

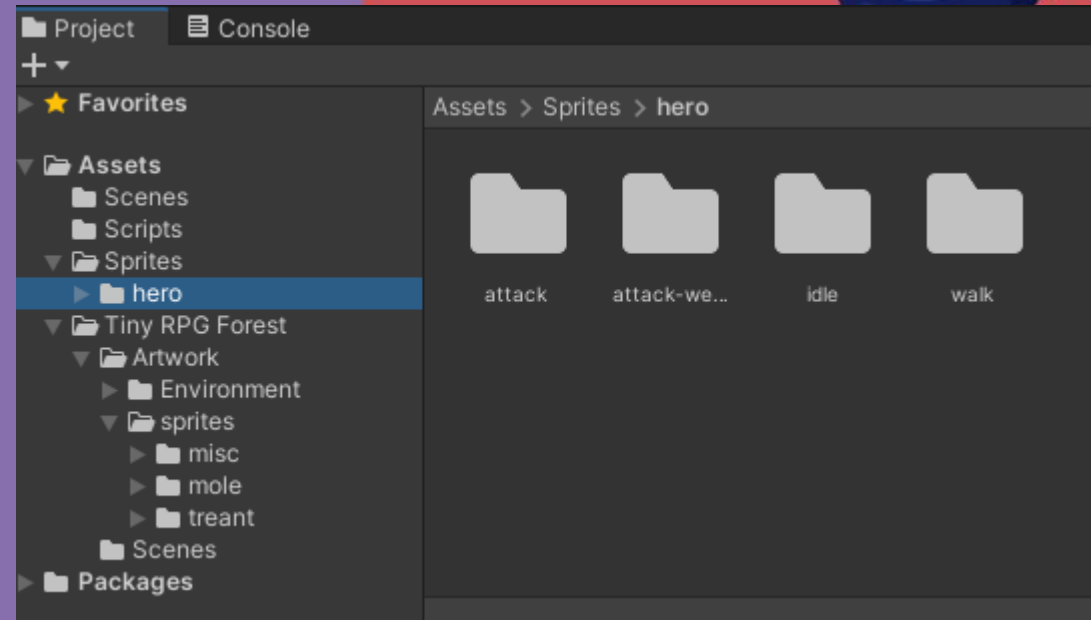
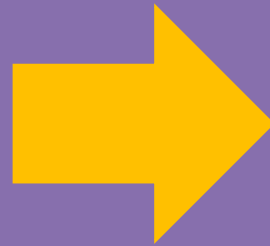
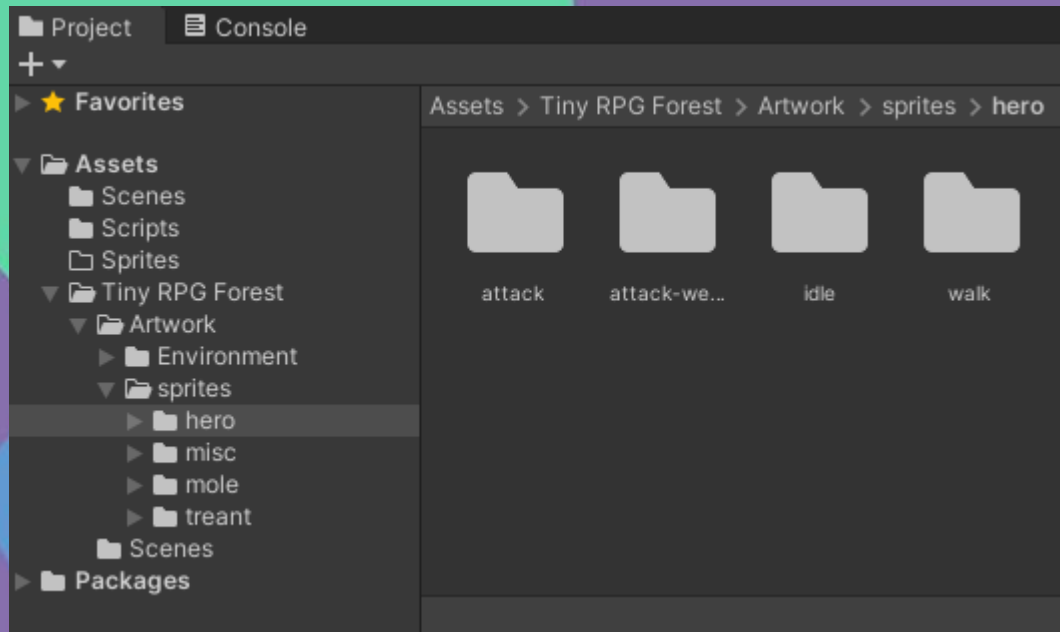


# CRIE SEU PRIMEIRO JOGO UTILIZANDO UNITY

- Projeto Tiny RPG Forest -  
(personagem)

# Projeto Tiny RPG Forest

- Ajustar o ASSET IMPORTADO ao nosso projeto
  - Selecionar de Tiny RPG Forest -> Artwork -> sprites -> hero
  - Arrastar essa pasta hero para dentro da nossa pasta Sprites



Conforme usamos o asset importado, vamos colocando dentro do nosso **PROJETO ORGANIZADO**

# Projeto Tiny RPG Forest

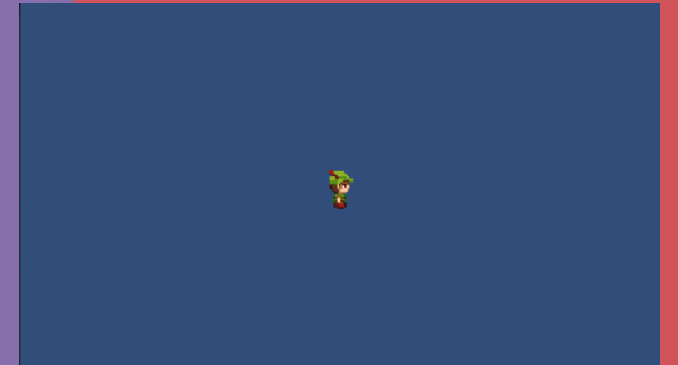
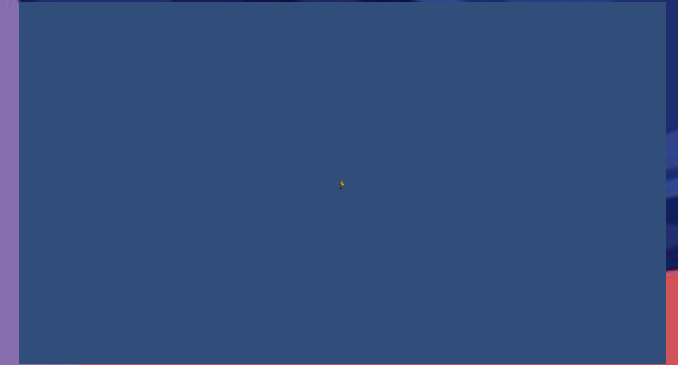
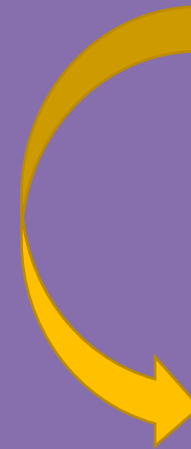
- O nosso personagem tem 4 estados:
  - Idle: estado inicial, parado
  - Walk: quando o personagem for andar
  - Attack: quando o personagem for atacar sem arma
  - Attack wepon: quando o personagem for atacar com arma
- Para cada um desses estados, temos direção:
  - Back: de costas
  - Front: de frente
  - Side: de lado



**Inicialmente vamos focar no personagem no estado IDLE na direção SIDE**

# Projeto Tiny RPG Forest


- Apagar o quadrado de Hierarchy
- Arrastar a figura hero-idle-side para dentro do Hierarchy
  - Nomear para Player
  - Setar o tag para Player
  - Execute o Unity ou visualize pelo View Game
- Nosso personagem está muito pequeno!
  - Para ajustar poderá mexer na escala
  - Ou ajustar a imagem:
    - Selecione a image hero-idle-side em Project, e ajuste no Inspector para que os pixels per unit sejam 20, ao invés de 100! Pressione o botão Apply.



