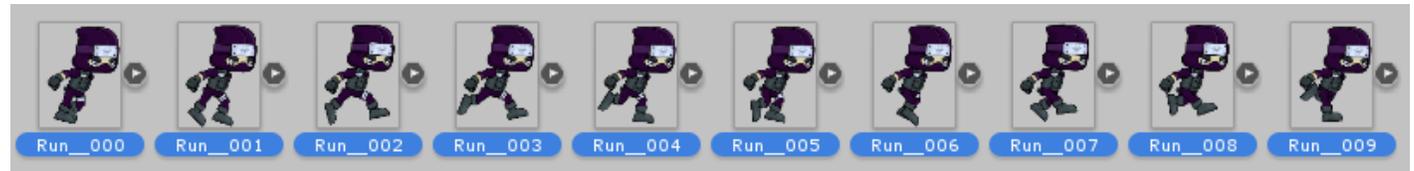


# ANIMAÇÕES

Prof. Me. Hélio Esperidião

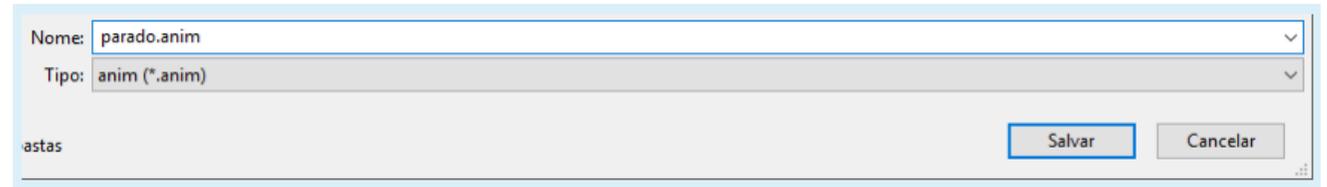
# Sprite

- Os sprites foram inventados como um método rápido de animação. Utilizando várias imagens agrupadas numa tela.



# Selecione os assets que compõem uma animação.

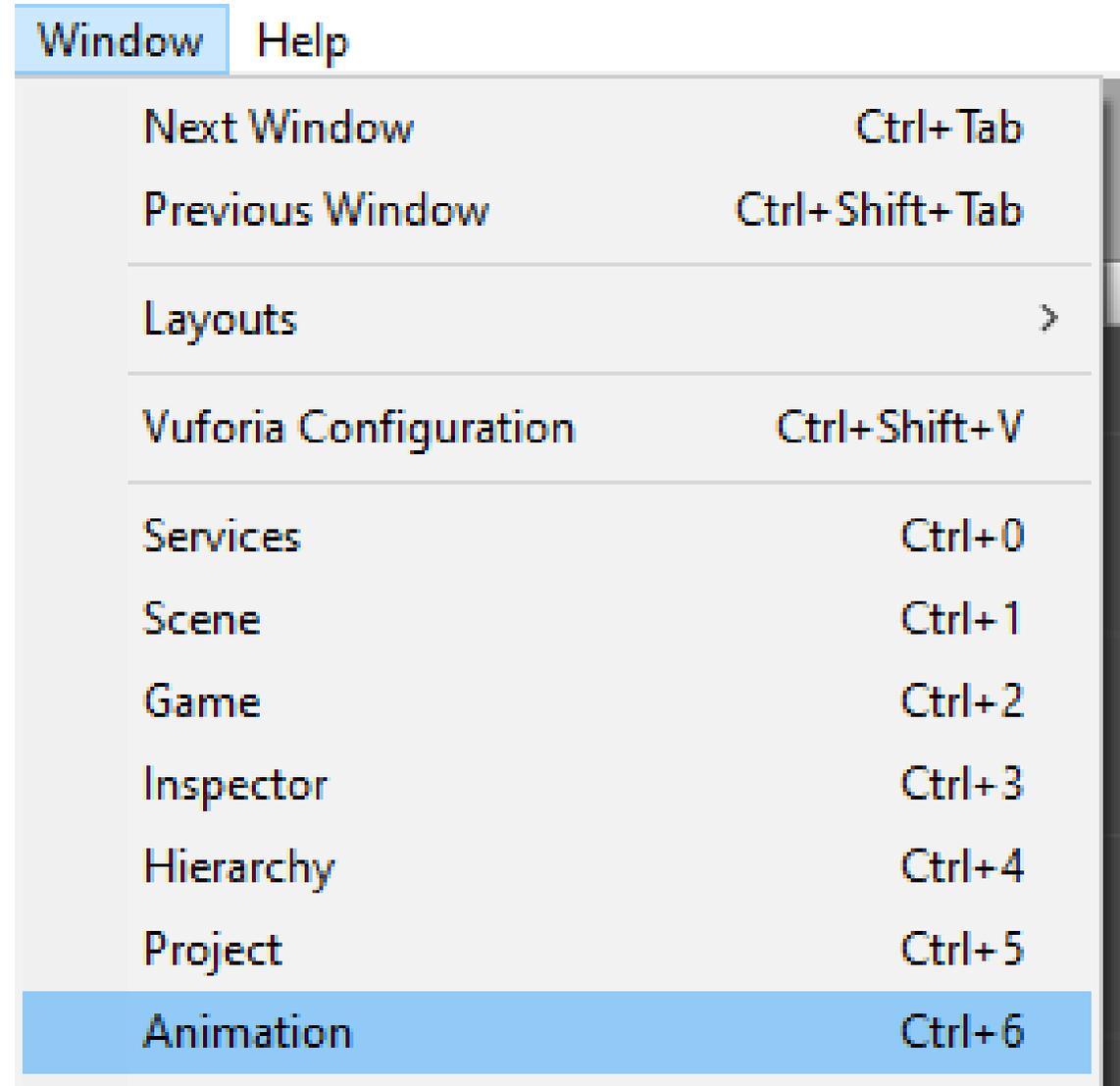
- Arraste para a cenário todas as imagens que formam uma animação de uma única vez.
- Ao soltar as imagens o unity vai solicitar que você salve uma animação.
- Salve a primeira animação com nome de “Parado.anim”



