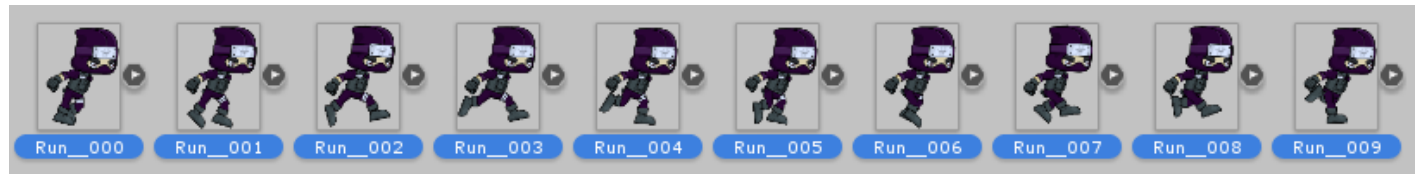


ANIMAÇÕES

Prof. Me. Hélio Esperidião

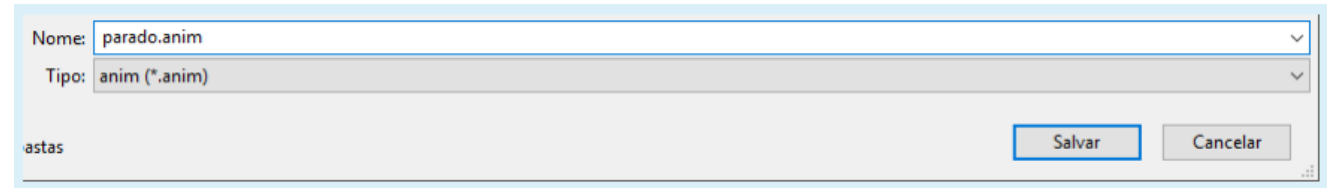
Sprite

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



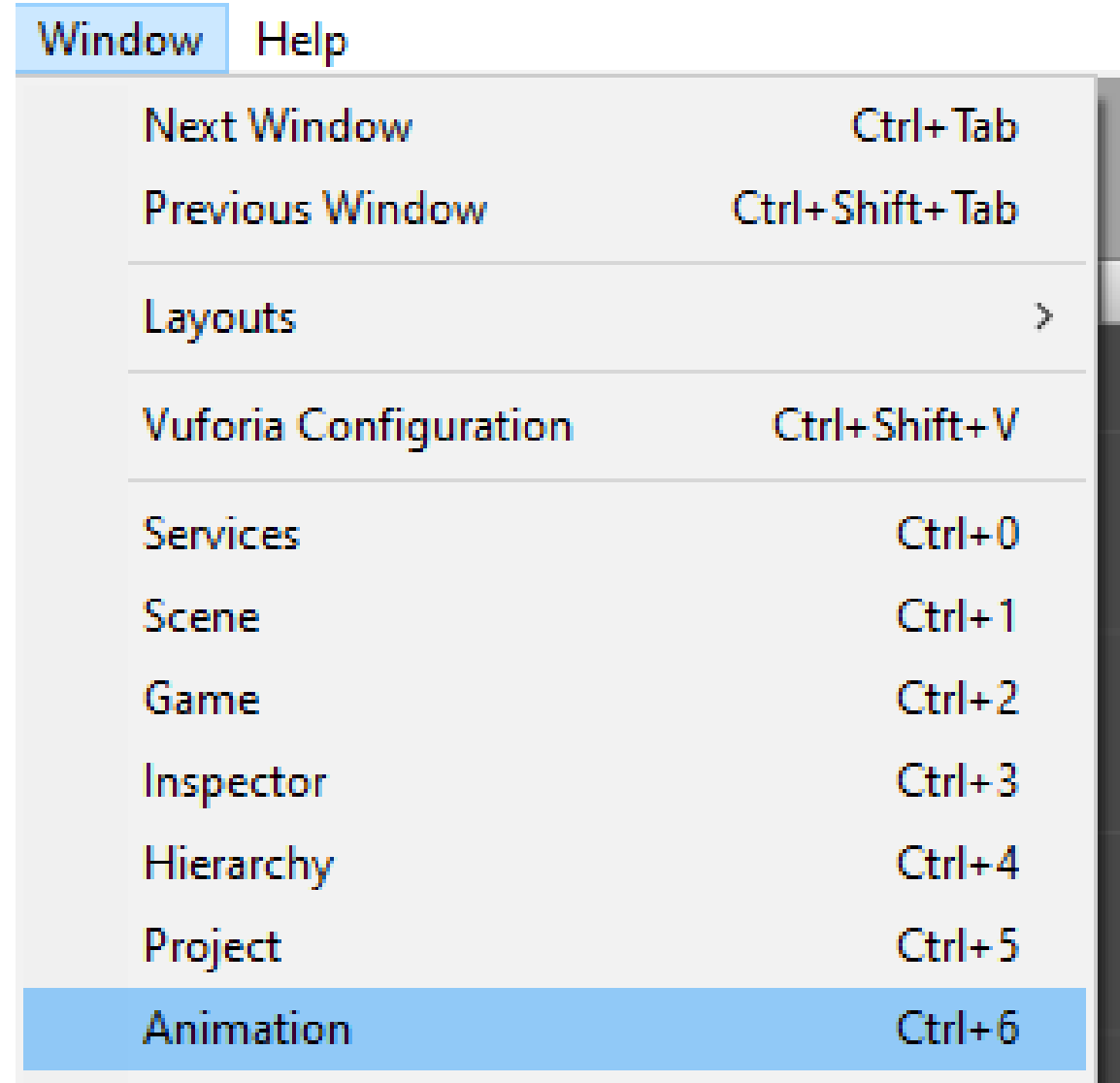
Selecione os assets que compõem uma animação.