**Lembrar de fazer o ajuste do tamanho para todas as imagens do personagem!**



# Projeto Tiny RPG Forest



Sua vez de fazer!

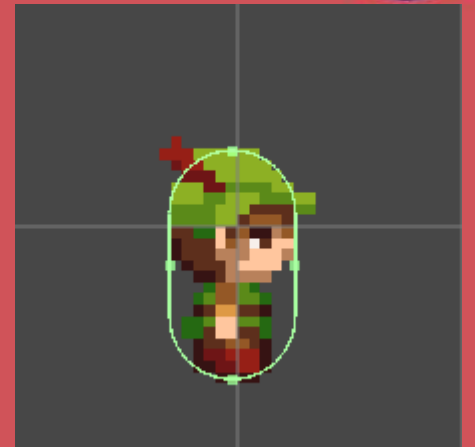
**Ajustar o tamanho em todas as imagens do  
personagem!**

- Ajustar a imagem:
  - Selecione a image em Project, e ajuste no Inspector para que os pixels per unit sejam 20, ao invés de 100!
  - Pressione o botão Apply.



# Projeto Tiny RPG Forest

- Com o Player
  - Adicionar um script scrPlayer
  - No script do Player, colocar as movimentações necessárias, como o quadrado (andar para a direita, esquerda, cima e embaixo)
  - Adicionar Rigidbody 2D, para acrescentar a física ao personagem e acionar a colisão
    - Congelar a rotação Z em Constraints
  - Adicionar um colisor do tipo CapsuleCollider 2D
  - Ajuste para que o colisor fique bem justo ao personagem



**Toda aula tem é uma revisão de conteúdo dado!**

# Projeto Tiny RPG Forest

Sua vez de fazer!

**Fazer o personagem andar**

- A codificação é igual ao quadrado





# Projeto Tiny RPG Forest

- Vamos ajustar o personagem para virar quando for para direita e para a esquerda. Para isso, vamos rotacionar a figura através do script do Player: scrPlayer
- Criar mais algumas variáveis:
- Ajustar o método Update():

```
public float eixoX = 1f;  
public bool olhandoDireita = true;
```

```
void Update()  
{  
    horizontal = Input.GetAxisRaw("Horizontal");  
    vertical = Input.GetAxisRaw("Vertical");  
    if (horizontal > 0 && !olhandoDireita)  
    {  
        eixoX = 1;  
        olhandoDireita = true;  
    }  
    else if (horizontal < 0 && olhandoDireita)  
    {  
        eixoX = -1;  
        olhandoDireita = false;  
    }  
    transform.localScale = new Vector3(eixoX, 1, 1);  
}
```





# Iniciando o Projeto Tiny RPG Forest

## Performance em jogos

Segundo passo  
utilizar  
operações  
performáticas:  
**eixoX \*= -1;**

Primeiro passo  
modularizar: criar  
um método para  
fazer a virada  
**void Virar(){}**

```
void Update()
{
    horizontal = Input.GetAxisRaw("Horizontal");
    vertical = Input.GetAxisRaw("Vertical");
    if (horizontal > 0 && !olhandoDireita) {
        Virar();
    }
    else if (horizontal < 0 && olhandoDireita) {
        Virar();
    }
}

2 referências
void Virar()
{
    olhandoDireita = !olhandoDireita;
    eixoX *= -1;           // é a mesma coisa que eixoX = eixoX * -1;
    transform.localScale = new Vector3(eixoX, 1, 1);
}
```



# Iniciando o Projeto Tiny RPG Forest

## Performance em jogos

Terceiro passo utilizar valores já dispostos:  
**transform.localScale**

Quarto passo retirar espaços desnecessários

```
void Update()
{
    horizontal = Input.GetAxisRaw("Horizontal");
    vertical = Input.GetAxisRaw("Vertical");
    if (horizontal > 0 && !olhandoDireita) {
        Virar();
    }
    else if (horizontal < 0 && olhandoDireita) {
        Virar();
    }
}

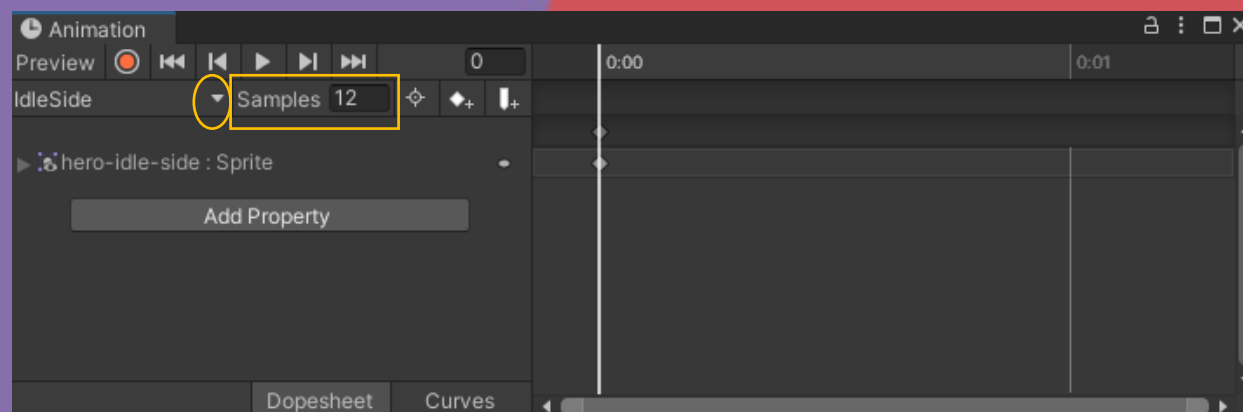
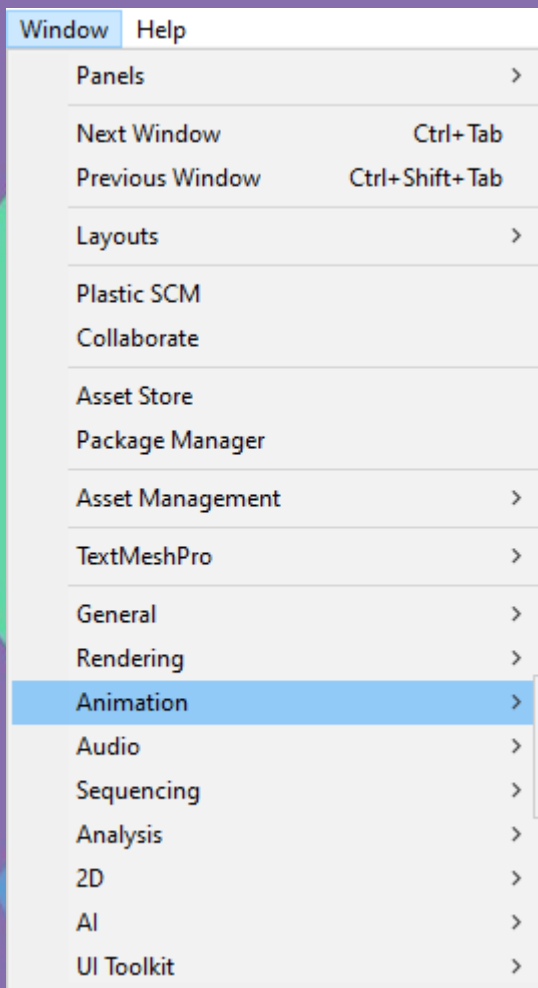
2 referências
void Virar()
{
    olhandoDireita = !olhandoDireita;
    eixoX = transform.localScale.x;
    eixoX *= -1;           // é a mesma coisa que eixoX = eixoX * -1;
    transform.localScale = new Vector3(eixoX, transform.localScale.y, transform.localScale.z);
}
```



