

Animação no Código

Agora temos a nossa árvore de animações feitas e temos que em algum lugar chamar esse parâmetro **"Movendo"** para fazer a nossa animação trocar, onde será esse lugar? Não é o código que controla todo o nosso jogo? Então é lá que essa troca será feita.

Lembre-se que o código não sabe que você tem um *Animator* no seu objeto então temos que dizer pro Unity que queremos utilizar o *Componente (Component)* para isto utilizamos o código `GetComponent<Animator>()` que recupera do nosso objeto um *Component* do tipo *Animator*.

Tendo o nosso *Componente* agora podemos entrar nele e passar um valor para o parâmetro do tipo `bool` para isso utilizamos `GetComponent<Animator>().SetBool("Movendo", true);` por exemplo para mudar o valor de **"Movendo"** para **Verdadeiro**, lembre-se de colocar essa linha no método `Update`.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ControlaJogador : MonoBehaviour {
6
7      public float Velocidade = 10;
8
9      // Update is called once per frame
10     void Update () {
11         float eixoX = Input.GetAxis("Horizontal");
12         float eixoZ = Input.GetAxis("Vertical");
13
14         Vector3 direcao = new Vector3(eixoX, 0, eixoZ);
15
16         transform.Translate(direcao * Velocidade * Time.deltaTime);
17     }
18 }
19
```

Agora o personagem jogador sempre fica se correndo como podemos fazer ele parar quando soltamos as teclas? Para isso é utilizado a estrutura de controle conhecida como `if` onde temos que entre parênteses colocar o que queremos testar no nosso código e abrir/fechar chave para as instruções que irão acontecer quando essa condição for satisfeita, da seguinte forma:

```
if (o que queremos testar)
{
}
}
```

Neste caso o que será testado é se as teclas estão sendo apertadas e para isso temos a variável `direcao` que guarda os valores das teclas que utilizamos para movimentar nosso jogador, logo vamos testar se os valores não estão zerados pois se estiverem nenhuma tecla foi apertada.

Colocando as linhas abaixo no `Update` abaixo dos códigos já existentes estamos verificando se a nossa variável de `direcao` não (símbolo `!=`) é igual (símbolo `=`) a zero no eixo X, Y e Z. Neste caso alguma tecla está sendo pressionada então devemos fazer o personagem andar passando o valor de **"Movendo"** como **Verdadeiro (true)**.

```
if (direcao != Vector3.zero)
{
    GetComponent<Animator>().SetBool("Movendo", true);
}
```

Agora resta somente o teste de caso ele não esteja se movendo porém testar em outro `if` é desnecessário já que temos o `else` para fazer isso, o `else` testa o posto do nosso `if` então se aquela condição não acontecer o que irá acontecer é o que está dentro do nosso `else`. Então se colocarmos as linhas abaixo será feito um teste se alguma tecla está sendo apertada caso isso aconteça nosso jogador irá se mover, caso isto não aconteça o código executa o que está dentro do `else` e irá fazer nosso jogador parar. É importante lembrar que isso acontece de imediato pois estas linhas estão dentro do nosso `Update` que roda o tempo todo no nosso jogo.

```
if (direcao != Vector3.zero)
{
    GetComponent<Animator>().SetBool("Movendo", true);
}
else
{
    GetComponent<Animator>().SetBool("Movendo", false);
}
```