# Abra a janela de animation

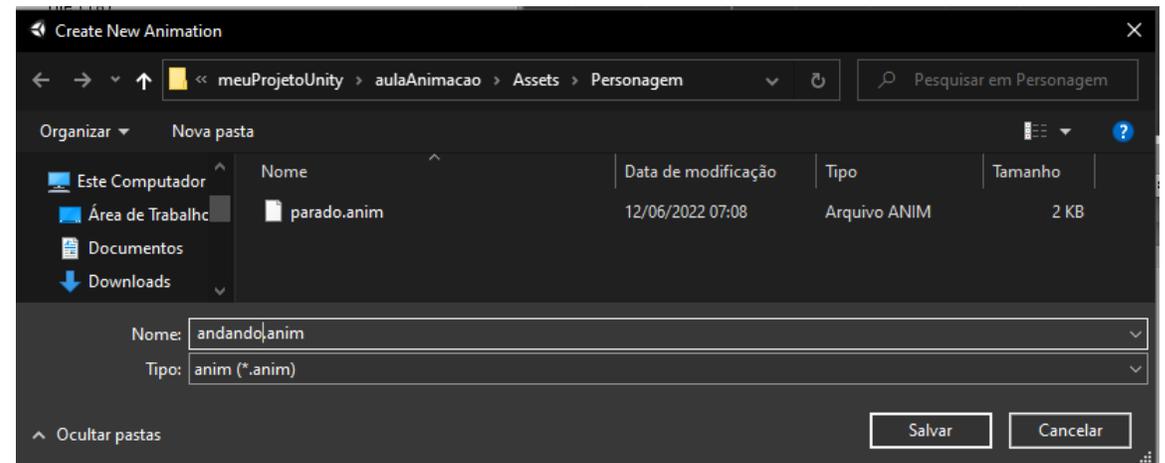
---

- O Animation permite configurar os assets que são parte de uma animação.
- Também permite configurar outras animações (andando, pulando, correndo)



# Criar novo Clip (Animação)

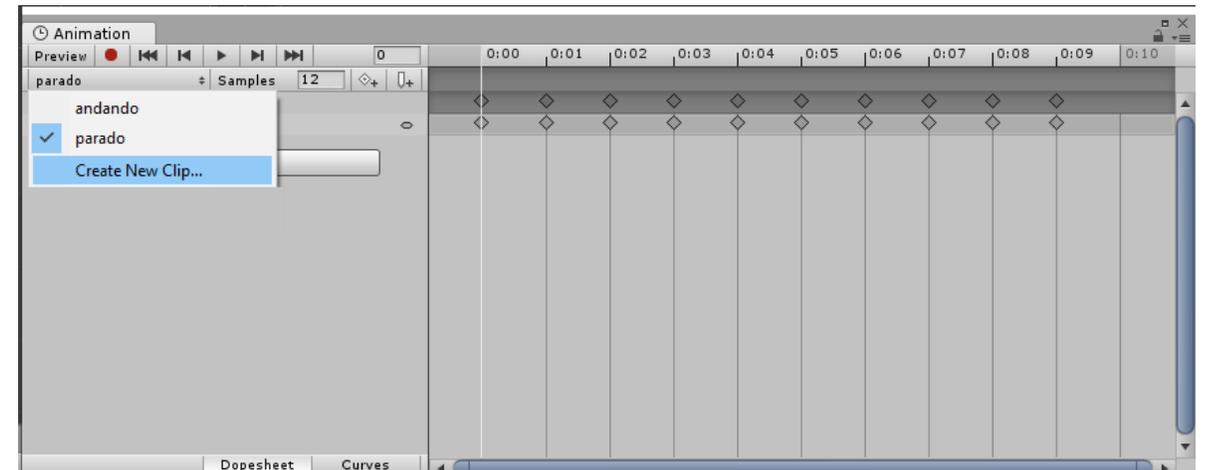
- Ao arrastar as imagens que compõem a animação de andando o Unity pedirá para salvar um arquivo. Anim
- Salve como “andando.anim”
- Verifique que já existe o arquivo parado.anim que foi criado anteriormente.



# Click em parado e selecione “create new clipe”

---

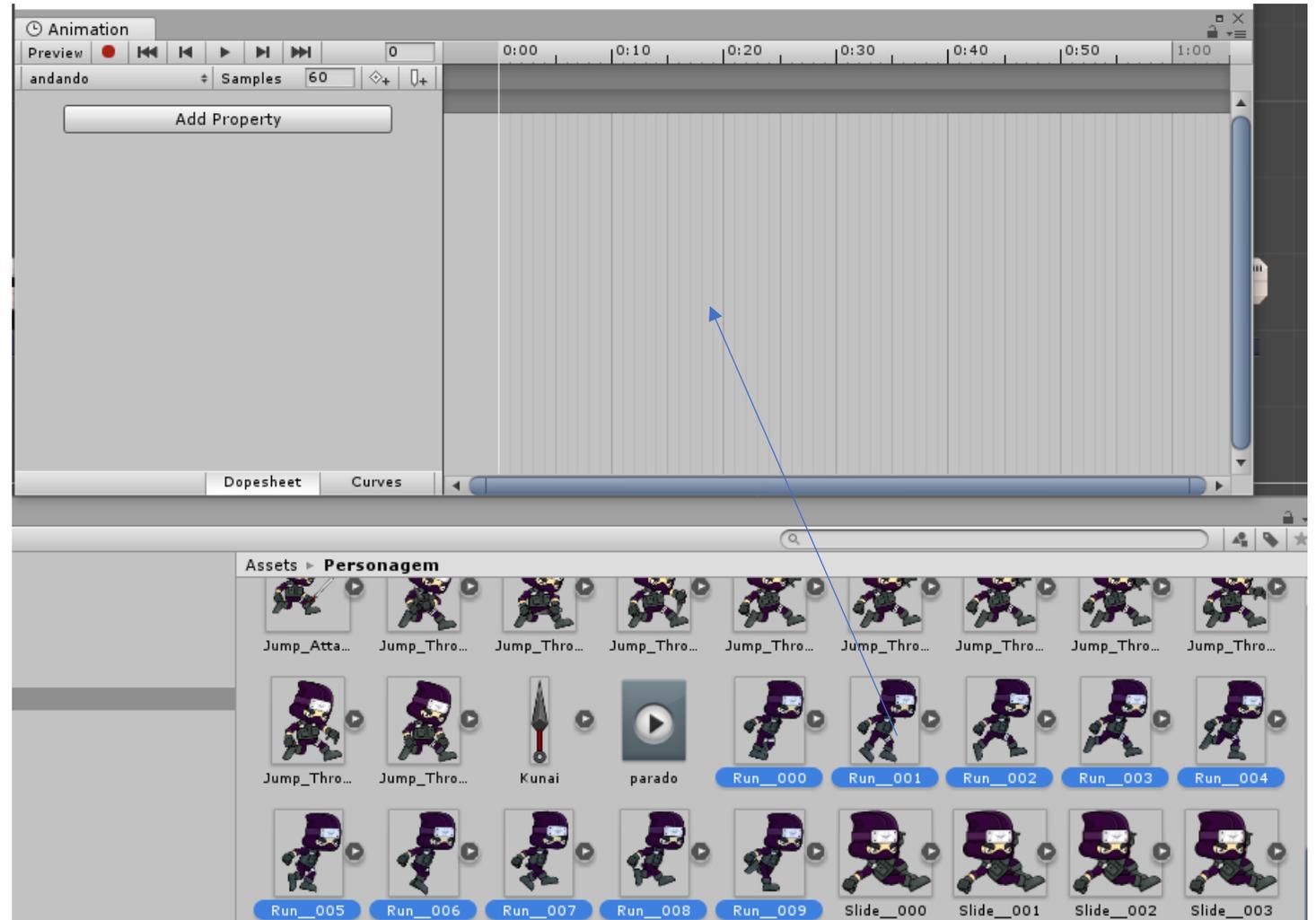
- Click em “Create new Clip”
  - Crie um novo Clip com nome de “andando”
- Um clip é uma animação, exemplo:
  - Clip andando
  - Clip parado
  - Clip pulando