- 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”



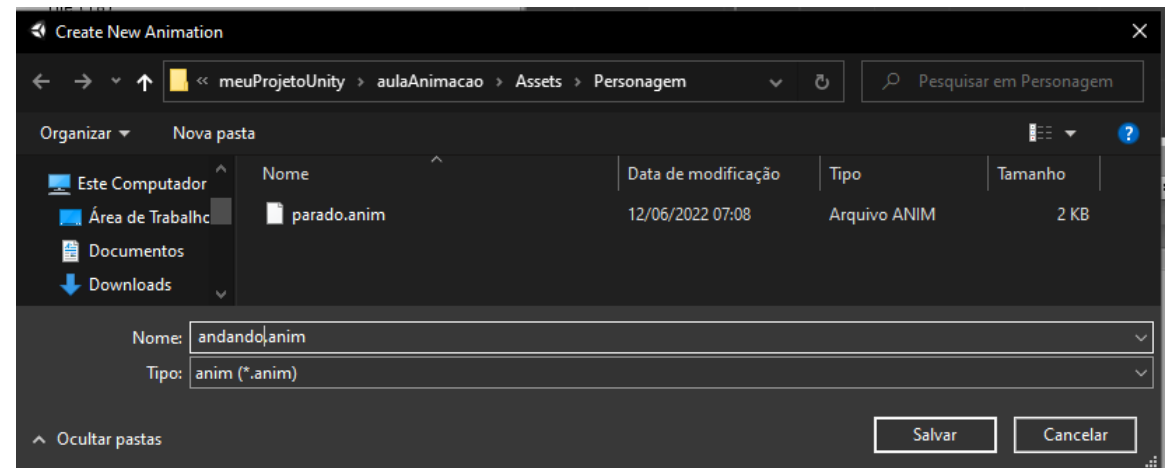
Abra a janela de animation

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



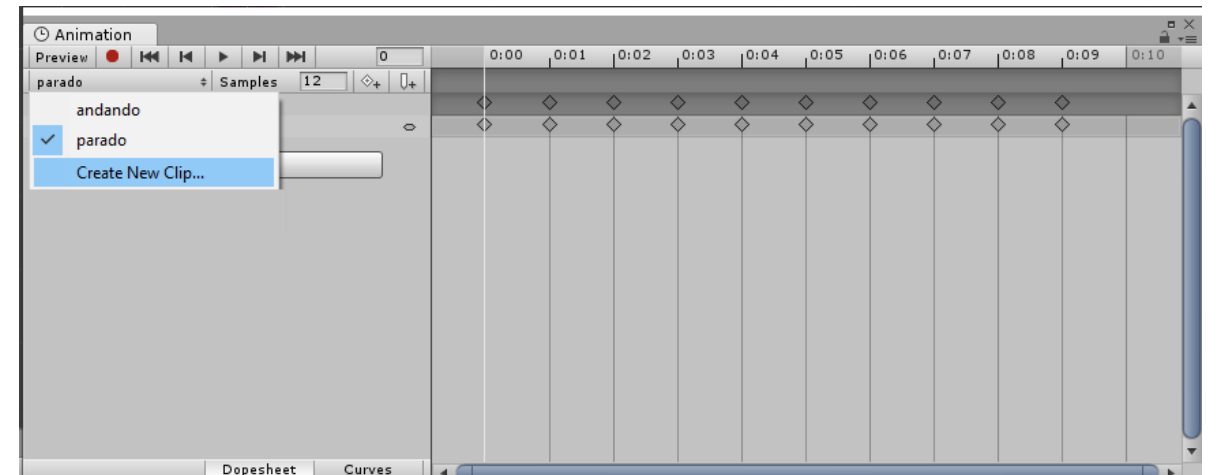
Criar novo Clip (Animação)

- 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.



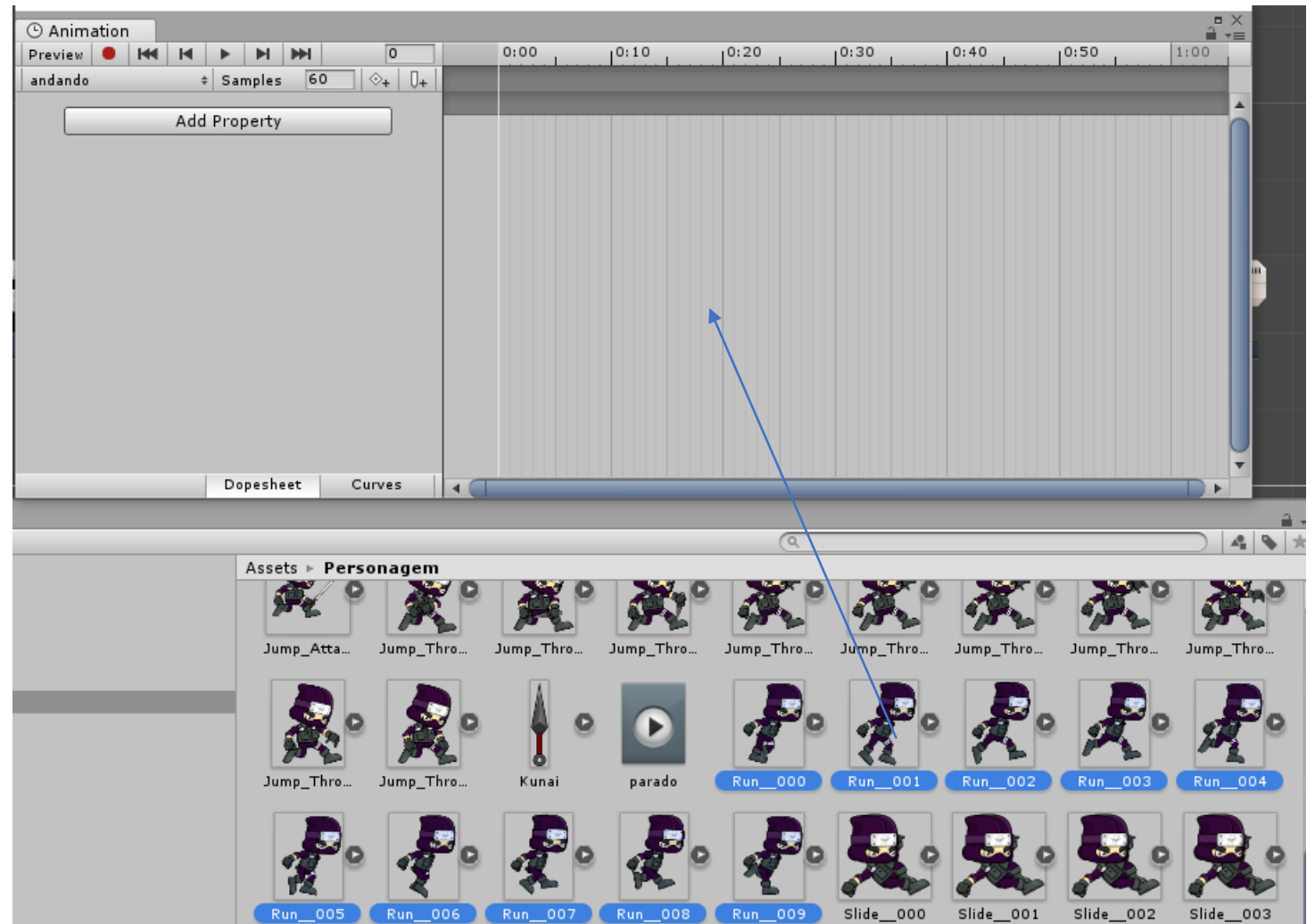
Click em parado e selecione “create new clipe”