# Projeto Tiny RPG Forest

## Utilizando o ANIMATION:

- Selecionar o Player em Hirarchy
- Abrir o Animation (Window -> Animation -> Animation)
- Criar uma animação IdleSide, pressionando o botão **CREATE**
  - **ORGANIZAÇÃO**: guardar as animações na pasta Animation, neste caso, dentro desta pasta acrescentar a pasta Hero
- Arrastar a imagem da pasta Asset -> Sprites -> hero -> idle -> hero-idle-side para o lado direito do Animation. Salva automaticamente
- Ajustar o samples Para 12 quadros



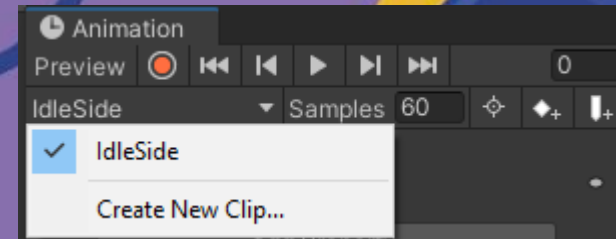
**Animação é no ANIMATION!**



# Projeto Tiny RPG Forest

Utilizando o ANIMATION -> Hero:

- No ANIMATION, onde está IdleSide, abra o menu suspenso e crie uma nova animação
- Selecionar todas as imagens de:
  - Asset -> Sprites -> hero -> Walk -> hero-Walk-side
- Arrastar as imagens selecionadas para dentro do ANIMATION e criar a animação WalkSide
- Dê o play no Animation e veja o que acontece na View Game



**Animação é no ANIMATION!**

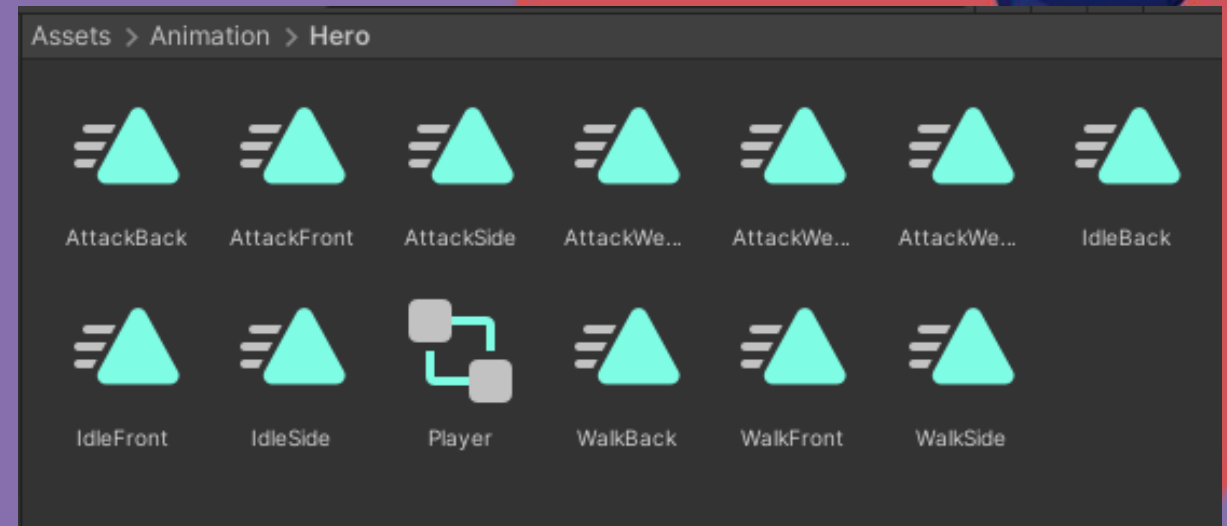


# Projeto Tiny RPG Forest

Sua vez de fazer!

Fazer todas as animações do personagem

- AttackBack
- AttackFront
- AttackSide
- AttackWeapon
- AttackWeaponBack
- AttackWeaponFront
- ☒ AttackWeaponSide
- IdleBack
- IdleFront
- IdleSide
- WalkBack
- WalkFront
- WalkSide



# Projeto Tiny RPG Forest

Fazer todas as animações do personagem

Sua vez de fazer!

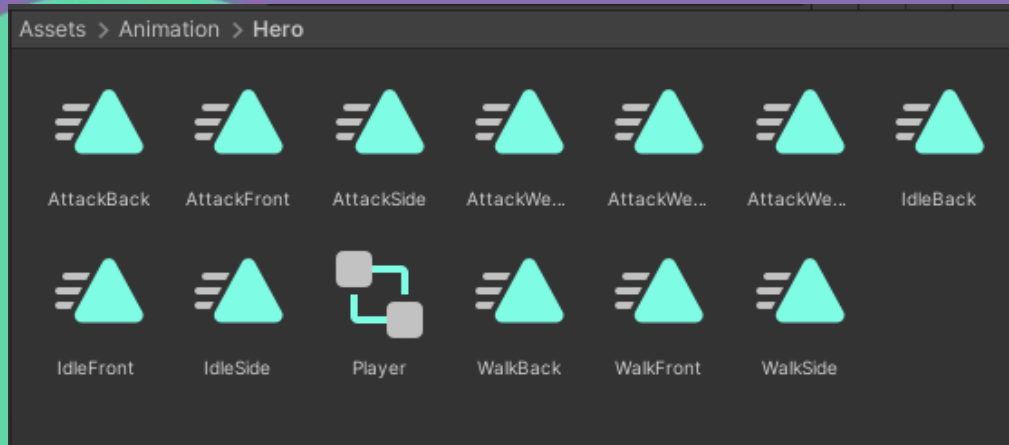
## Utilizando o ANIMATION:

