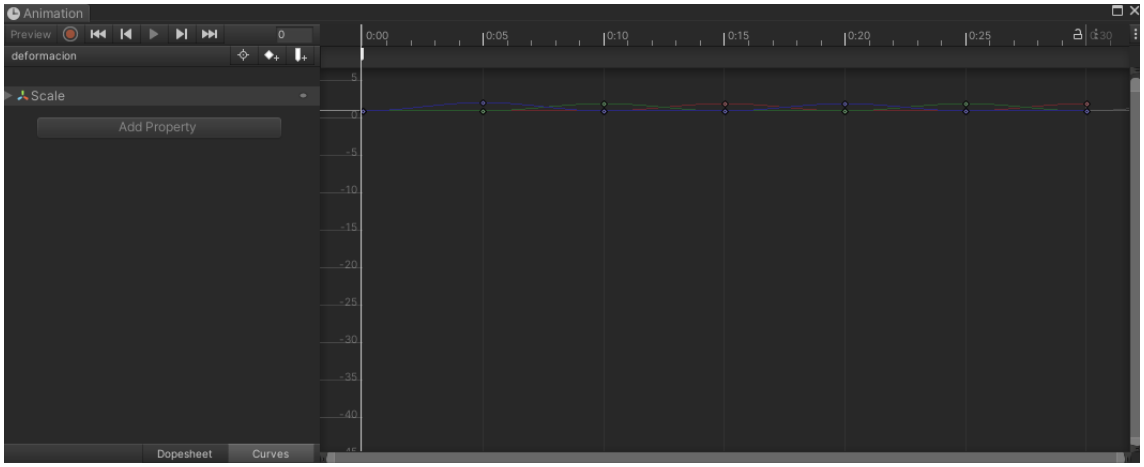
Examen PDMD Unity.

Primera Animación:

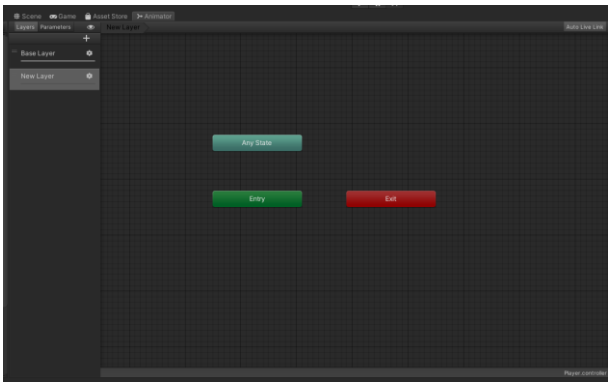
En esta primera animación lo que se nos pide es que hagamos una animación donde una bola se deforme en el momento en el que el jugador salte, para esto vamos a seguir los siguientes pasos:

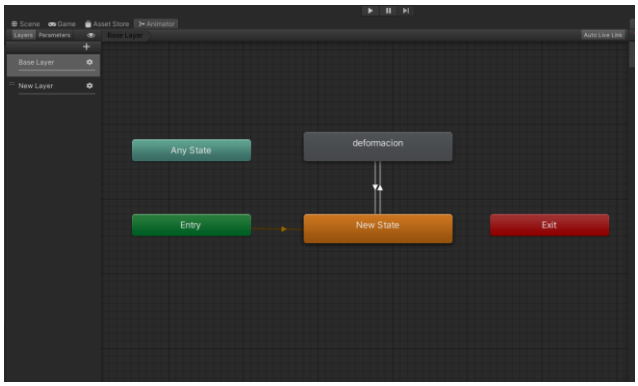
Lo primero que tenemos que hacer para crear una animación en unity es seleccionar el objeto que queremos animar en nuestro caso seleccionaremos la bola.

Una vez seleccionada le daremos a Window>Animation, una vez tengamos abierta la ventana de animación lo que debemos es darle al botón de grabar animación y hacer modificaciones el objeto a nuestro gusto desde el estado actual hacia el nuevo estado, es importante tener en cuenta la barra de tiempo ya que cada vez que modifiquemos el objeto debemos modificar la barra ya que sino no se vera afectado el cambio.