- 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



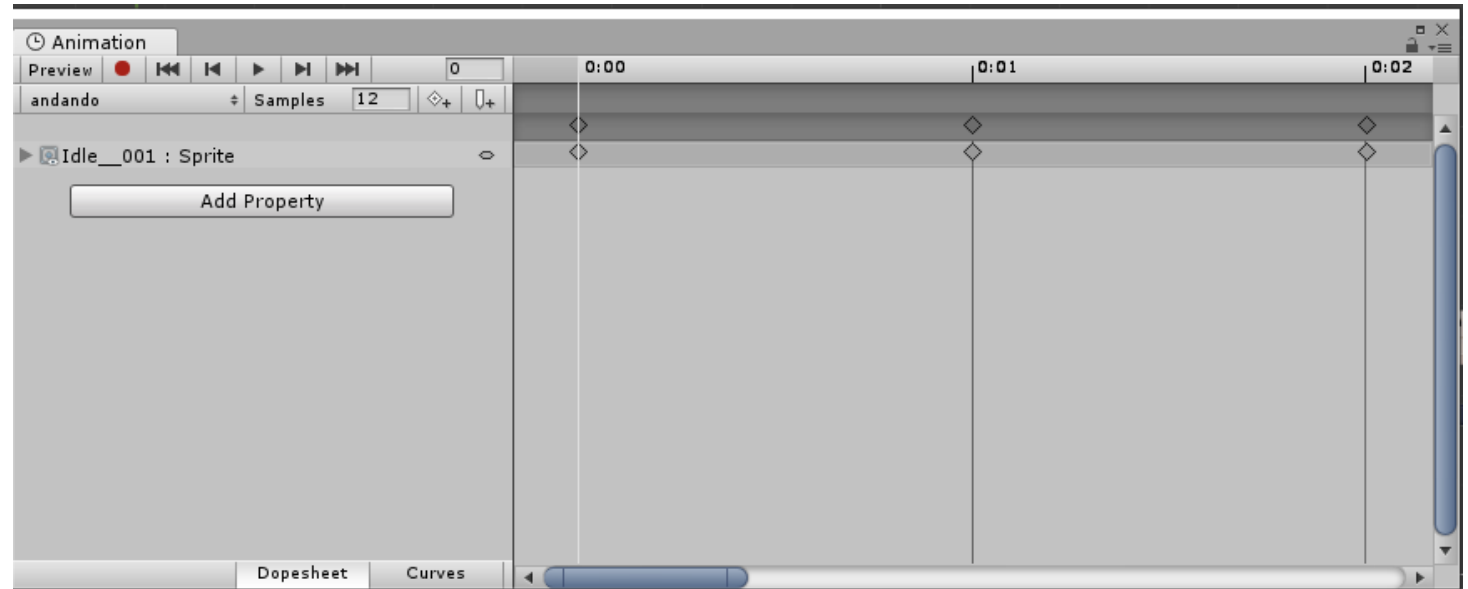
Criar nova animação

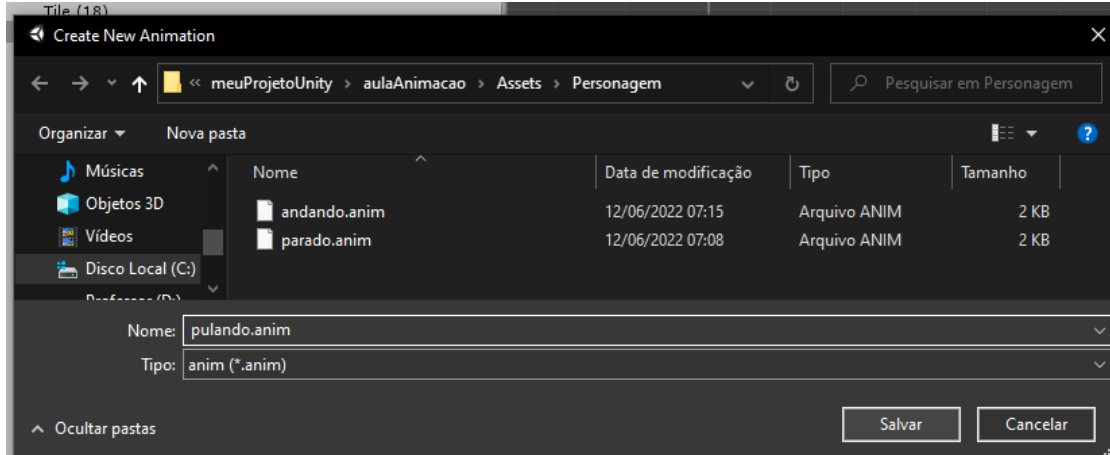
- Selecione todas as imagens da animação
- Arraste para dentro da time-line do Animation



