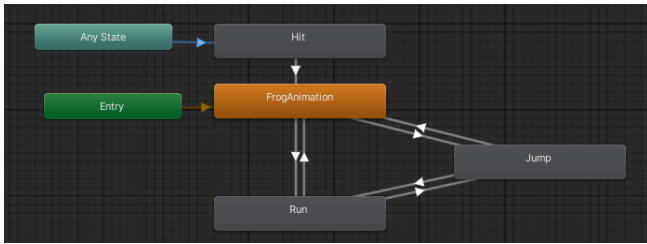


Lección 8:

Ahora animaremos nuestro player para hacer una animación Hit al momento de chocar con los obstáculos de nuestro juego “Spike y Head Spike” lo primero es incluir nuestra animación en nuestro árbol de animaciones.



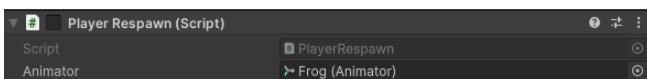
Teniendo esto listo ahora modificamos los script Player Respawn y los de cada objeto enemigo para PlayerRespawn utilizamos un método que activara la animación y volvera a cargar la escena para aparecer al principio del nivel y en los Script de los enemigos ahora en lugar de destruir el objeto Player lo que hara sera invocar el metodo que acabamos de crear en PlayerRespawn

```
public void PlayerDamaged()
{
    animator.Play("Hit");
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```

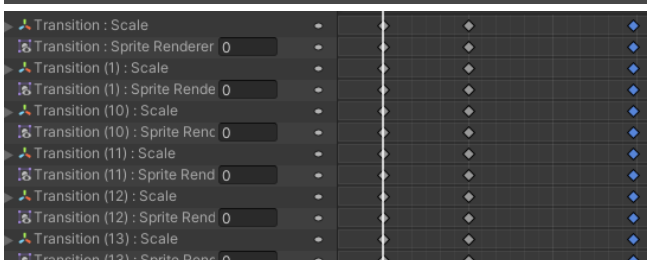
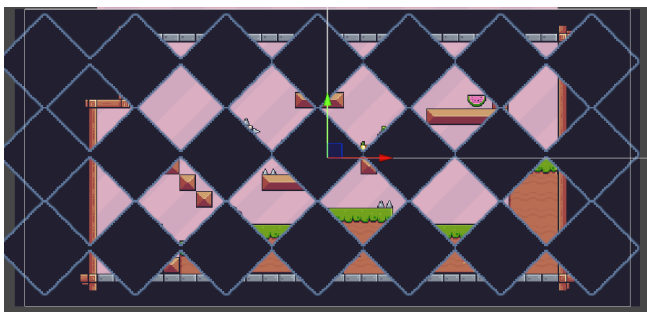
```
public class DamageObject : MonoBehaviour
{
    // Mensaje de Unity | 0 referencias
    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.transform.CompareTag("Player"))
        {
            Debug.Log("Player Damaged");
            collision.transform.GetComponent<PlayerRespawn>().PlayerDamaged();
        }
    }
}
```

```
public Animator animator;
```

Ahora faltara señalar en el inspecto PlayerRespawn el animator para que pueda encontrar la animacion Hit



para hacer un poco mas dinamico el cambio entre escenas agregaremos una animación de rombos, elegimos la animación de nombre transition y colocamos de la siguiente forma y animamos scale y la transparencia para un efecto fade y llamamos a esta animación en El Script fruitManager para que suceda cada fin de nivel.



```
public GameObject transition;

// Mensaje de Unity | 0 referencias
private void Update()
{
    AllFruitsCollected();
}

1 referencia
public void AllFruitsCollected()
{
    if (transform.childCount == 0)
    {
        Debug.Log("No quedan frutas, Victoria!");
        levelCleared.gameObject.SetActive(true);
        transition.SetActive(true);
        Invoke("ChangeScene", 1);
    }
}
```