# Criar nova animação

---

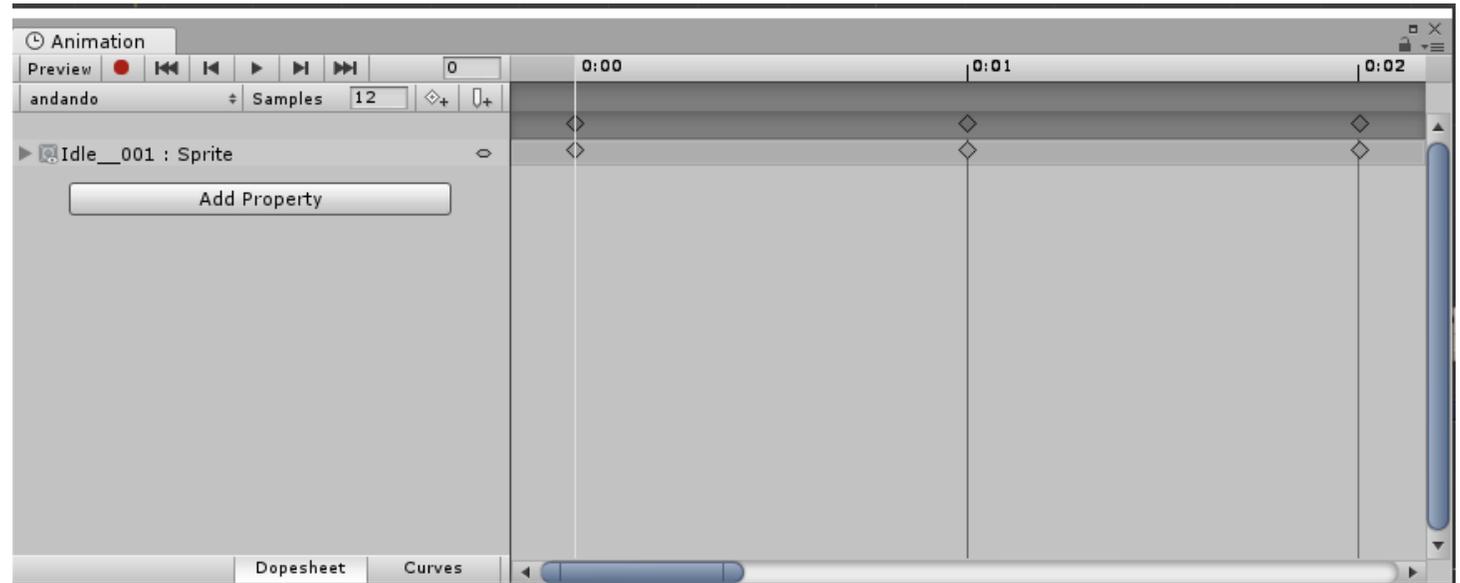
- Selecione todas as imagens da animação
- Arraste para dentro da time-line do Animation

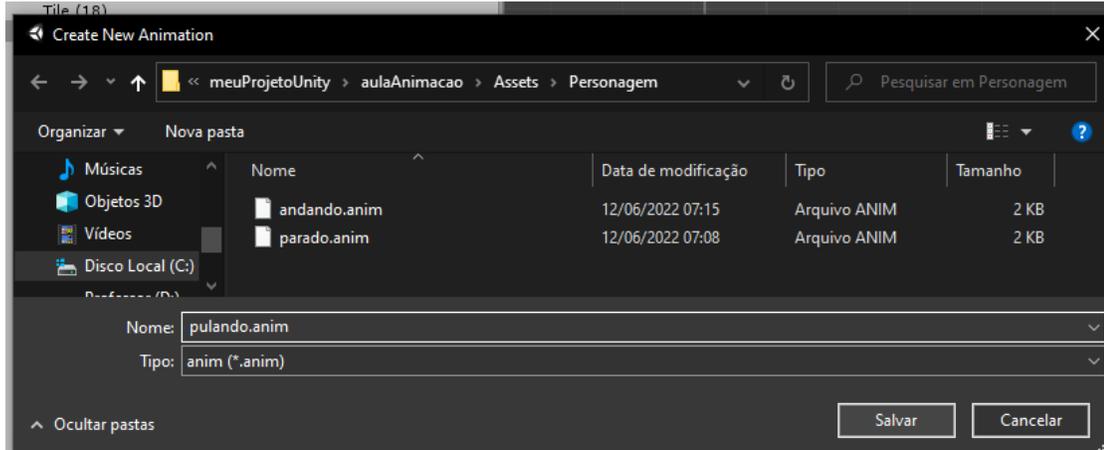


# Samples

---

- **Samples:** Taxa de amostragem, Modifique de 60 para 12
- Em todos os clipes modifique para 12.