Samples

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



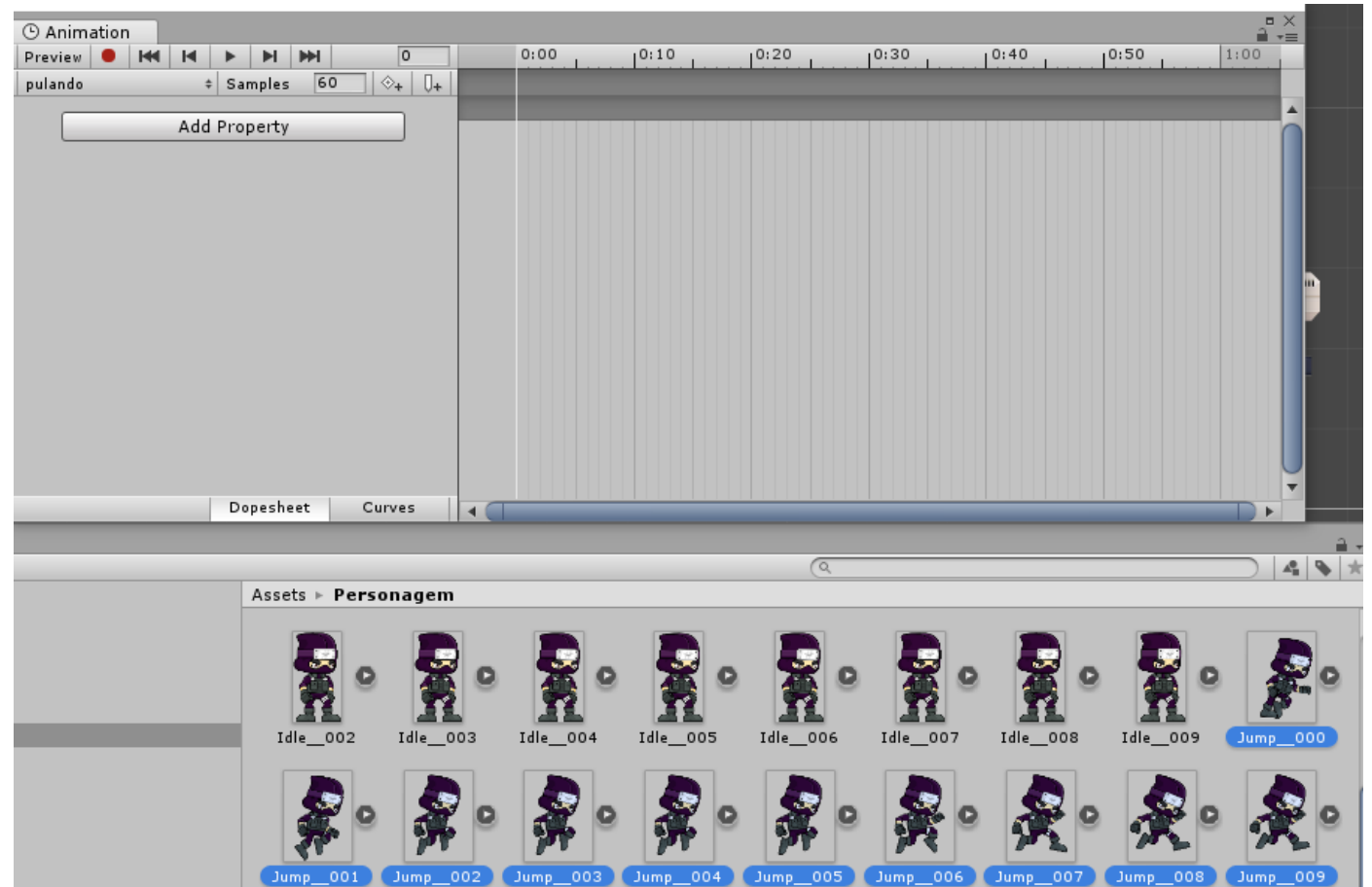


Crie outra animação de **pulando**

- 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

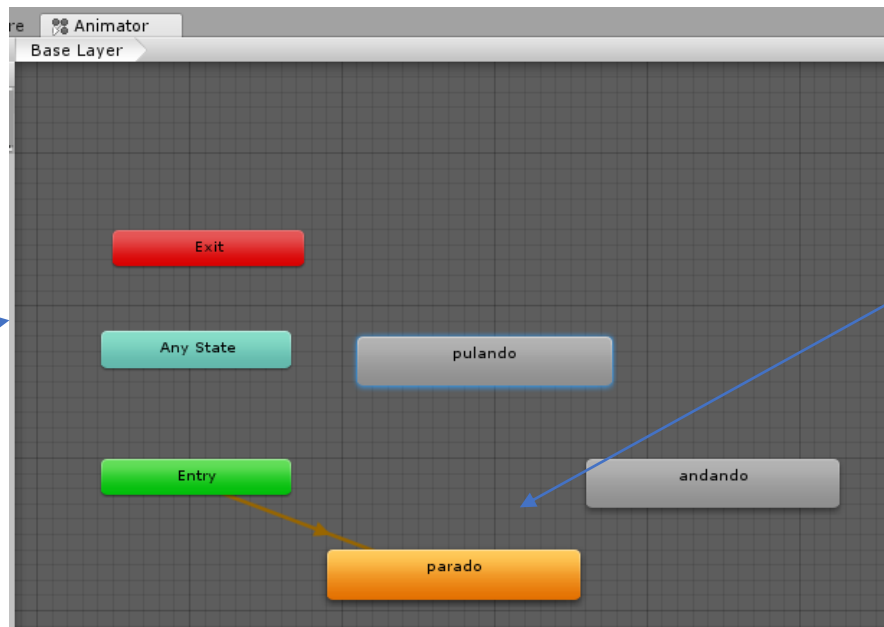
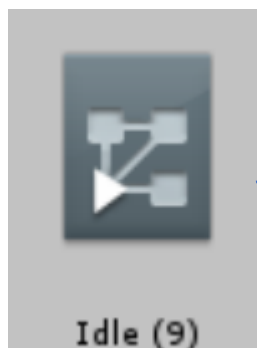
- Ao criar uma animação também é criado automaticamente um animador,
- Ele pode ser encontrado em meio a seus assets por meio da imagem ao lado



