

Unity

Platformer 2D: Animação 2D

Animator

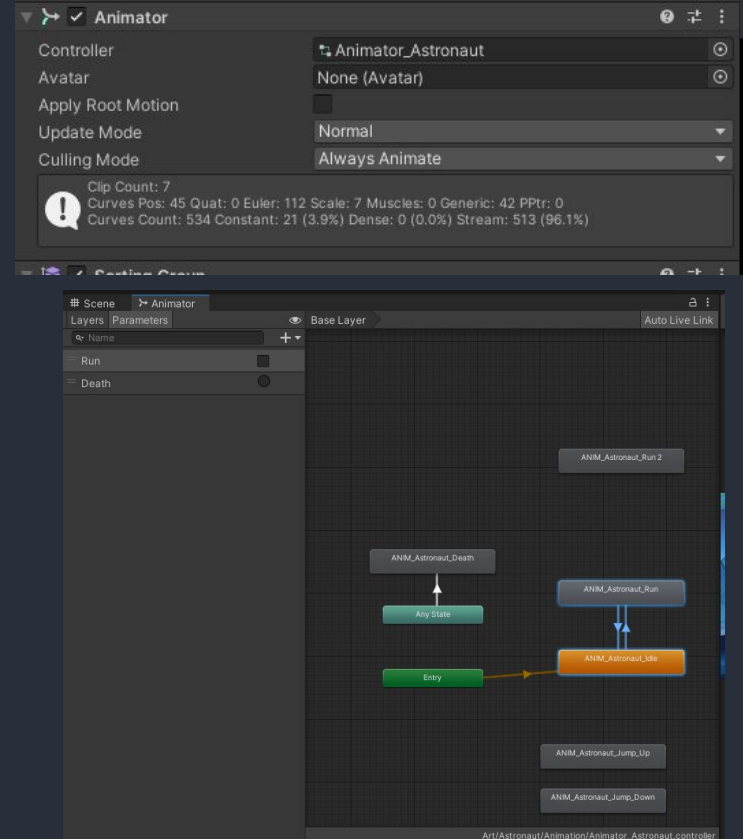


Animator

Podemos utilizar 2 tipos de sistemas de animação na Unity:

Animator (mais novo e robusto)
Animation (mais antigo e simples)

*Vamos entender melhor um pouco sobre o **Animator**.*



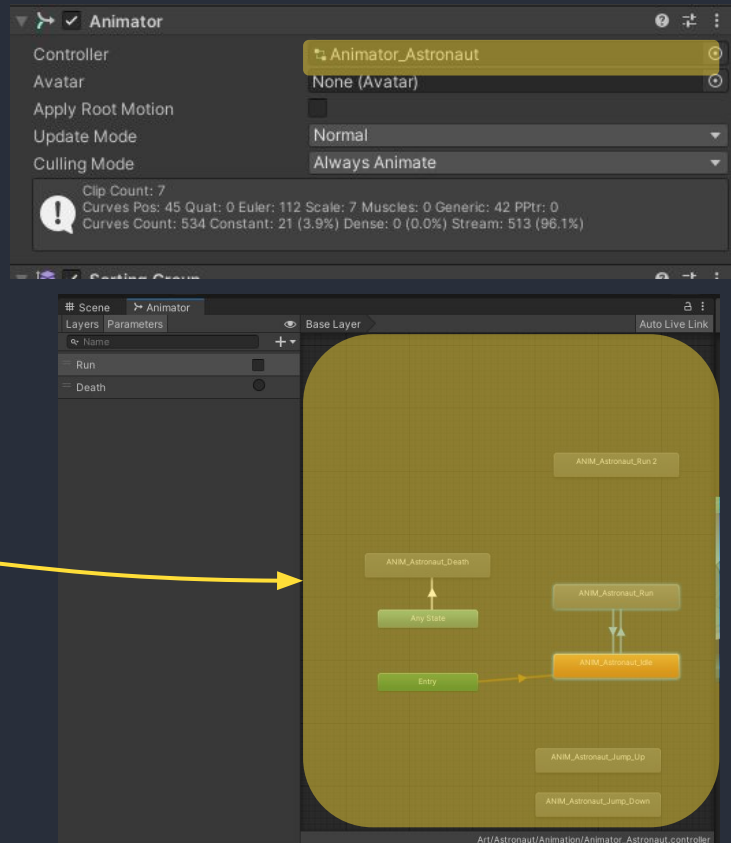
Animator

*O Animator trabalha sempre utilizando um **controlador**.*

*Para acessar a janela com detalhes do controlador, basta apenas um duplo clique nele, e a janela **Animator** abrirá.*

Nela, é possível visualizar todas as animações presentes, assim como suas possíveis transições.