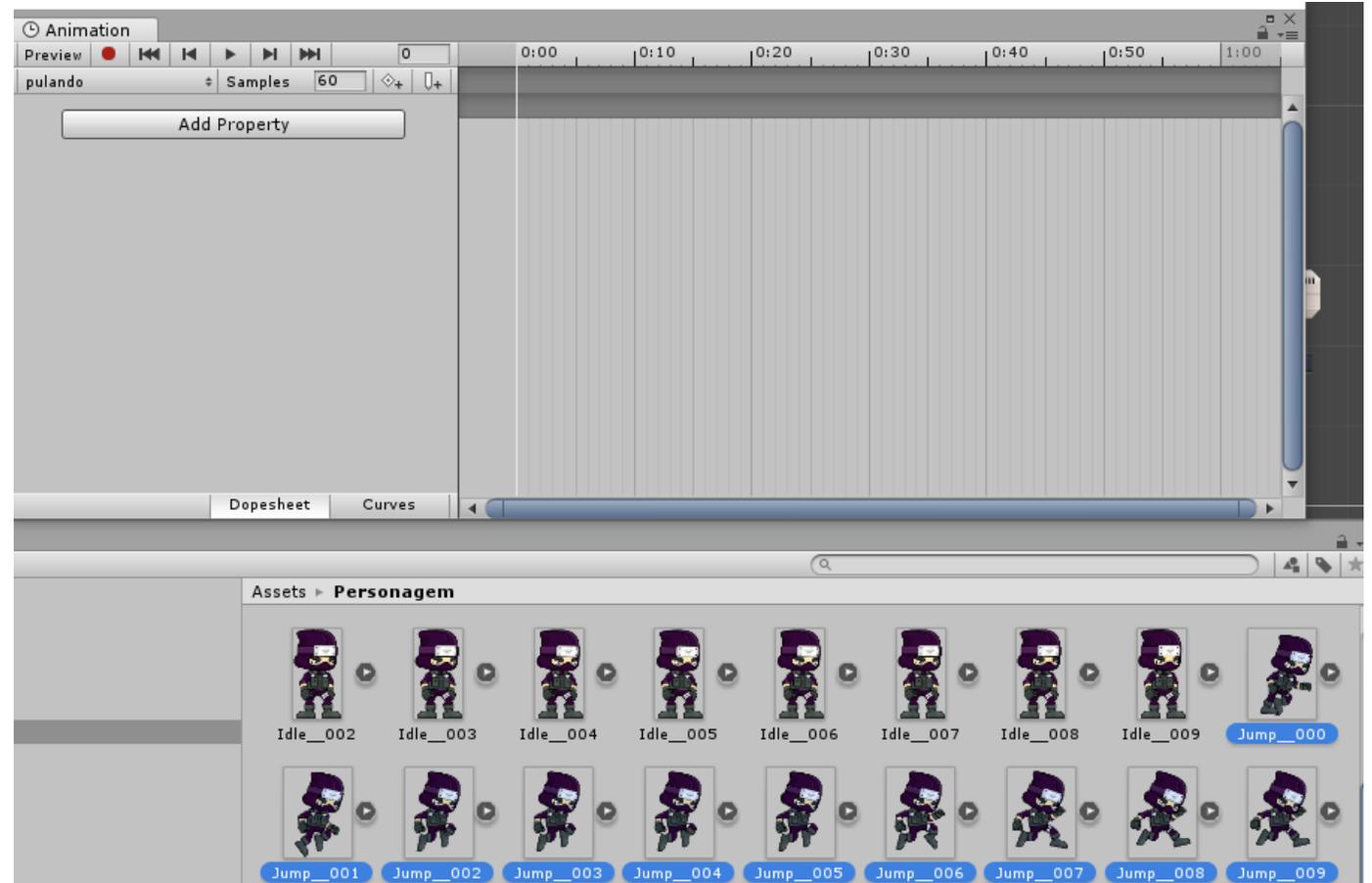


# Crie outra animação de pulando

- Click novamente em “Create new clip”

# Animação de jump(pulo)

- Selecione todas as imagens da animação
- Arraste para dentro da time-line do Animation
- Salve como pulando.anim



# ANIMATOR

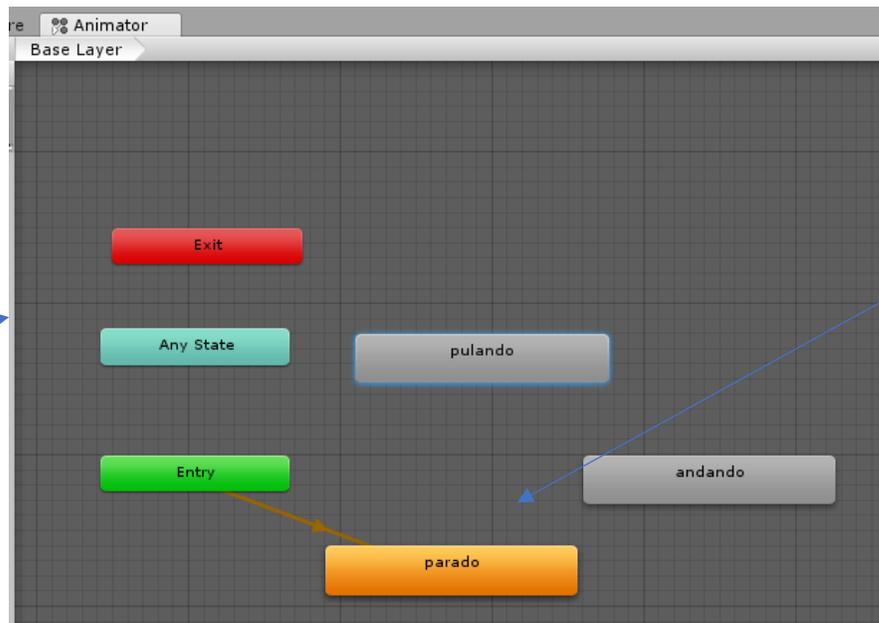
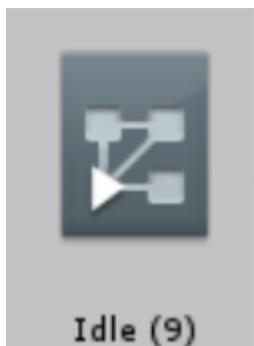
- Ao criar uma animação também é criado automaticamente um animador,
- Ele pode ser encontrado em meio a seus assets por meio da imagem ao lado



Idle (9)

# ANIMATOR

- Ao clicar duas vezes no componente a esquerda o animator irá aparecer em sua janela.
- O componente foi criado quando você arrastou os primeiros acessos da animação idle para dentro do cenário.