Una vez tenemos la animación debemos crear un Animator, este se puede crear desde Window>Animation>Animator una vez dentro arrastramos nuestra animación y esto vinculara la nueva animación de deformación al estado que representa la deformación de la bola.





Esta primera imagen es el Animator default y la segunda imagen es como quedaría con la animación integrada, en este caso lo que hice fue crear una animación soporte la cual esta

estática donde no hay ningún cambio para hacer una transición mediante la creación de un trigger, esas dos flechas blancas son las transiciones las cuales están vinculadas con el trigger que posteriormente configuraremos en nuestro código. La función de estas flechas es delimitar el momento en el que se va a llevar acabo esta animación, en el momento que el jugador salte va a pasar de la primera animación a la segunda que seria cuando la bola se deforma y cuando acabe esta animación volverá a la primera.

El siguiente paso que tenemos que realizar es la configuración del código para activar la animación en este caso la animación se debe activar cuando la bola salte.

Código:

```
using UnityEngine;

public class BallController : MonoBehaviour
{
    private Rigidbody rb;
    private Animator anim;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space)) // Detecta si se presiona
la tecla de salto (barra espaciadora)
        {
            Jump(); // Llama a la función para hacer que la bola salte
        }
    }

    void Jump()
    {
        // Aquí pones el código para hacer que la bola salte
        rb.AddForce(Vector3.up * 10f, ForceMode.Impulse);

        // Activa el trigger en el Animator para iniciar la transición
hacia la animación de deformación
        if (anim != null)
        {
            anim.SetTrigger("Salto"); // Activa el trigger "Jump" en
el Animator
        }
    }
}
```

Enemigo:

En cuanto al enemigo tenemos que tener en cuenta que debemos tener una superficie lo suficientemente grande para que este se desplace, esto lo podemos hacer con un simple plano y agrandándolo según nos interese.

Para hacer realizar la persecución hacia el jugador utilice el siguiente código:

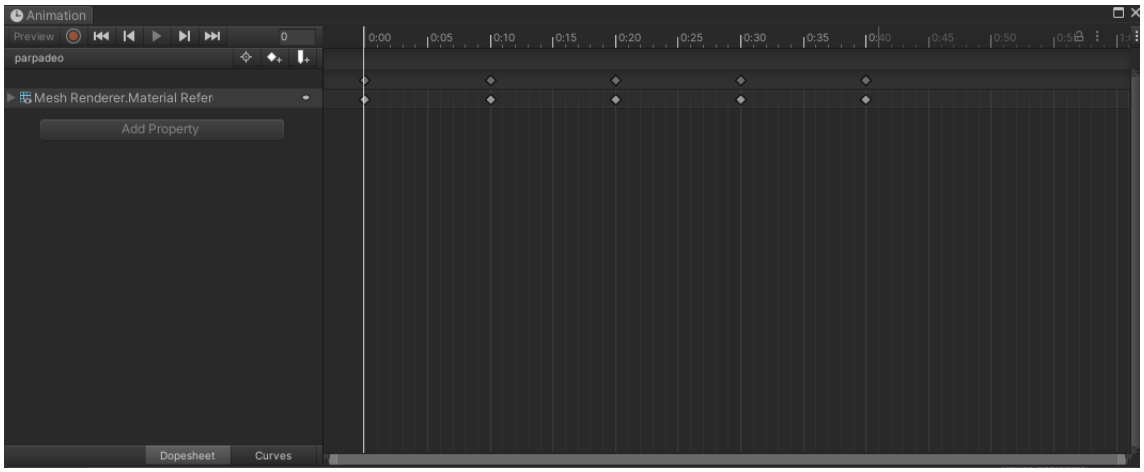
```
// Buscar el objeto con la etiqueta "Player" y obtener su transform
player = GameObject.FindGameObjectWithTag("Player").transform;

// Calcular la dirección hacia la posición actual del jugador
Vector3 targetDirection = player.position - transform.position;

// Normalizar la dirección para obtener una velocidad constante
targetDirection.Normalize();

// Mover el enemigo hacia el jugador con la velocidad especificada
transform.Translate(targetDirection * speed * Time.deltaTime);
```

Para hacer que la animación donde la bola parpade a color rojo cada vez que el enemigo este cerca lo primero que necesitamos es crear la animación.