Para adicionar animações que não estão presentes no animator, basta arrastar a animação aqui:



Animator

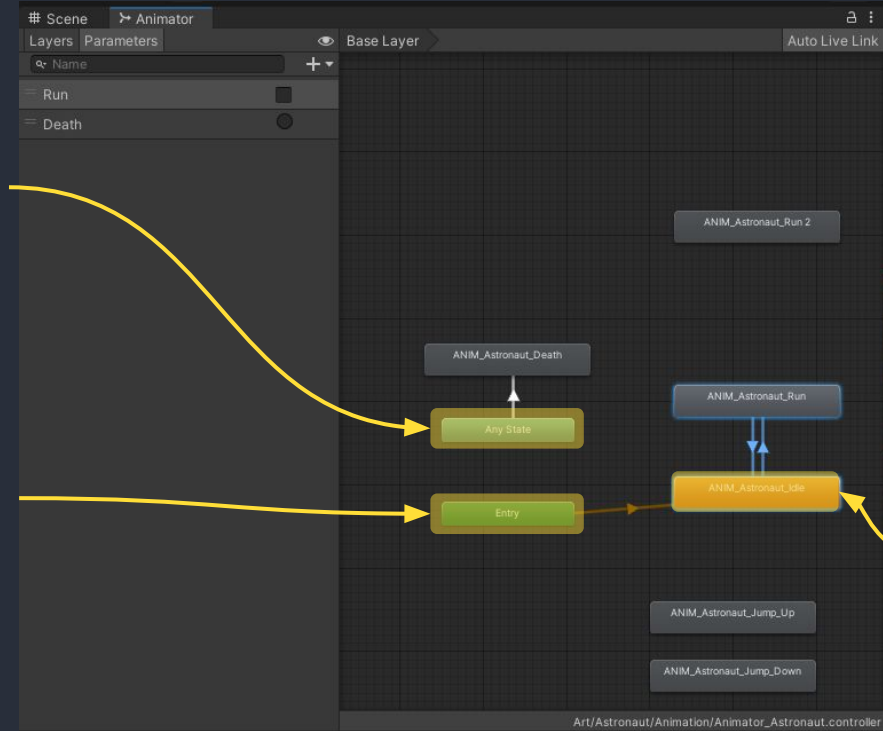
Existem sempre 3 estados necessários no controlador:

Any State

Significa que uma transição feita neste estado pode acontecer em **qualquer** animação que estiver tocando.

Entry

A entrada do animator. A primeira coisa que vai acontecer após o animator começar a tocar.