- Selecionar o Player em Hirarchy
- Abrir o Animation (Window -> Animation -> Animation)
- Criar uma animação X, pressionando o botão **CREATE**
  - **ORGANIZAÇÃO:** guardar as animações na pasta Animation, neste caso, dentro desta pasta acrescentar a pasta Hero
- Arrastar a imagem da pasta Asset -> Sprites -> hero -> X para o lado direito do Animation. Salva automaticamente
- Ajustar o samples  
Para 12 quadros

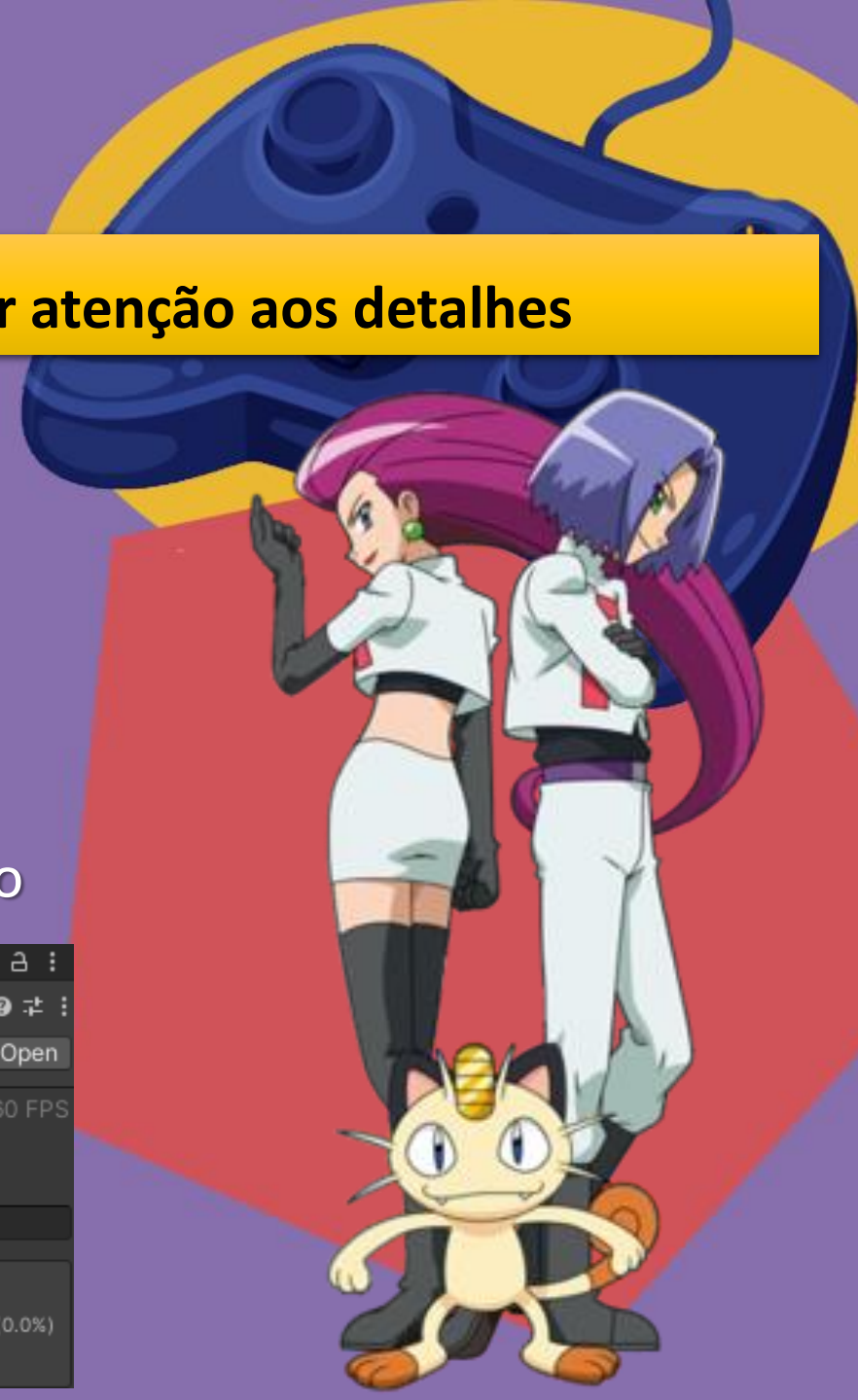
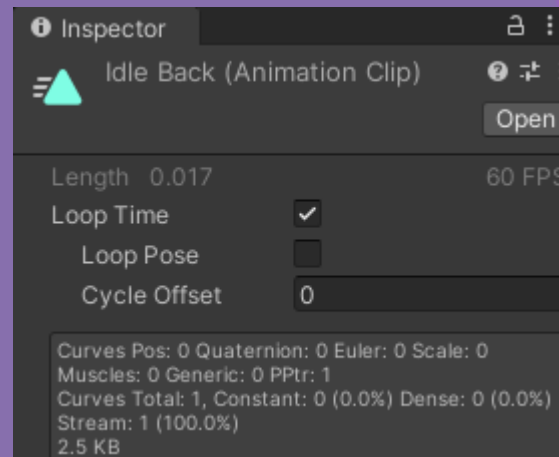


# Projeto Tiny RPG Forest

Prestar atenção aos detalhes



- Todas as animações tem como padrão o LOOP automático
- Se for necessário, desmarcar o LOOP
  - Selecionar a animação
  - No **Inspector**, desmarcar **Loop Time**