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

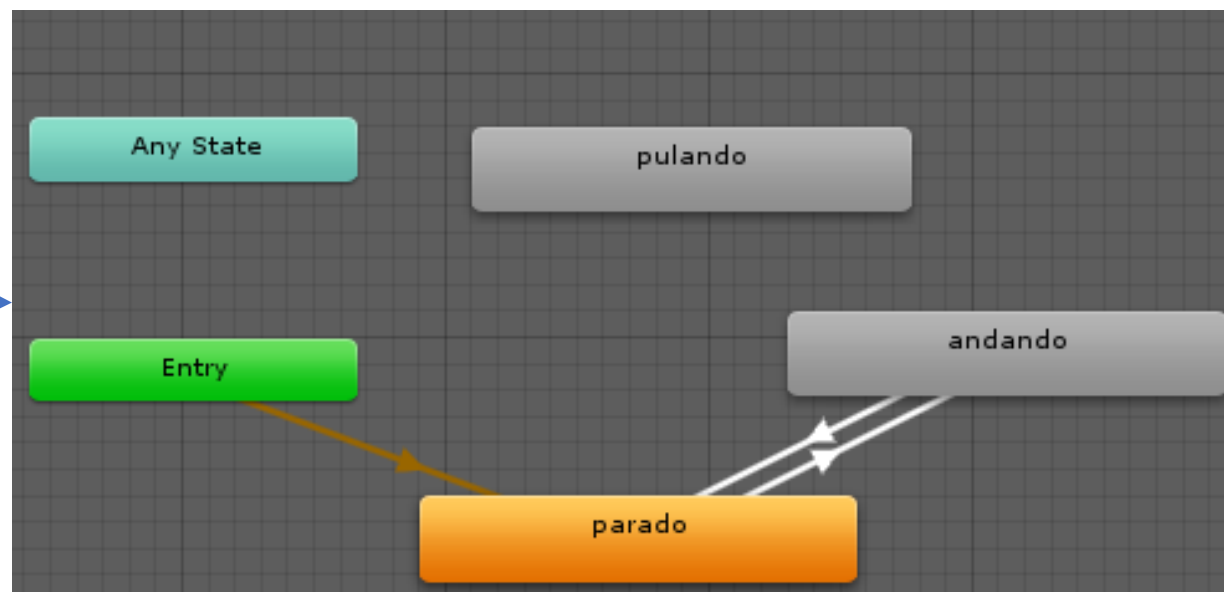
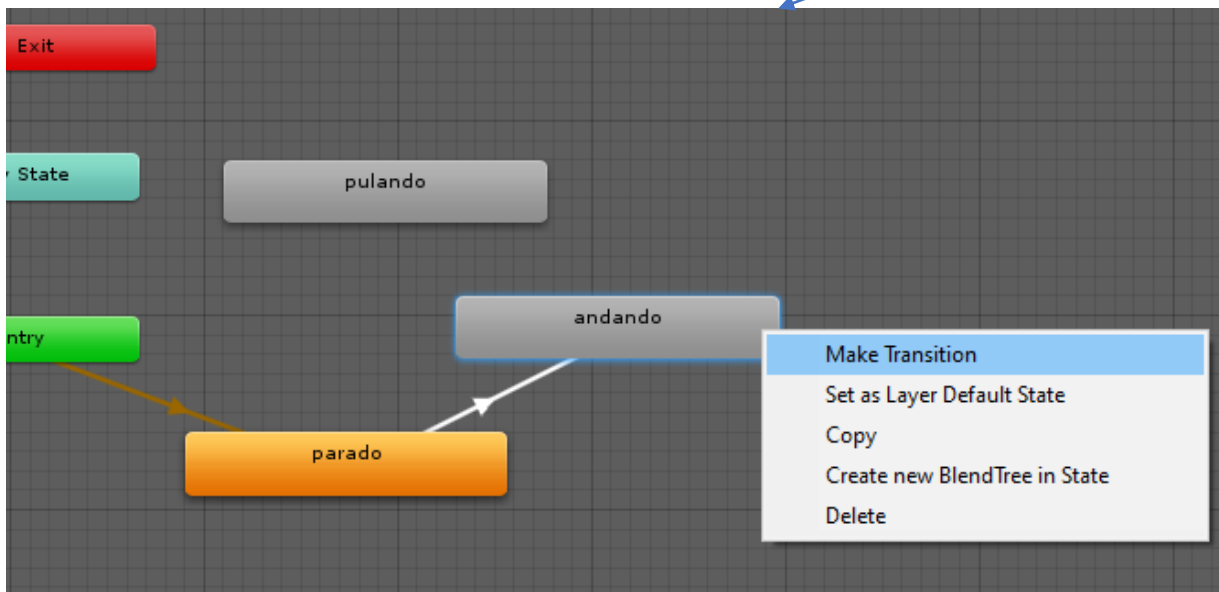
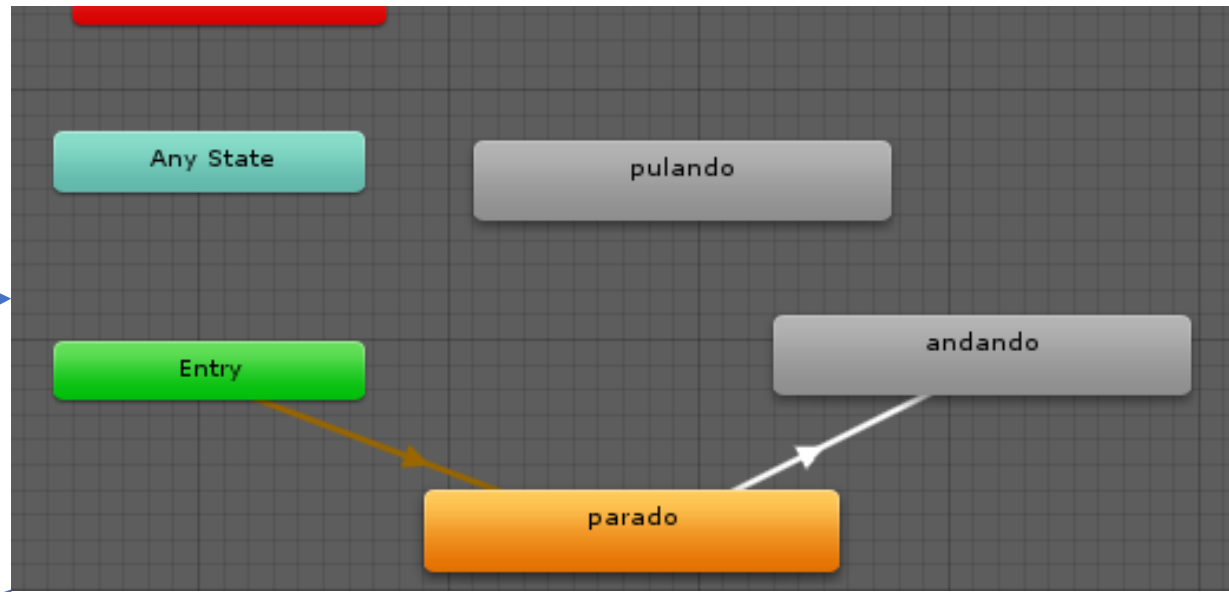
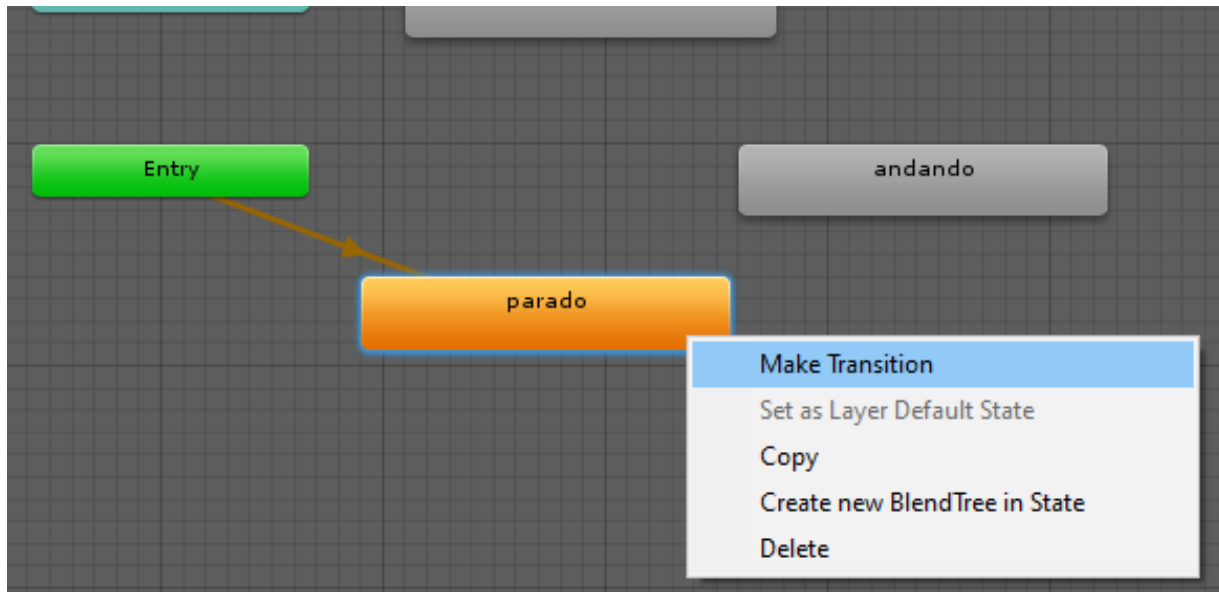
As transições ocorrem quando um personagem troca de uma animação para a outra.