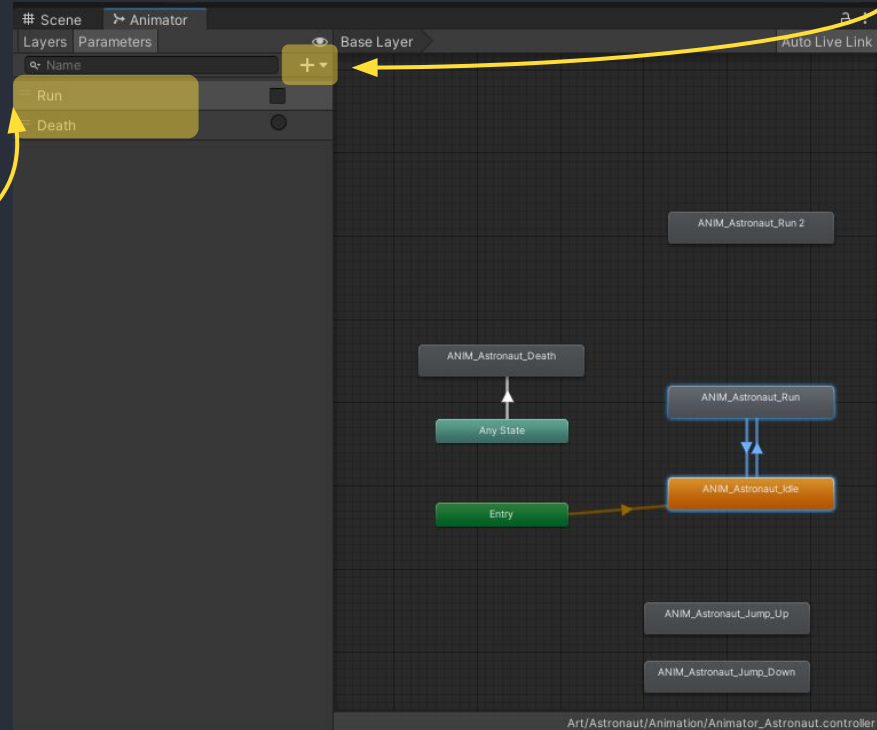
Layer Default State

É a animação que fica em **laranja**. Ela vai ser a animação "padrão", a primeira a ser tocada após o animator começar.

Animator

Para adicionarmos parâmetros para as transições, basta adicionarmos aqui

Esses são os parâmetros que esse controlador tem já criados agora.



Animator

Os parâmetros podem ser de 4 tipos diferentes:

*Float
Int
Bool
Trigger*

Float

*Aceita um valor **float**, e é acessado no código da seguinte maneira:*

Retorna o valor atual:

```
animator.GetFloat("Nome do Float criado no controlador");
```

Alterar valor

```
animator.SetFloat("Nome do Float criado no controlador," valor);
```

Int

*Aceita um valor **int**, e é acessado no código da seguinte maneira:*

Retorna o valor atual:

```
animator.GetInt("Nome do Int criado no controlador");
```

Alterar valor

```
animator.SetInt("Nome do Int criado no controlador", valor);
```

Animator

Os parâmetros podem ser de 4 tipos diferentes:

Float

Int

Bool

Trigger

Bool

*Aceita um valor **bool**, positivo e negativo, e é acessado no código da seguinte maneira:*

Retorna o valor atual:

```
animator.GetBool("Nome do Bool criado no controlador");
```

Alterar valor

```
animator.SetBool("Nome do Bool criado no controlador", valor);
```

Trigger

Aceita um valor de trigger. Ao ser chamado no código, ele "trigga" alguma ação uma vez, e se auto reseta após ser usado.

Retorna o valor atual:

```
animator.SetTrigger("Nome do Trigger criado no controlador");
```

Resetar valor do trigger

```
animator.ResetTrigger("Nome do Int criado no controlador");
```

Animator

Os parâmetros podem ser de 4 tipos diferentes:

Float

Int

Bool

Trigger

Exemplos

```
animator.SetTrigger("PlayIntro");
```

```
animator.GetBool("Running");  
animator.SetBool("Running", true);
```

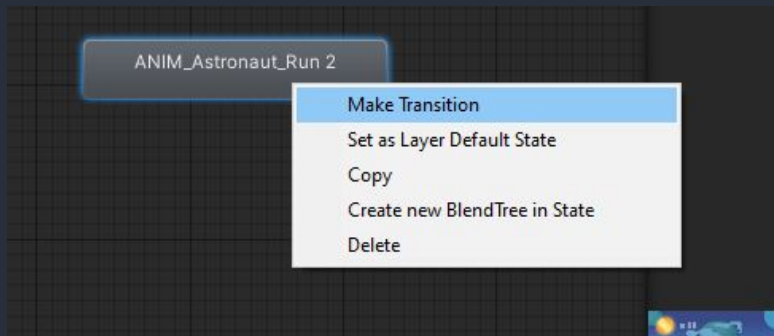
```
animator.SetFloat("Jump", 3f);
```


Animator

As transições de animações podem ser feitas da seguinte maneira:

*Clique com o botão direito na animação que deseja realizar uma transição, e selecione **Make Transition**.*