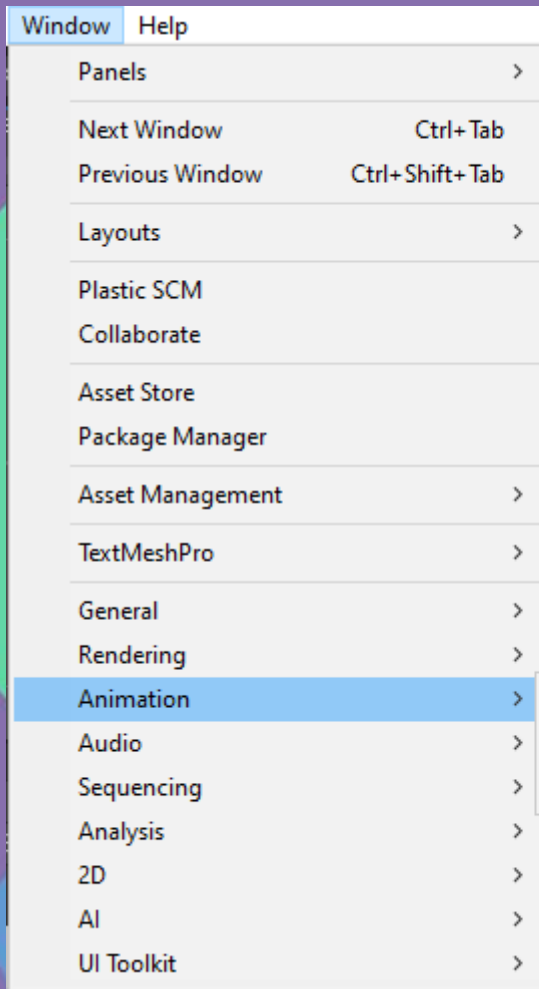




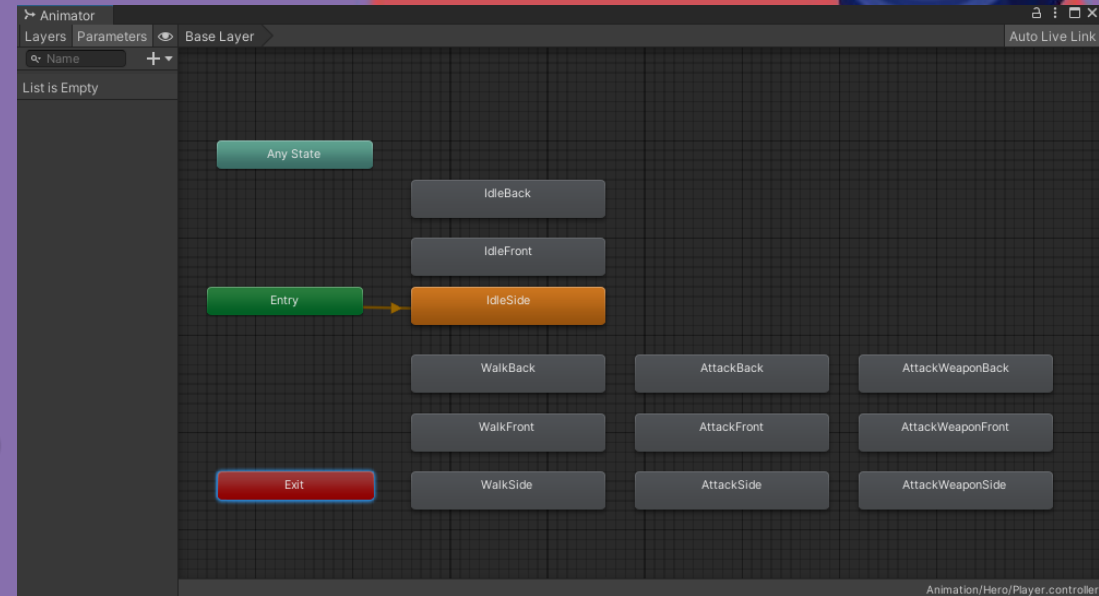
# Projeto Tiny RPG Forest

## Utilizando o ANIMATOR:

- Selecionar o Player em Hirarchy
- Abrir o Animation (Window -> Animation -> Animator)
- Vai aparecer todas as animações que foram feitas
- As animações ficam na pasta Asset -> Animation



- Setar para que o primeiro estado seja idleSide



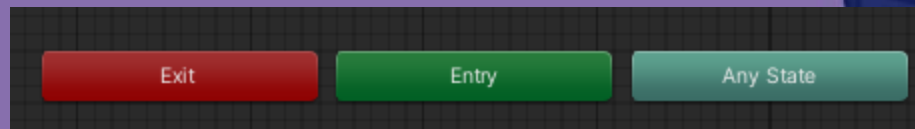
**Transição de animações é no ANIMATOR!**



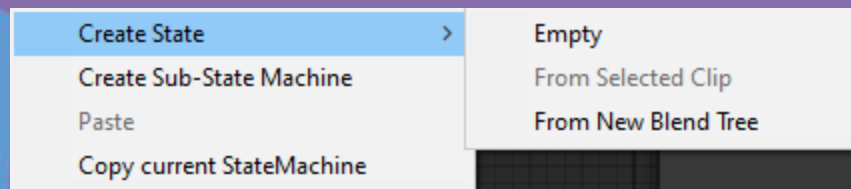
# Projeto Tiny RPG Forest

## Utilizando o ANIMATOR:

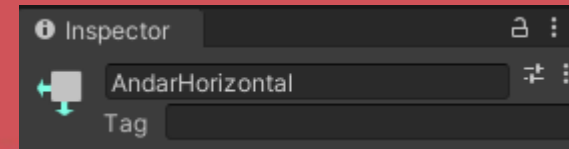
- Ainda no animator, apagar todas as transições criadas, ficando somente **ANY STATE**, **ENTRY** e **EXIT**



- Criar um parâmetro velocidadeHorizontal do tipo float
- Criar uma forma de controle chamada BlendTree e nomear como AndarHorizontal
  - BlendTree é uma estrutura de controle com múltiplas animações que serão mescladas suavemente
  - Para criar, pressione o botão direito do mouse sobre a área cinza do Animator

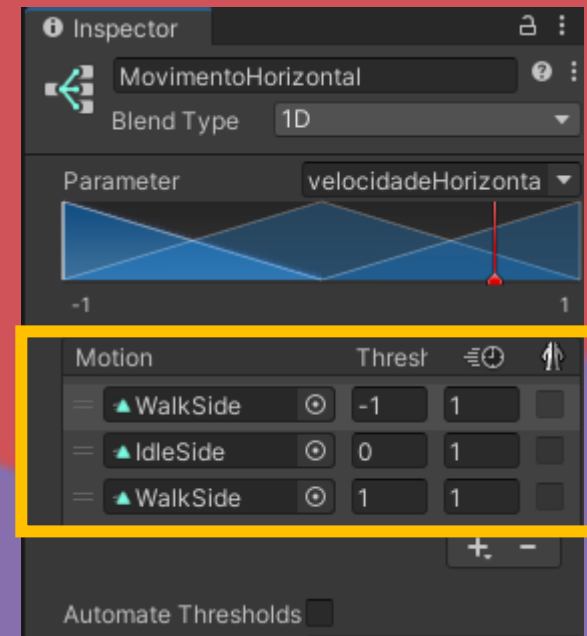
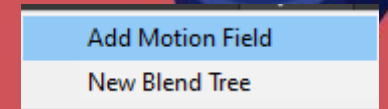
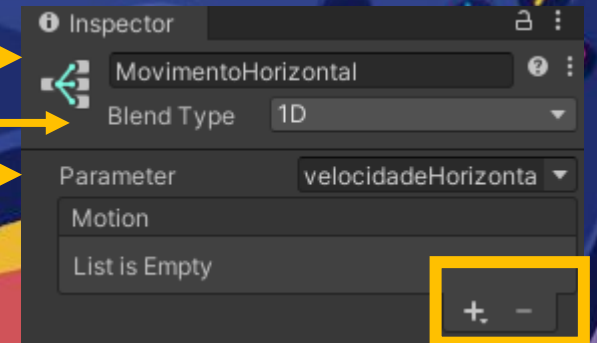
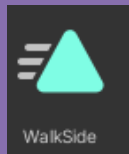
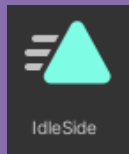
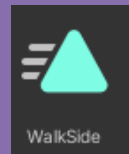


- Para nomear, é necessário selecionar o Blend Tree e utilizar o Inspector
- Para entrar no controlador do AndarHorizontal, é necessário dar um duplo clique nele