Animação criada quando você arrastou as imagens para o scene

# Transição

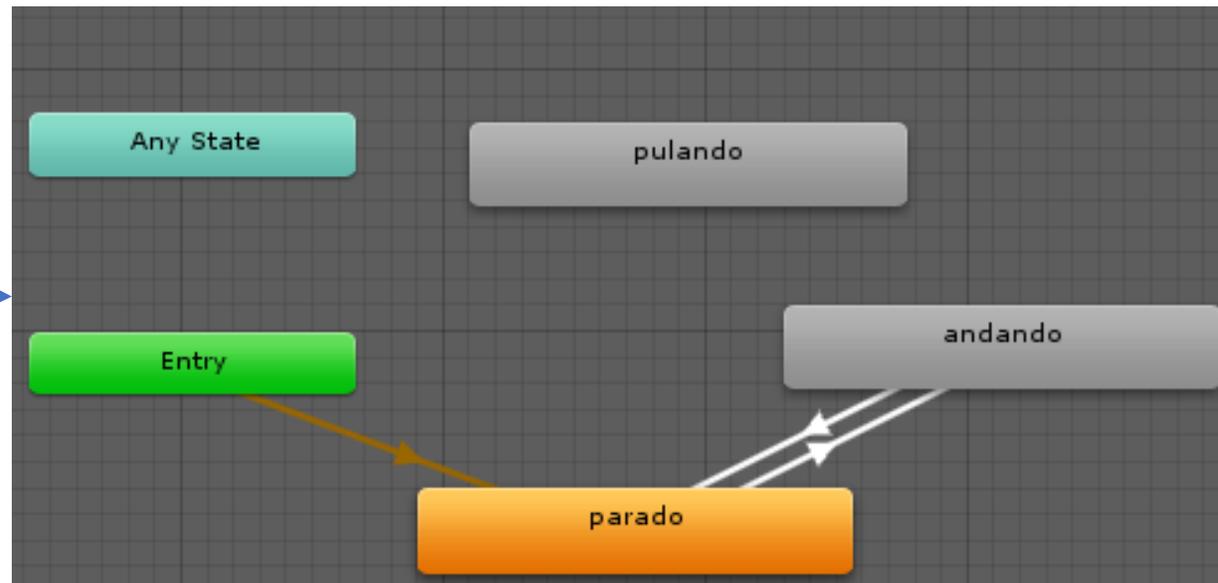
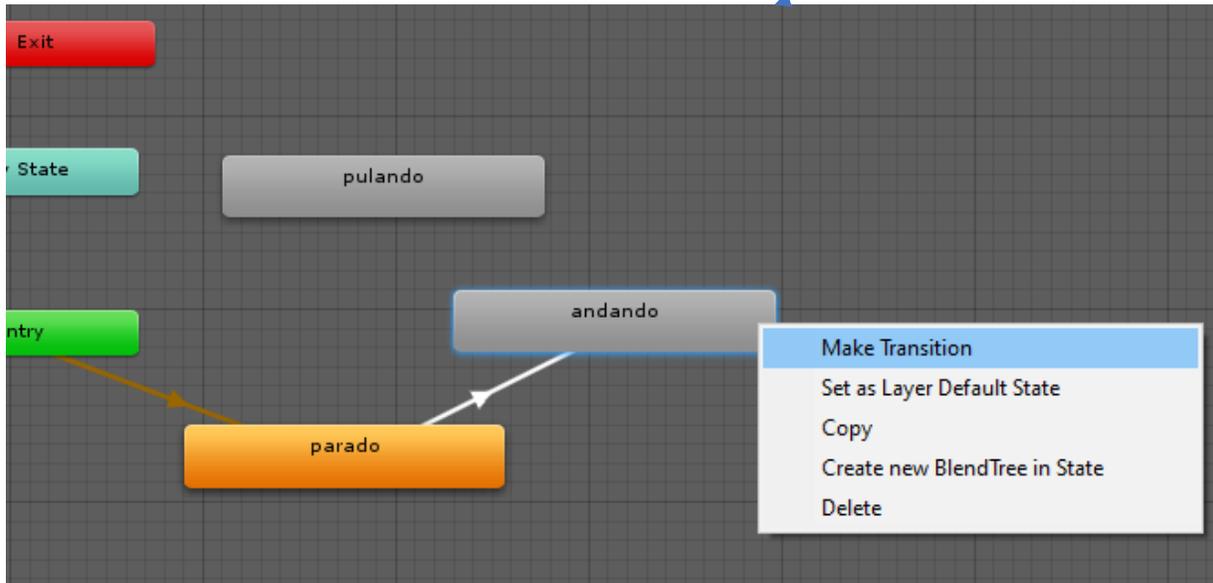
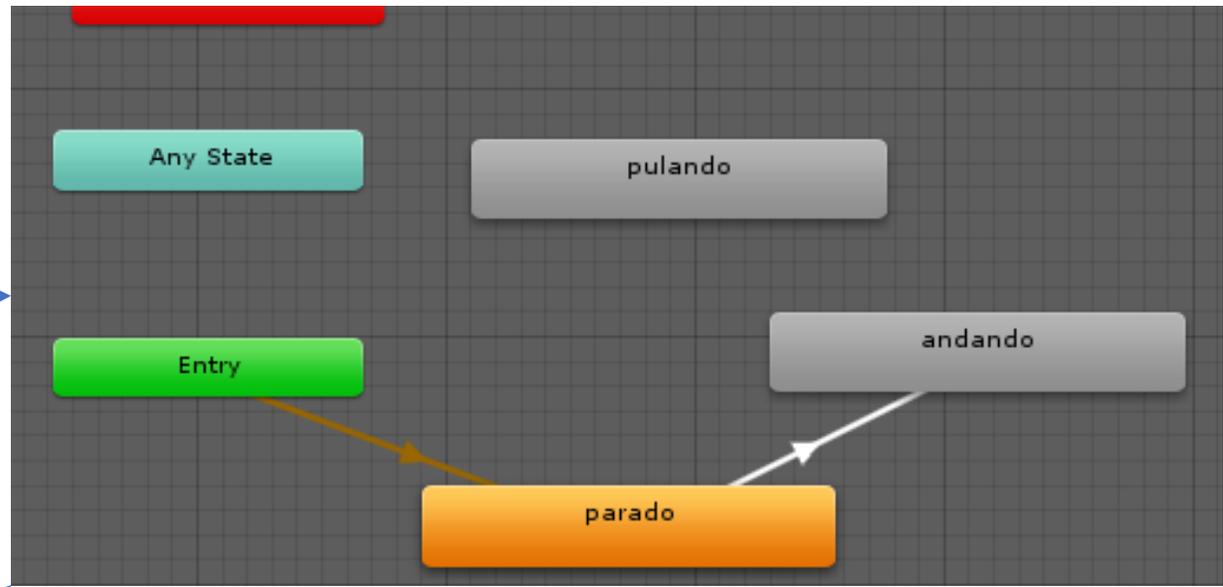
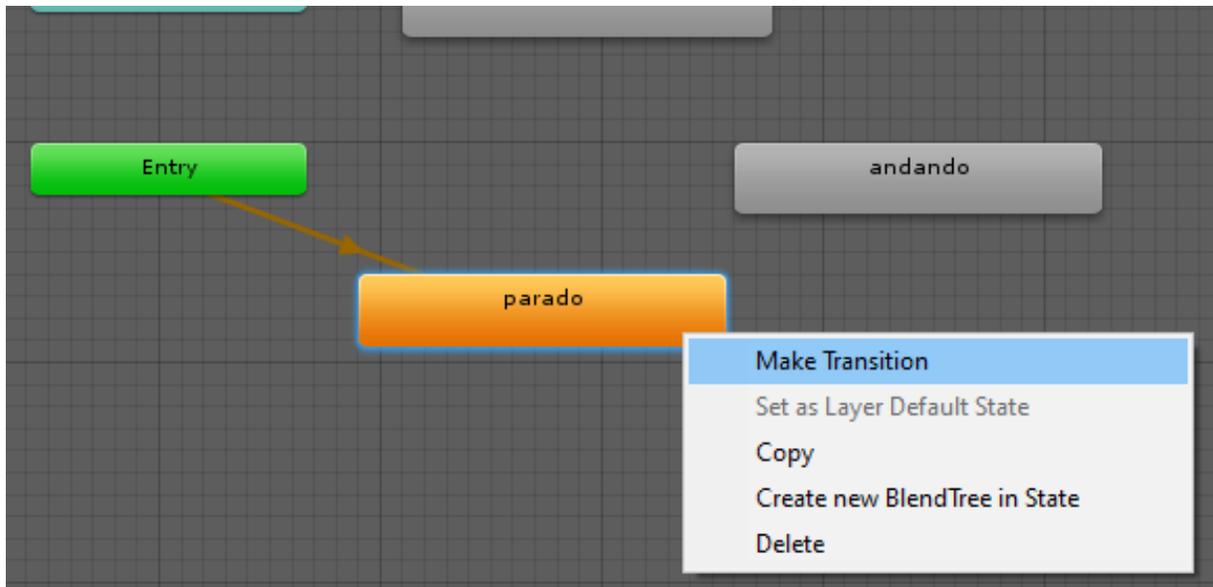
As transições ocorrem quando um personagem troca de uma animação para a outra.