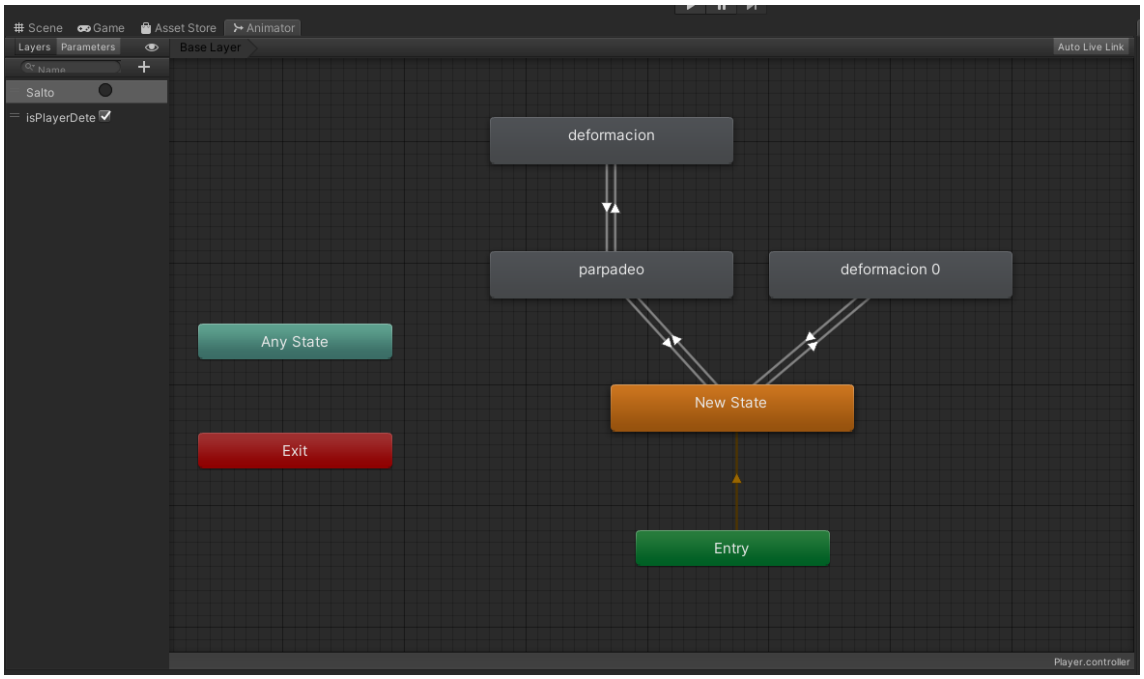


Una vez tenemos esto vamos a crear un nuevo parámetro en el apartado de Animator, este parámetro será un boolean el cual se pondrá en true en el momento que enemigo este cerca y se pondrá false en el momento que el enemigo este fuera del rango.

Una vez tenemos el parámetro creado implementamos un simple if en el código que determina el estado del boolean.

```
if (Vector3.Distance(transform.position, player.position) <=
detectionRadius)
{
    isPlayerDetected = true;
}
else
{
    isPlayerDetected = false;
}
```

Una vez tenemos todo integrado en nuestro código es necesario introducir la animación en el Animator y crear las transiciones.



En el Animator he creado dos ramificaciones principales donde en el caso de que el enemigo este cerca nuestra el jugador empezara a parpadear y en el caso de que saltemos este se deformara, en el caso de que el enemigo no este cerca nuestra se utilizara la segunda rama donde solo esta la animación de deformación.

Las flechas que hay entre las diferentes animaciones son transiciones estas son necesarias para determinar en que momento va a suceder cada animación. Dentro de estas flechas es donde podemos situar las diferentes condiciones y parámetros creados anteriormente.