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

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

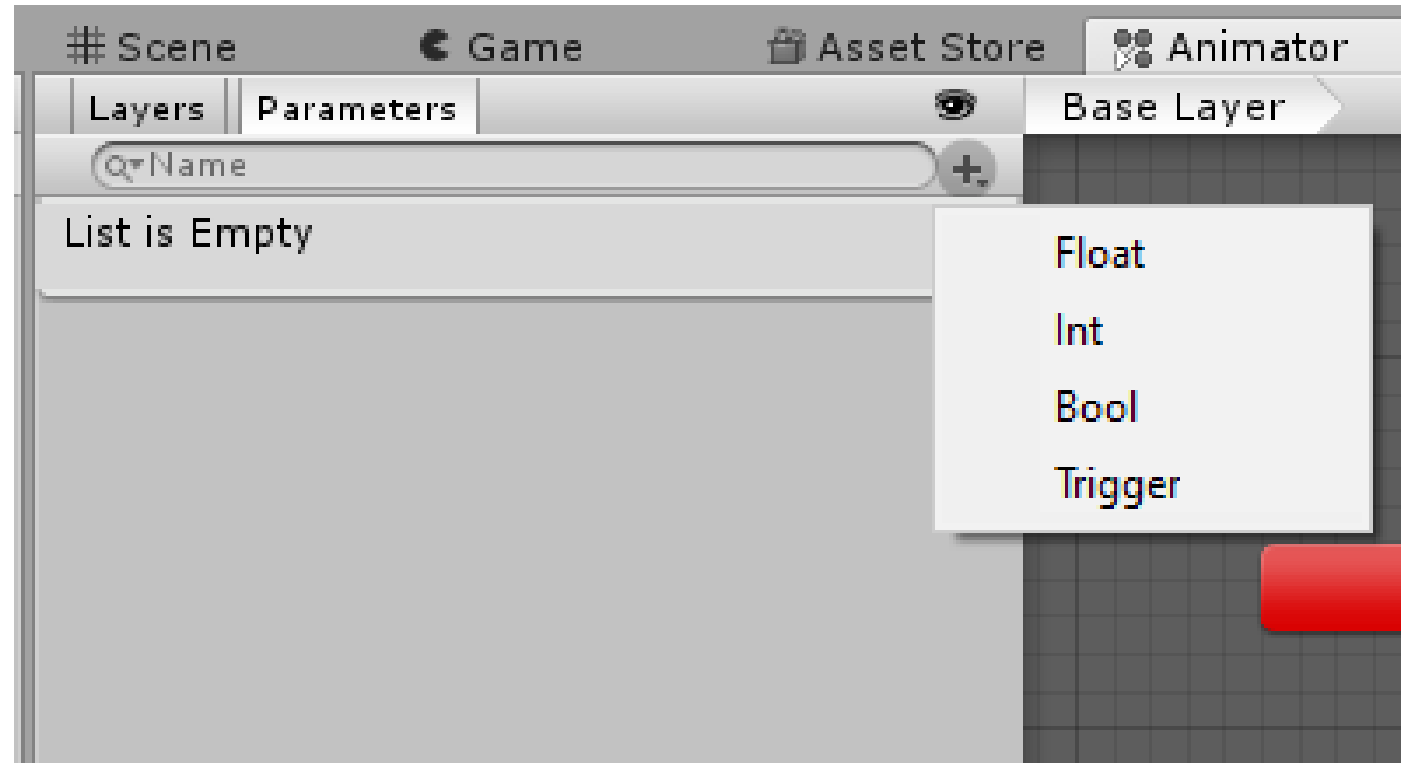
O próximo slide apresenta a criação das transições entre parado e andando.

- 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



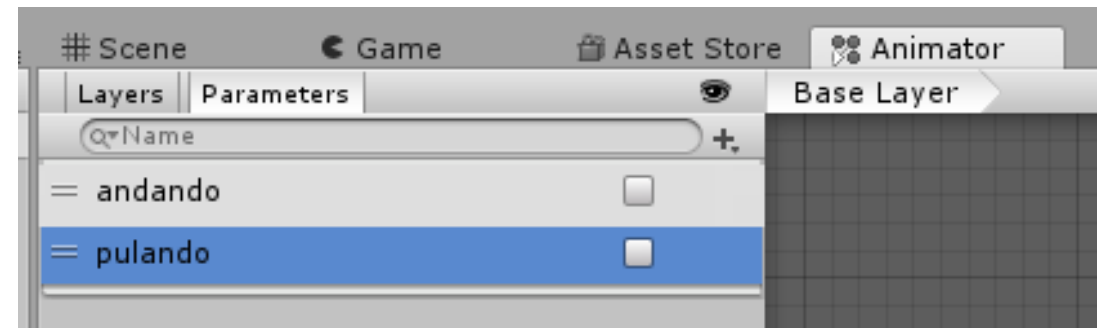
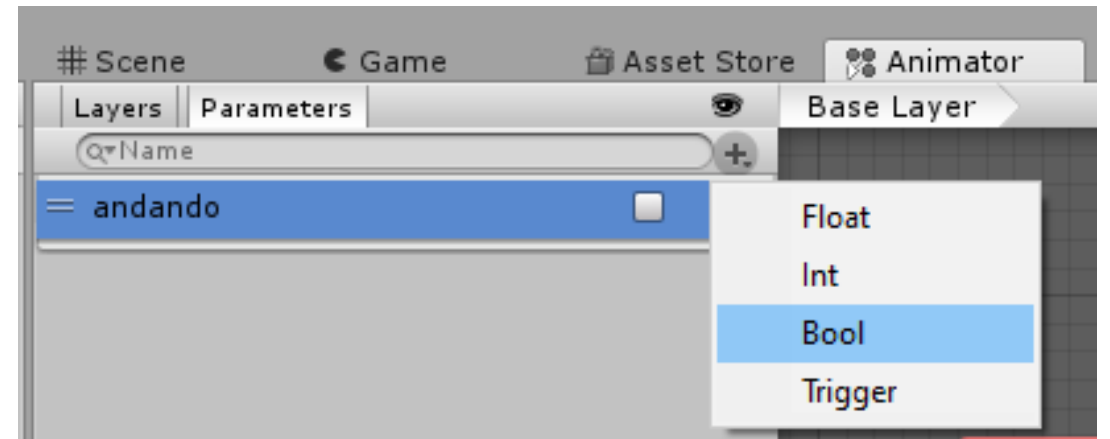
Criar um novo parâmetro de transição