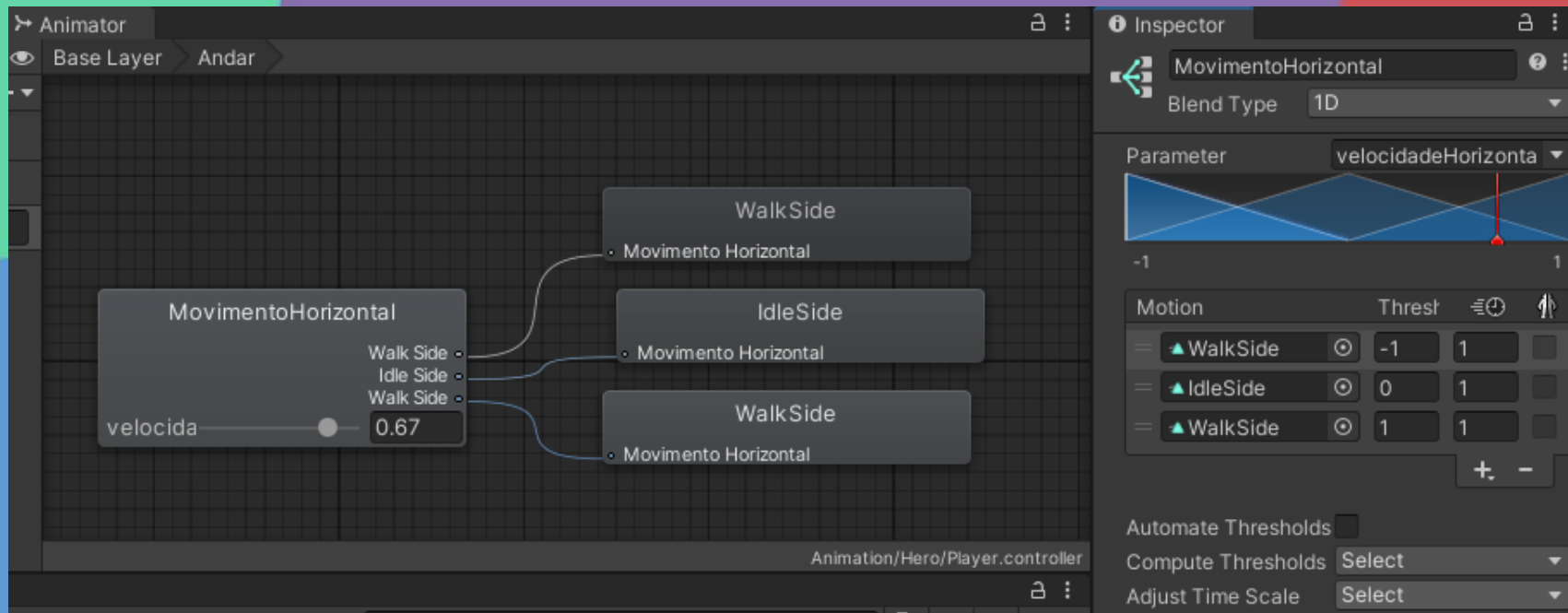
# Projeto 4 – SunnyLand – Player

- Mudar o nome para MovimentoHorizontal
- Deixar o BlendType 1D (baseado em 1 valor para sua alternância, 1 parâmetro)
- Parâmetro velocidadeHorizontal
- Movimento serão 3:
  - Andar para a esquerda
  - Parado
  - Andar para a direita



# Projeto 4 – SunnyLand – Player

- Arrastar/Selecionar a animação referente para cada espaço
- Desmarcar o Thresholds, porque vou definir quais são os valores
- E definir os valores -1, 0 e 1 que são as velocidades esperadas



# Projeto 4 – SunnyLand – Player

- Ajustar a codificação do scrPlayer:
  - Criar mais uma variável: `Animator animPlayer;`
  - Ajustar o método Update() para usar o Animator:


```
void Update()
{
    horizontal = Input.GetAxisRaw("Horizontal");
    vertical = Input.GetAxisRaw("Vertical");
    animPlayer.SetFloat("velocidadeHorizontal", horizontal);

    if (horizontal > 0 && !olhandoDireita) { Virar(); }
    else if (horizontal < 0 && olhandoDireita) { Virar(); }
}
```





# Projeto Tiny RPG Forest



Sua vez de fazer!

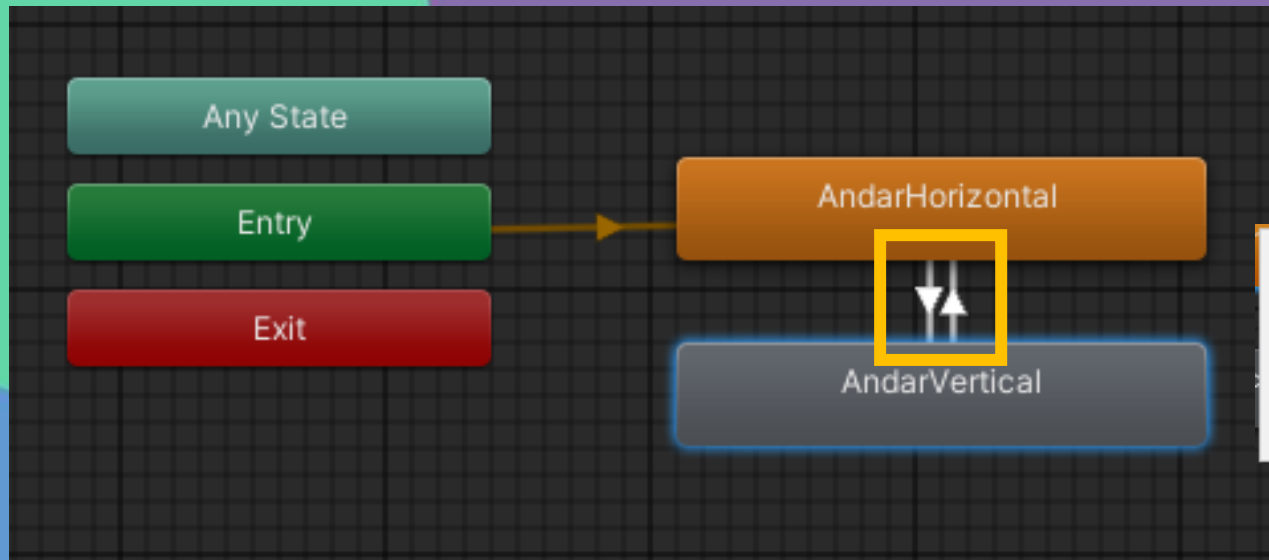
**Fazer um blendtree para o movimento vertical**