Exemplo. Ele está parado respirando(Animação de idle);

- Quando o jogador movimentar o personagem há uma transição de parado (idle) para andando(run)

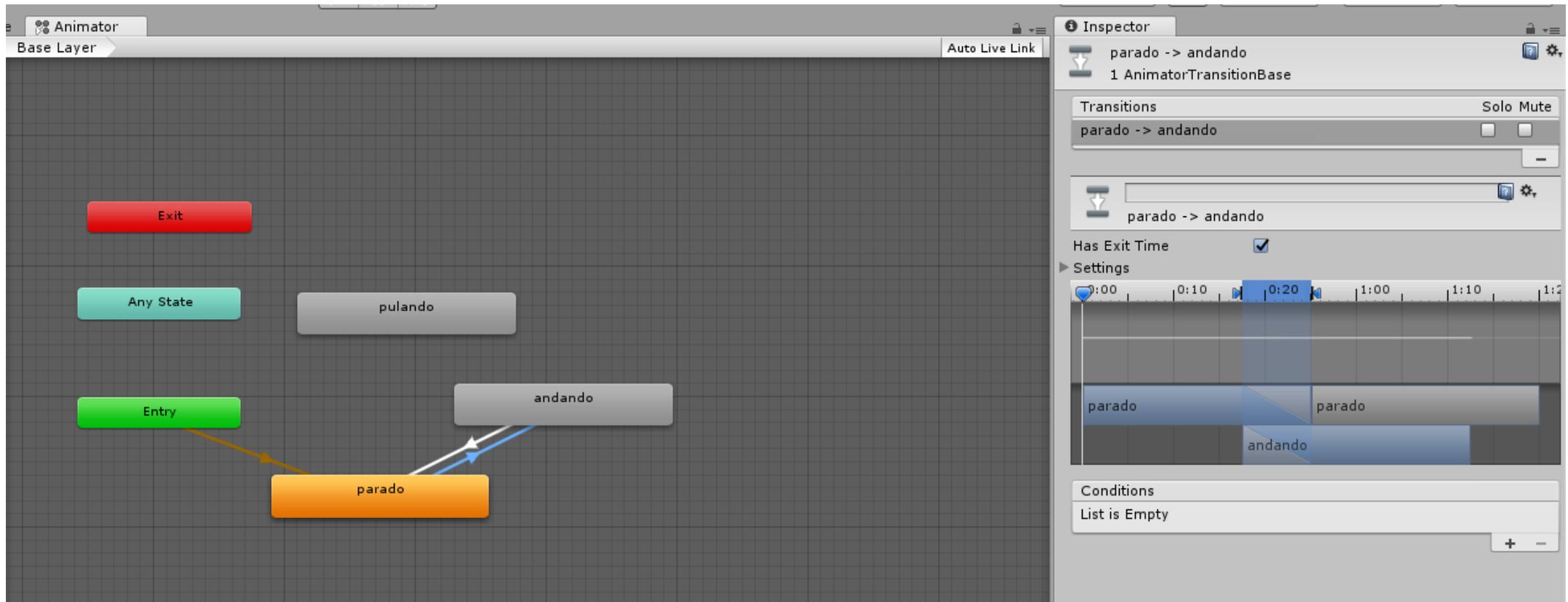
O próximo slide apresenta a criação das transições entre parado e andando.

- Click com o botão direito sobre a animação desejada e escolha a opção: “make transition”, faça de andando para parado e de parado para andando