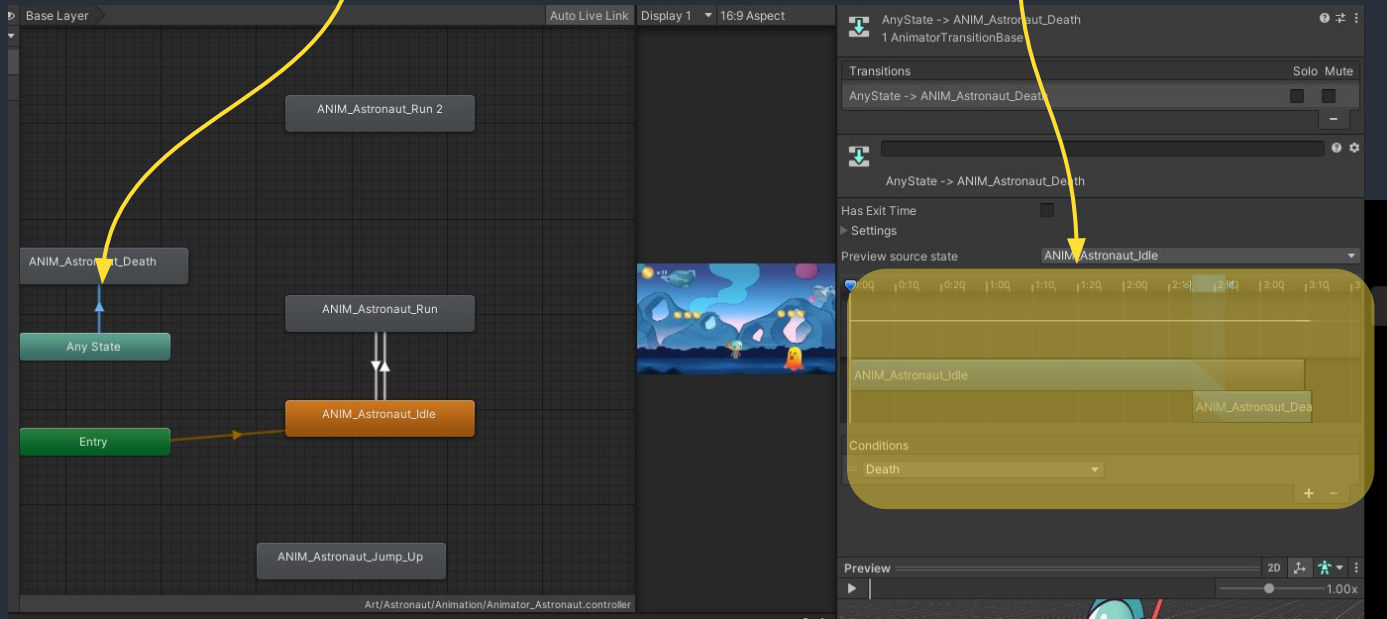
Após isso, basta selecionar a animação final que acontecerá a transição.



Animator

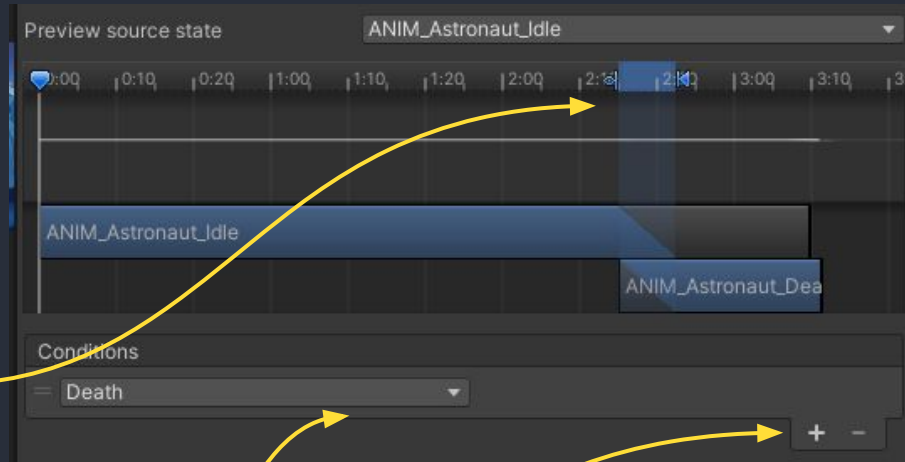
As transição aparecerão desta maneira no controlador:

Quando seleccionadas, podemos ajustar os parametros aqui:



Animator

Aqui podemos ajustar a velocidade e ponto de transição da nossa animação.



Aqui podemos adicionar, remover e ajustar quais as condições de transição, baseado nos parâmetros que criamos no controlador.