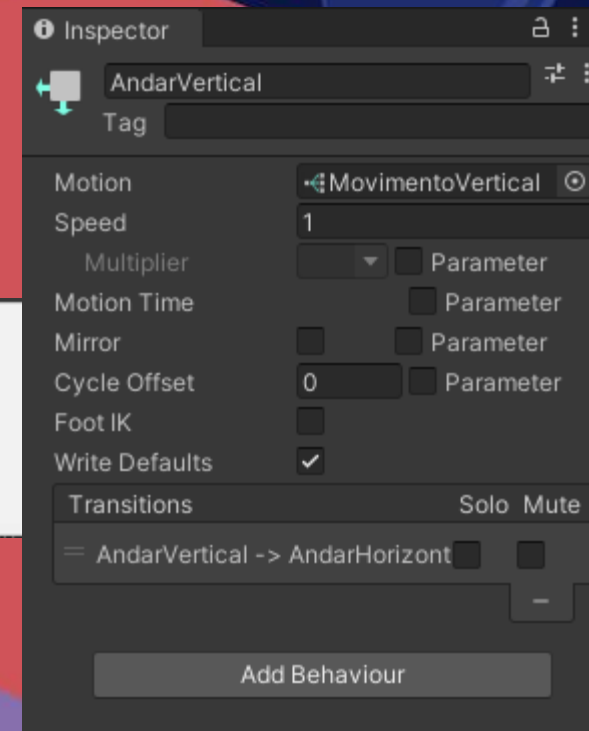
# Projeto Tiny RPG Forest

## Utilizando o ANIMATOR:

- Criar um parâmetro velocidadeVertical do tipo *Float*
- Criar um parâmetro olhandoCimaBaixo do tipo *Trigger*
- Criar uma forma de controle chamada BlendTree e nomear como **AndarVertical**
  - Para entrar no controlador do AndarVertical



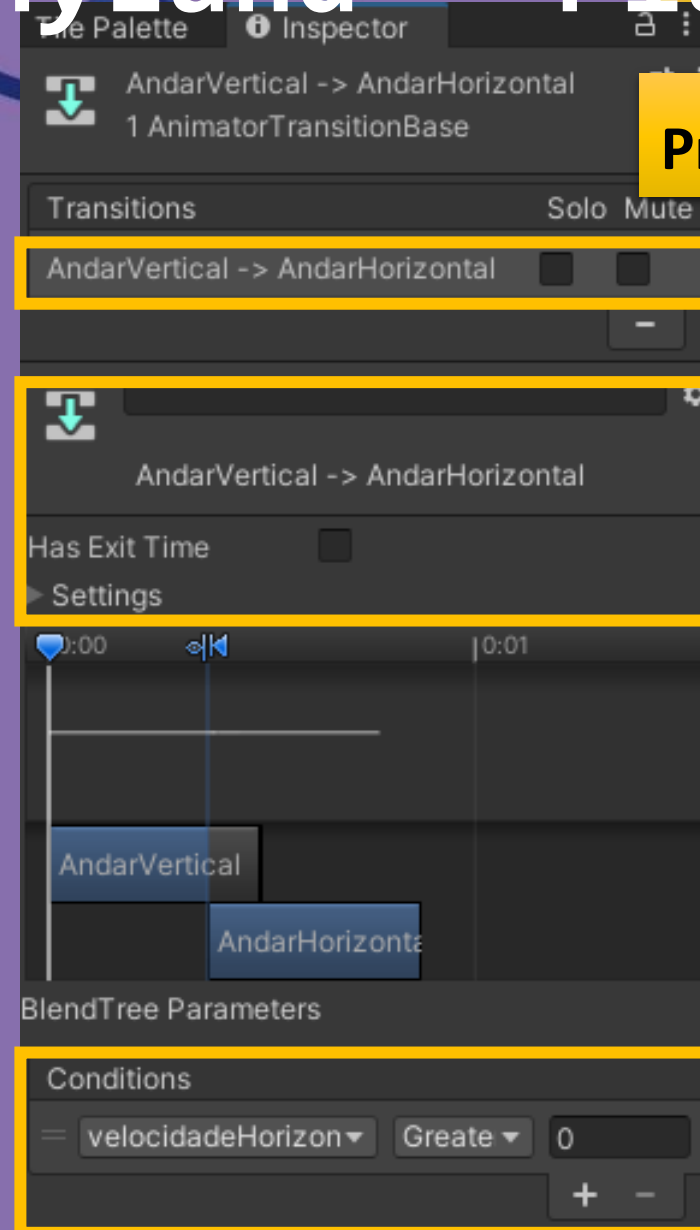
- Make Transition
- Set as Layer Default State
- Copy
- Create new BlendTree in State
- Delete



- Criar transições, botão direito em cima do estado

# Projeto 4 – SunnyLand – Player

- Transição AndarHorizontal para AndarVertical
  - Não ter demora nas transições das animações
  - Depender do parâmetro criado olhandoCimaBaixo



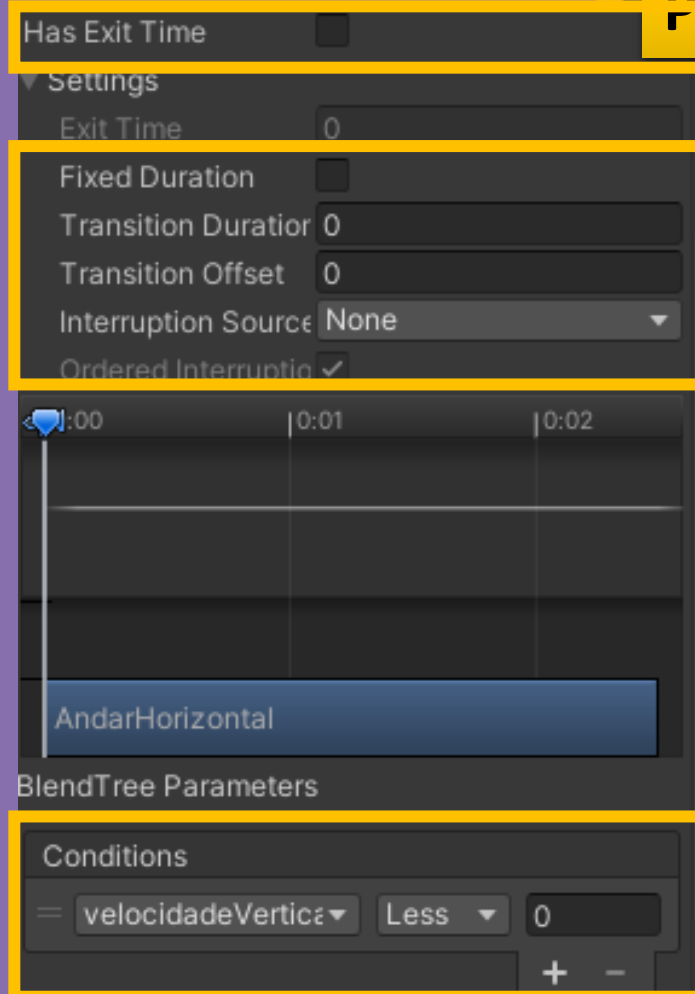
**Prestar atenção aos detalhes**





# Projeto 4 – SunnyLand – Player

- Transição AndarVertical para AndarHorizontal
  - Não ter demora nas transições das animações
  - Depender do parâmetro criado velocidadeVertical sendo que fique menor do que 0