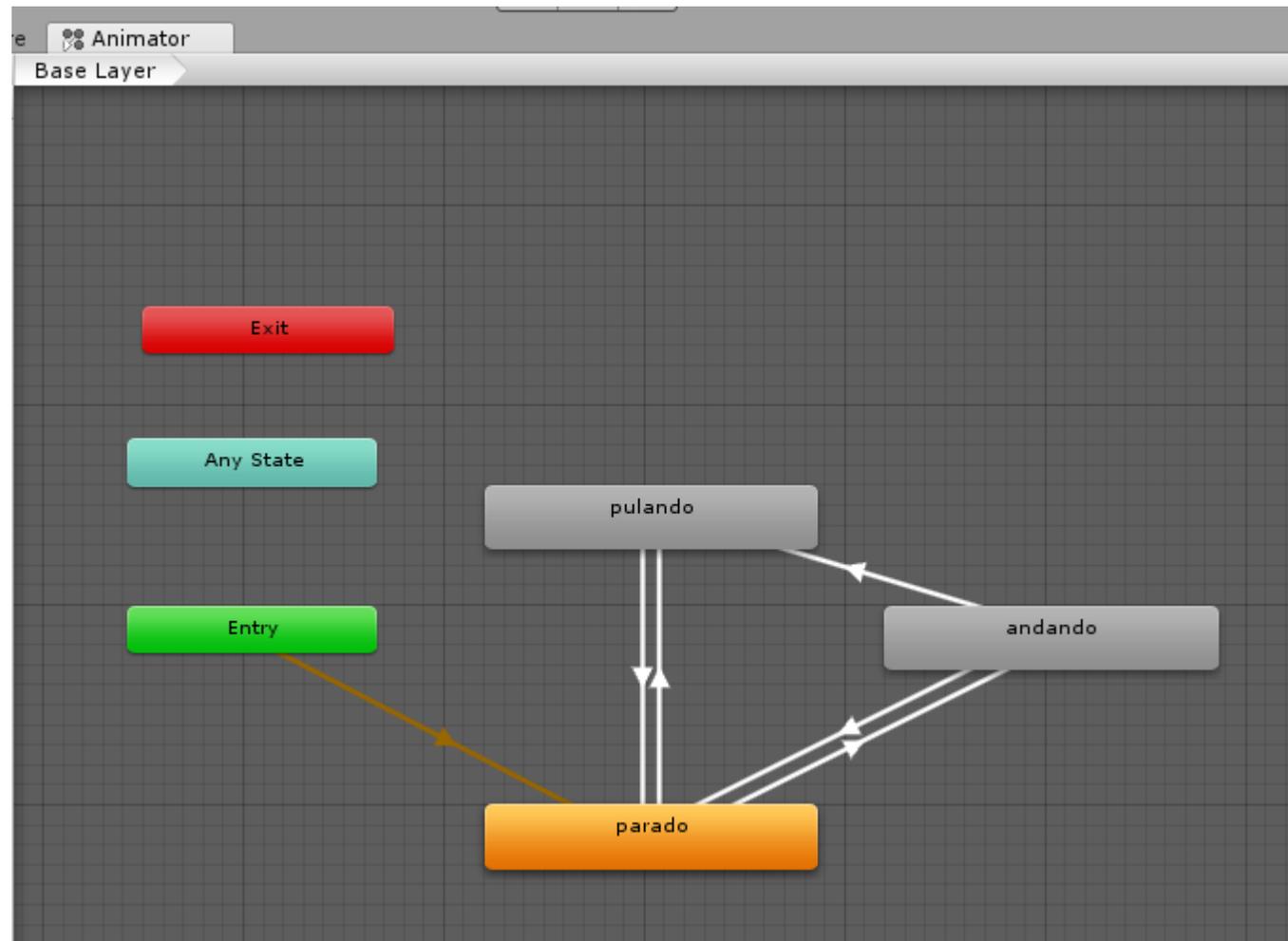


# Condições

- As condições são estruturas que permitem alternar entre as animações conforme o valor de variáveis.



# Configure a transição de andando para pulando



# Tempo de transição entre as animações

- Para cada transição remova a propriedade “Has exit time” e diminua o tempo de transição
- Faça as mesmas configurações para todas as transições.

The image displays a Unity Animator interface. On the left, a state machine diagram shows transitions between states: 'parado' (orange), 'andando' (grey), 'pulando' (grey), 'Any State' (cyan), and 'Exit' (red). A transition from 'parado' to 'andando' is highlighted with a blue arrow. On the right, the Inspector window shows the configuration for the 'parado -> andando' transition. The 'Has Exit Time' checkbox is unchecked, and the transition duration on the timeline is set to 0:20. A red box highlights the duration value, and a blue callout box points to it with the text 'Diminua o tempo de transição'. Another blue callout box points to the 'Has Exit Time' checkbox with the text 'Desmarque'.