- 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



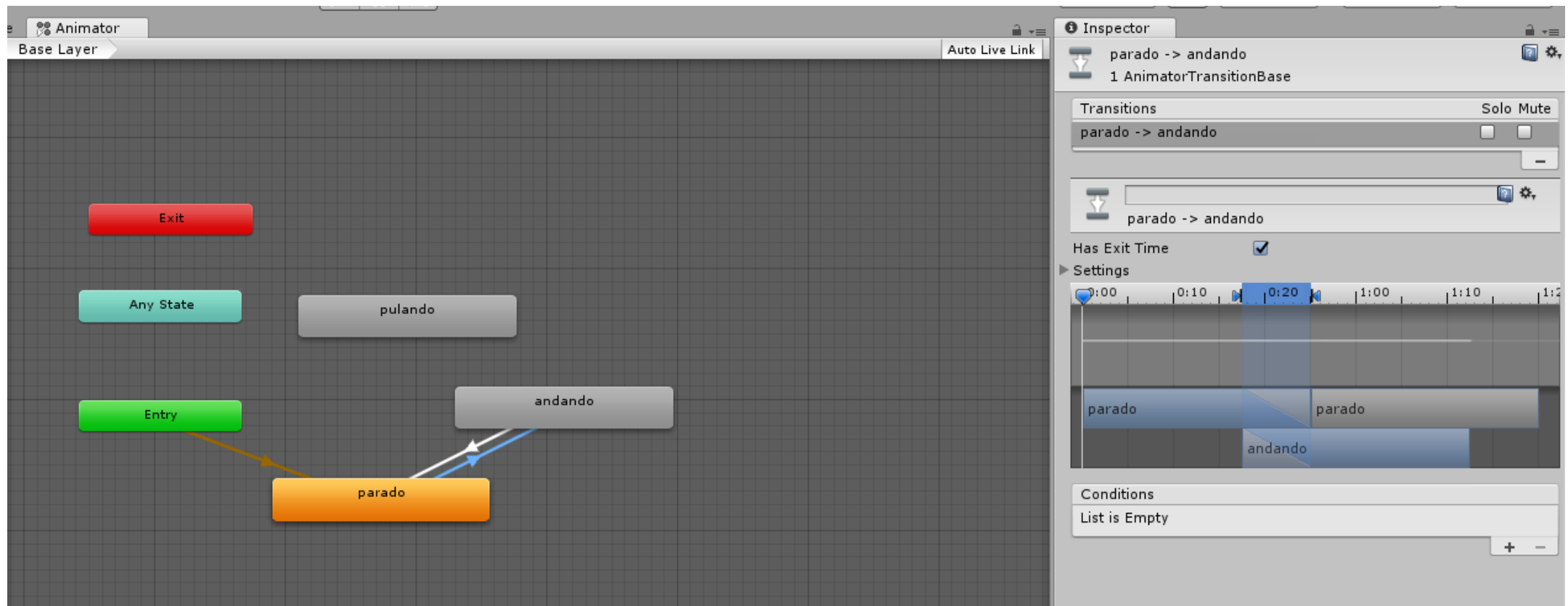
Crie o parâmetro pulando do tipo bool

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



Condições

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



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 is the Animator Controller window with a state machine graph. It includes states: 'Exit' (red), 'Any State' (cyan), 'Entry' (green), 'parado' (orange), 'pulando' (grey), and 'andando' (grey). Transitions are shown with arrows: a blue arrow from 'parado' to 'andando', and a white arrow from 'andando' to 'parado'. On the right is the Inspector window for the selected transition 'parado -> andando'. It shows the transition name, 'Solo Mute' checkbox, 'Has Exit Time' checked, and a 'Settings' section with a timeline. The 'Conditions' section at the bottom shows the condition '= andando' set to 'true'. A blue arrow points from the 'true' value to a blue callout box labeled 'Condição de transição'. Another blue arrow points from the transition arrow in the state machine to a blue callout box labeled 'Observe a seta da transição'.

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, a state machine diagram displays states: Exit (red), Any State (cyan), pulando (grey), andando (grey), and parado (orange). Transitions are shown with arrows: a blue arrow from 'andando' to 'parado', and a white arrow from 'parado' to 'andando'. On the right, the Inspector panel shows the selected transition 'andando -> parado'. The 'Conditions' section is set to 'andando' with a value of 'false'. The 'Settings' section shows a timeline with a blue bar for the 'andando' state and a grey bar for the 'parado' state, with a transition point at 0:20. A blue box with an arrow points to the 'false' value in the conditions field.

Condição de transição

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 shows the Unity Animator interface. On the left, the state machine graph displays states: 'Exit' (red), 'Any State' (cyan), 'parado' (orange), 'pulando' (grey), 'andando' (grey), and 'Entry' (green). Transitions are shown between 'parado' and 'andando'. On the right, the Inspector panel shows the configuration for the transition 'parado -> andando'. The 'Has Exit Time' checkbox is unchecked, and the transition duration on the timeline is set to 0:20. Two blue callout boxes with arrows point to these elements: 'Desmarque' points to the 'Has Exit Time' checkbox, and 'Diminua o tempo de transição' points to the duration marker on the timeline.