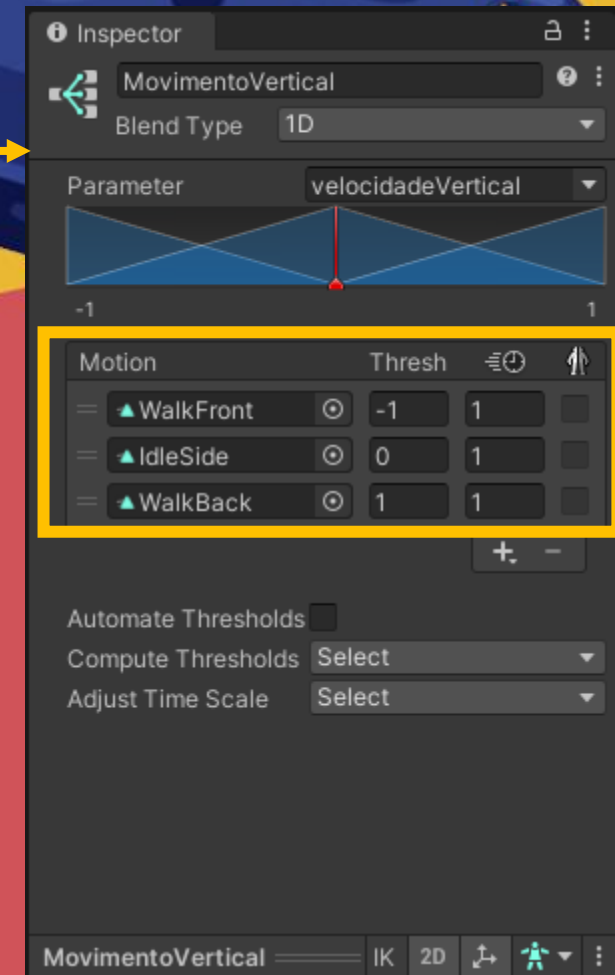
Prestar atenção aos detalhes





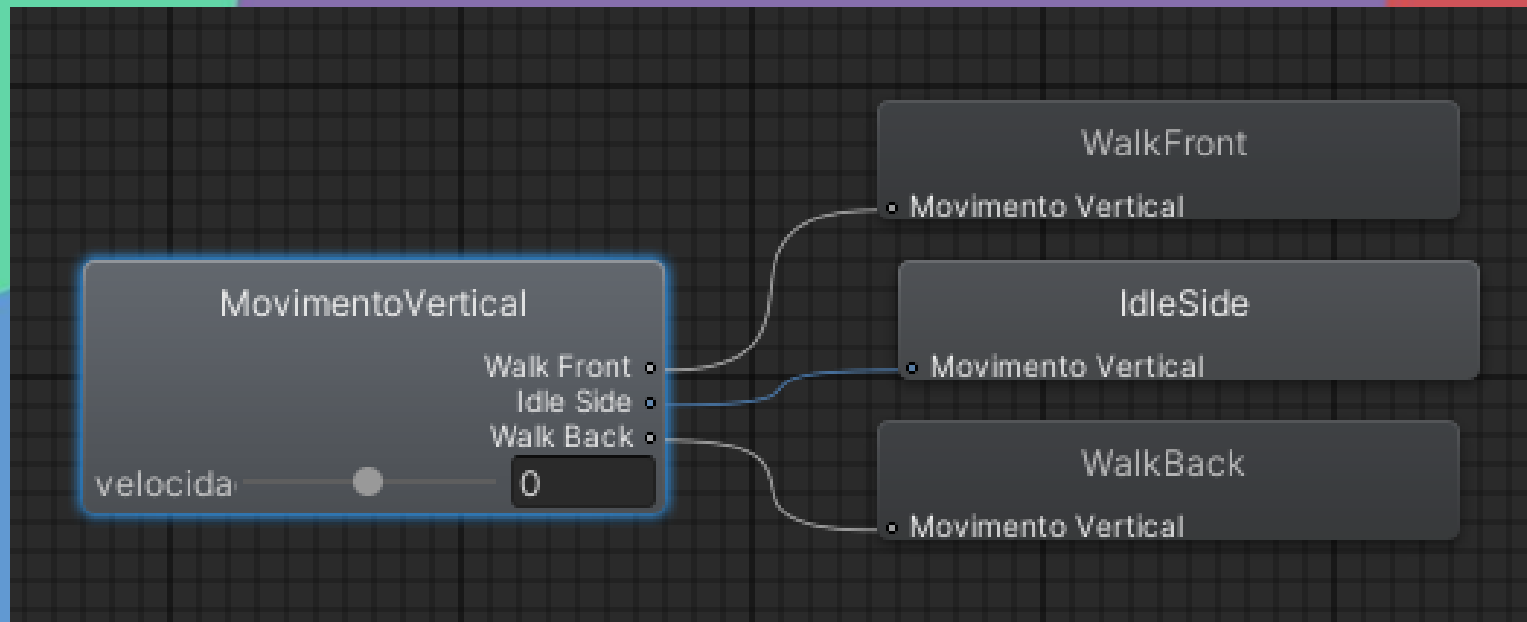
# Projeto 4 – SunnyLand – Player

- Mudar o nome para MovimentoVertical
- Deixar o BlendType 1D (baseado em 1 valor para sua alternância, 1 parâmetro)
- Parâmetro velocidadeVertical
- Movimento serão 3:
  - Andar para a cima
  - Parado
  - Andar para a baixo



# Projeto 4 – SunnyLand – Player

- Arrastar/Selecionar a animação referente para cada espaço
- Desmarcar o Thresholds, porque vou definir quais são os valores
- E definir os valores -1, 0 e 1 que são as velocidades esperadas



# Projeto 4 – SunnyLand – Player

- Ajustar a codificação do scrPlayer:

- Criar as variáveis: 

```
public bool olhandoCima = false;  
public bool olhandoBaixo = false;
```

- Ajustar o método Update() para usar o Animator:

```
void Update()  
{  
    horizontal = Input.GetAxisRaw("Horizontal");  
    vertical = Input.GetAxisRaw("Vertical");  
    animPlayer.SetFloat("velocidadeHorizontal", horizontal);  
    if(horizontal == 0 && vertical == 0) { EstarParado(); }  
    else {  
        if (horizontal > 0 && !olhandoDireita) { Virar(); }  
        else if (horizontal < 0 && olhandoDireita) { Virar(); }  
        if (vertical > 0 || vertical < 0) { SubirDescer(); }  
    }  
}
```





# Projeto 4 – SunnyLand – Player

- Ajustar a codificação do scrPlayer:
  - Criar os seguintes métodos:

```
void EstarParado()
{
    olhandoDireita = true;
    olhandoCima = false;
    olhandoBaixo = false;
    eixoX = transform.localScale.x;
    if (eixoX < 0) { eixoX *= -1; }
    animPlayer.SetFloat("velocidadeVertical", vertical);
    animPlayer.SetFloat("velocidadeHorizontal", horizontal);
    transform.localScale = new Vector3(eixoX, transform.localScale.y, transform.localScale.z);
}
1 referência
void SubirDescer()
{
    animPlayer.SetFloat("velocidadeVertical", vertical);
    animPlayer.SetTrigger("olhandoCimaBaixo");
}
```



PRONTO!

Aguardo vocês na próxima  
aula, com o UNITY ABERTO  
NO PROJETO!