Inspector

parado -> andando  
1 AnimatorTransitionBase

Transitions Solo Mute  
parado -> andando

Has Exit Time

Settings

0:00 0:10 0:20 1:00 1:10 1:20

parado andando

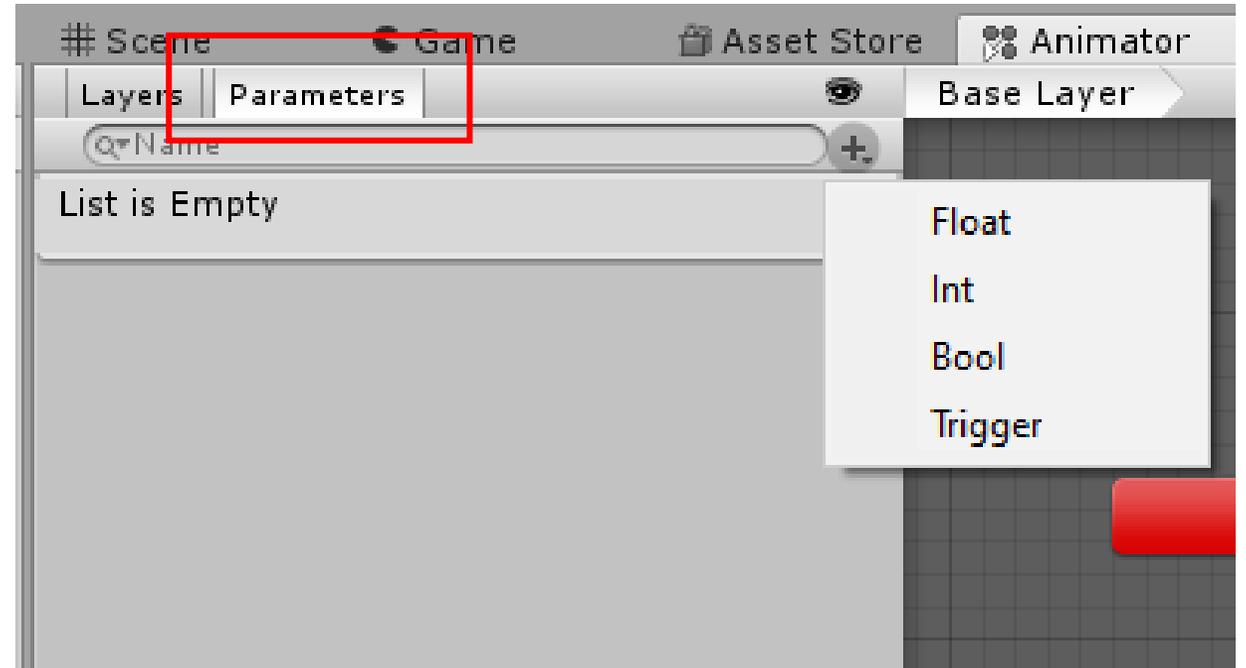
Conditions  
= andando true

Desmarque

Diminua o tempo de transição

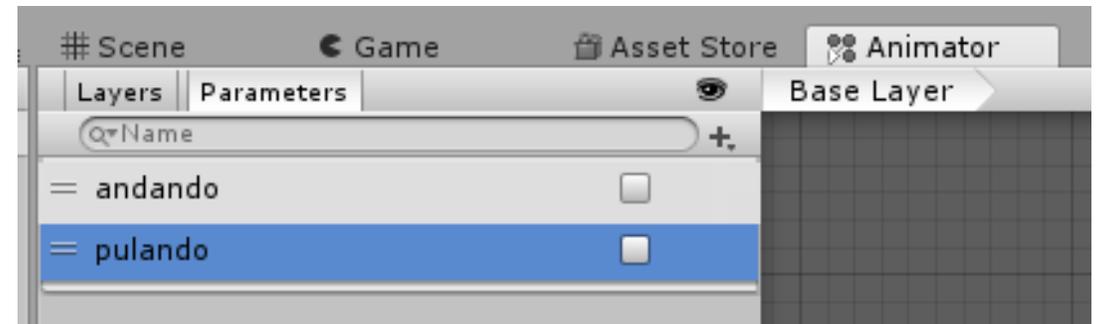
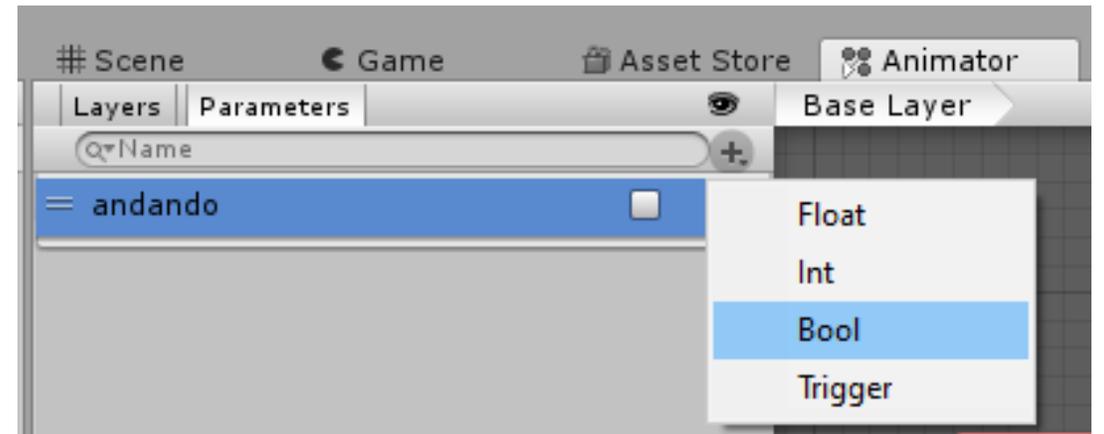
# Criar um novo parâmetro de transição

- Do lado esquerdo da janela do animator existe a ferramenta de criação e manipulação de parâmetros (**Parameters**).
- Click no botão de “+” para criar um novo parâmetro do tipo bool.
- Crie o parâmetro “andando” do tipo Bool



# Crie o parâmetro pulando do tipo bool

- Crie também o parâmetro pulando
- Serão duas variáveis de parâmetro, a variável andando e a variável pulando



# Condição: andando==true.

- Se **andando** é igual a **true** é realizada a transição entre a animação de parado para andando.

The image shows the Unity Animator interface. On the left, the Hierarchy view displays several states: 'Exit' (red), 'Any State' (cyan), 'Entry' (green), 'pulando' (grey), 'andando' (grey), and 'parado' (orange). A transition arrow points from 'parado' to 'andando'. On the right, the Inspector window shows the selected transition 'parado -> andando'. The 'Conditions' section is expanded, showing the condition '= andando' set to 'true'. A blue box with an arrow points to this condition, labeled 'Condição de transição'. Another blue box with an arrow points to the transition arrow in the Hierarchy, labeled 'Observe a seta da transição'. The 'Settings' section shows a timeline with a blue bar for the transition, and the 'Has Exit Time' checkbox is checked.

Condição de transição

Observe a seta da transição

Condição:  
andando==true.

- Se **andando** igual a **false** é realizada a transição entre a animação de andando para parado.

Observe a seta da transição

The image shows the Unity Animator interface. On the left, the state machine is visible with states: Exit (red), Any State (cyan), pulando (grey), andando (grey), and parado (orange). Transitions are shown with arrows: a blue arrow from 'parado' to 'andando', and a white arrow from 'andando' to 'parado'. A blue callout box points to the transition arrow with the text 'Observe a seta da transição'. On the right, the Inspector panel shows the selected transition 'andando -> parado'. The 'Conditions' section is expanded, showing the condition '= andando' with a dropdown set to 'false'. A blue callout box points to this condition with the text 'Condição de transição'. The 'Settings' section shows a timeline with a blue bar for the transition from 0:00 to 0:20.

Condição de transição

Faça para todas as transições

# Programação.

```
public class Personagem : MonoBehaviour
{
    Animator PersonagemAnimator;
```

# Start()

```
void Start(){  
    //...  
    PersonagemAnimator = gameObject.GetComponent<Animator> ();  
    //...  
}
```

# Update

```
void Update(){  
    MovimentoHorizontalFlip();  
    PuloDuplo();  
}
```

# MovimentoHorizontal()

Modifica o valor da variável de animação “andando” para true para que aconteça a animação de andando

```
void MovimentoHorizontalFlip(){  
    // Obtém a direção horizontal a partir do input do jogador (teclado ou controle)  
    // Input.GetAxis("Horizontal") retorna um valor entre -1 e 1:  
    // -1 para esquerda, 1 para direita, 0 se nenhuma tecla do eixo horizontal estiver sendo pressionada  
    DirecaoHorizontal = Input.GetAxis ("Horizontal");  
  
    // Calcula a velocidade no eixo X multiplicando a direção pela velocidade máxima  
    VelocidadeX = VelocidadeHorizontalMaxima * DirecaoHorizontal;  
  
    // Mantém a velocidade atual no eixo Y (para não interferir na gravidade ou pulo)  
    VelocidadeY = CorpoRigidoPersonagem.velocity.y;  
  
    // Cria um novo vetor de velocidade com os valores calculados  
    VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);  
  
    if (DirecaoHorizontal != 0) {  
        PersonagemAnimator.SetBool ("andando", true);  
    } else {  
        PersonagemAnimator.SetBool ("andando", false);  
    }  
  
    // Aplica o vetor de velocidade ao Rigidbody2D para mover o personagem  
    CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;  
    if (DirecaoHorizontal < 0) {  
        Rerender.flipX = true;  
    } else if (DirecaoHorizontal > 0) {  
        Rerender.flipX = false;  
    }  
}
```

Entra se o personagem está se movendo

Entra se o personagem não está se movendo

# Modifique o valor dos parâmetros de transição

```
void PuloDuplo(){  
    if (Input.GetKeyDown(KeyCode.Space) && qtdPulos<2)  
    {  
        personagemAnimator.SetBool ("pulando", true);  
        personagemAnimator.SetBool ("andando", false);  
        float y = velocidadePulo;  
        float x = personagemRigidbody.velocity.x;  
        Vector2 movimento = new Vector2(x, y);  
        personagemRigidbody.velocity = movimento;  
        qtdPulos++;  
    }  
}
```

Modifica o valor da variável de animação “pulando” para true e a variável “andando” para false.

Essa modificação permite que aconteça a transição ente as animações

## OnCollisionEnter2D(Collision2D objetoTocado)

```
void OnCollisionEnter2D(Collision2D objetoTocado) {  
    //...  
    personagemAnimator.SetBool ("pulando", false);  
    personagemAnimator.SetBool ("andando", true);  
    //...  
}  
}
```

Se o personagem toca em qualquer elemento com colisor, o valor das variáveis de animação são modificadas.

Se o personagem toca em algo ele não está pulando e volta a andar.