Inspector

parado -> andando
1 AnimatorTransitionBase

Transitions

parado -> andando

parado -> andando

Has Exit Time

Settings

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

parado

andando

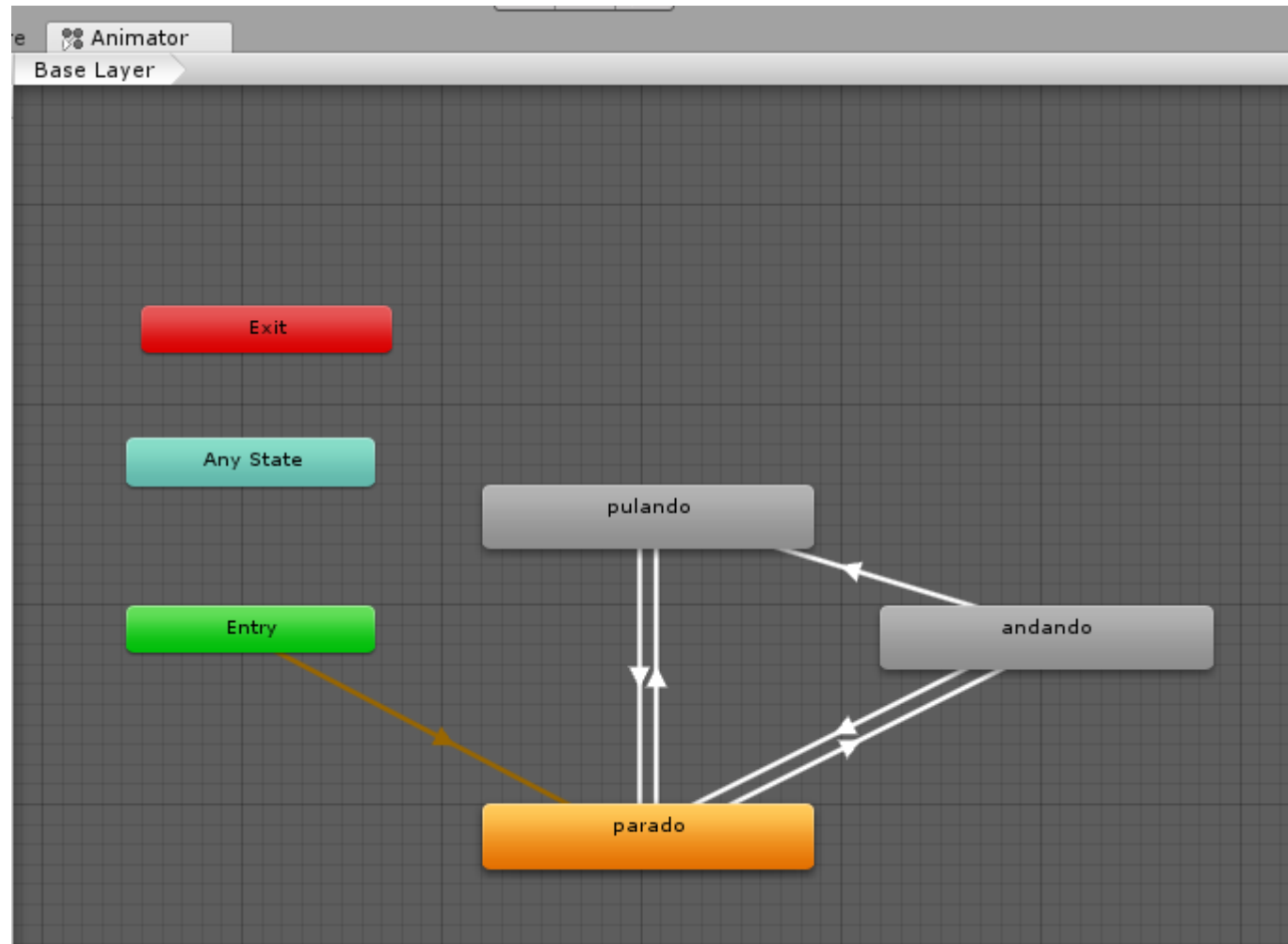
Conditions

= andando true

Desmarque

Diminua o tempo de transição

Configure a transição de andando para pulando



De parado para pulando e de parada para pulando

The image displays the Unity Animator interface. On the left, the state machine is visible in the Base Layer, showing states: Exit (red), Any State (cyan), Entry (green), parado (orange), pulando (grey), and andando (grey). Transitions are shown with arrows: a blue arrow from parado to pulando, a white arrow from pulando to parado, a white arrow from parado to andando, and a white arrow from andando to parado. On the right, the Inspector panel shows the selected transition 'parado -> pulando' (1 AnimatorTransitionBase). The Transitions section shows 'parado -> pulando' with Solo and Mute checkboxes. The Settings section includes a timeline from 0:00 to 1:20, with a blue bar for 'parado' from 0:00 to 0:15 and a blue bar for 'pulando' from 0:15 to 0:20. The Conditions section shows '= parado' set to 'true'.

Programação.

```
public class Personagem : MonoBehaviour
{
    Animator personagemAnimator;
    Rigidbody2D personagemRigidbody;
    SpriteRenderer personagemSpriteRenderer;
    float direcaoHorizontal;
    float velocidadeMovimento;
    float velocidadePulo;
    int qtdPulos;
```

Start()

```
void Start(){
    direcaoHorizontal = 0;
    velocidadeMovimento = 5;
    velocidadePulo = 5;
    qtdPulos = 0;

    personagemRigidbody = gameObject.GetComponent<Rigidbody2D> ();
    personagemSpriteRenderer = gameObject.GetComponent<SpriteRenderer> ();
    personagemAnimator = gameObject.GetComponent<Animator> ();
    personagemRigidbody.freezeRotation = true;
}
```


Update

```
void Update(){  
    MovimentoHorizontal();  
    //PuLo();  
    PuloDuplo();  
}
```

MovimentoHorizontal()

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

```
void MovimentoHorizontal(){
    direcaoHorizontal = Input.GetAxis ("Horizontal");
    if (direcaoHorizontal != 0) {
        float x = direcaoHorizontal * velocidadeMovimento;
        float y = personagemRigidbody.velocity.y;
        Vector2 movimento = new Vector2 (x, y);
        personagemRigidbody.velocity = movimento;
        personagemAnimator.SetBool ("andando", true);
    } else {
        personagemAnimator.SetBool ("andando", false);
    }
    if (direcaoHorizontal > 0) {
        personagemSpriteRenderer.flipX = false;
    }
    if (direcaoHorizontal < 0) {
        personagemSpriteRenderer.flipX = true;
    }
}
```

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);  
    qtdPulos = 0;  
}  
}
```

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.