo un script para que el fondo se pueda mover junto con el jugador y el fondo tenga movimiento dando mas vida al juego

1. Se agregaron las variables para hacer que este script funcione.

```

public Vector2 velocidadMovimiento; // para el movimiento del fondo.
private Vector2 offset;
private Material material; //Referencia al material.
private Rigidbody2D jugadorRB;//referencia al jugador.

```

2. Se agrego un material a los fondos para que este fuera detectado por es script.

```

private void Awake()
{
    material = GetComponent<SpriteRenderer>().material; //El material es el mismo que tiene el sprite renderer.
    jugadorRB = GameObject.FindGameObjectWithTag("Player").GetComponent<Rigidbody2D>();
}

```

3. Se agrego la velocidad de movimiento que va atener el fondo al momento de moverse.

```

private void Update()
{
    offset = (jugadorRB.velocity.x * 0.1f) * velocidadMovimiento * Time.deltaTime;//el offset es igual a la velocidad de time delta tim
    material.mainTextureOffset += offset;
}

```

## Lluvia.

Se agrego un script a un objeto vacío el cual va a generar objetos en este caso rocas de manera aleatoria las cuales van caer en un espacio específico.

1. se colocaron las variables para que este script funcionara.

```

public GameObject[] Rocas;
public float SegundosSpawn = 0.5f;
public float MinPiedra;
public float MaxPiedra;

```

2. Colocamos una rutina para que vayan apareciendo rocas en esa área específica.

```

void Start()
{
    StartCoroutine(RockSpawn());
}

```

3. Aquí le decimos que las rocas van a salir de manera aleatoria en toda la zona donde esta colocado el objeto vacío y cuanto tiempo va a tardar para salir una roca distinta.  
Y después de cierto tiempo la roca se va a destruir para no saturar el juego.

```
IEnumerator RockSpawn()
{
    while(true)
    {
        var wanted = Random.Range(MinPiedra, MaxPiedra);
        var position = new Vector3(wanted, transform.position.y);
        GameObject gameObject = Instantiate(Rocas[Random.Range(0, Rocas.Length)], position, Quaternion.identity);
        yield return new WaitForSeconds(SegundosSpawn);
        Destroy(gameObject, 5f);
    }
}
```

## Menú principal y rocas.

Aquí se generaron 2 scripts que van de la mano para hacer mas interactivo el juego.

Cuando una roca toca al jugador este automáticamente es llevado al menú de inicio.

1. Se hizo el script de rocas, la cual al tocar al jugador este va a ser mandado a la primera escena el cual es un menú principal creado para este juego.

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        SceneManager.LoadScene(0); //Cada que una de las rocas toque al player, se va a reiniciar el juego.
    }
}
```

2. El menú principal va a contar con dos botones uno dice jugar y el otro dice salir.  
Al presionar jugar va a llevar a la segunda escena la cual es el juego y se va a jugar con normalidad.  
Al presionar salir, el juego se va a cerrar.

```
public void jugar()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1); //Para cambiar de escena al juego.
}

public void salir()
{
    Application.Quit(); //Para cerrar el juego.
}
```