

## Zumbi atacar jogador

### Transcrição

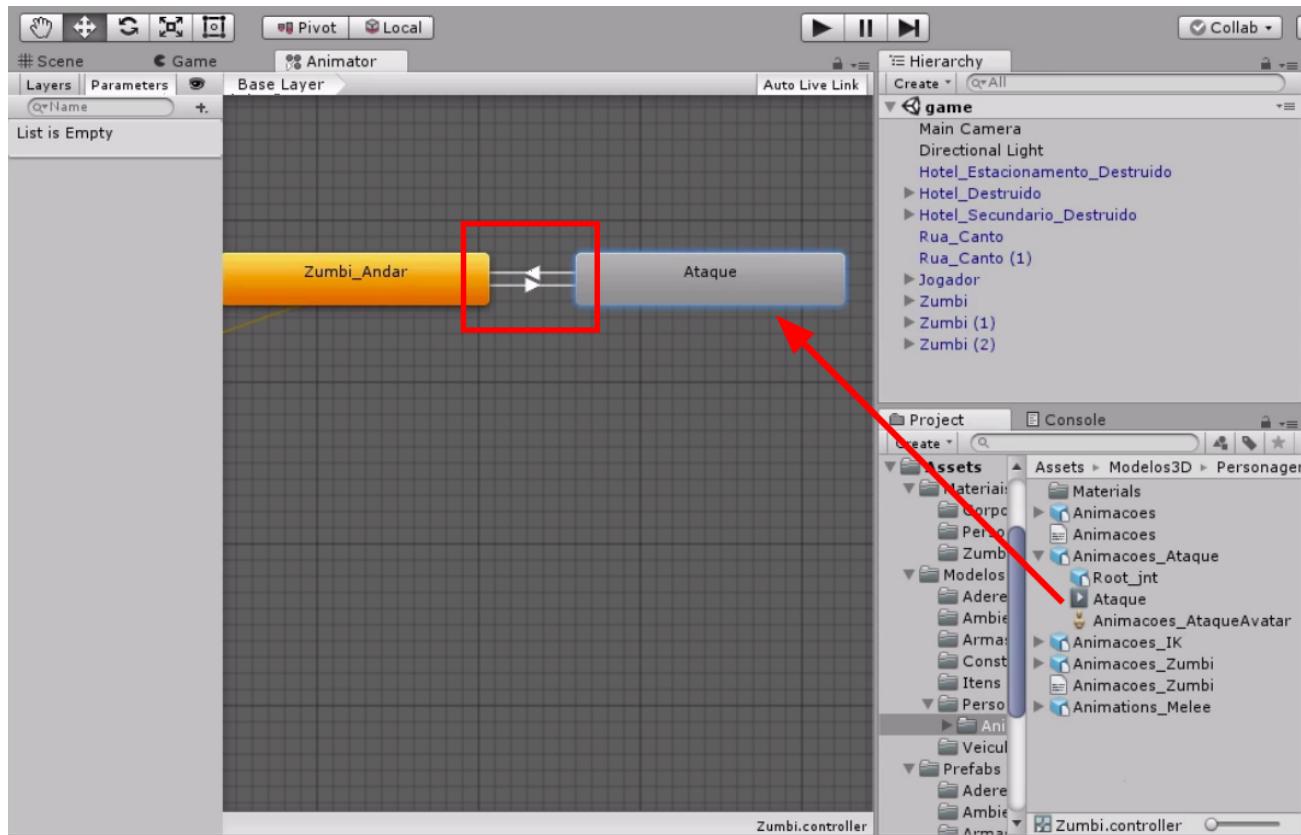
Anteriormente, configuramos o *script* da bala (`Bala.cs`) para que, ao acertar os zumbis, ambos desapareçam. Agora, precisamos fazer com que os inimigos ataquem a heroína.

Começaremos selecionando "Zumbi", em "Hierarchy", para depois aplicar as alterações de "Prefab" nas cópias deles. Em "Project > Assets" abriremos o *Animator* de "Zumbi", afinal, a única animação que aplicamos a ele foi a de andar. Adicionaremos o movimento de ataque acessando "Project > Modelos3D > Personagens > Animacoes" e selecionaremos "Animacoes\_Ataque", que já está editada para utilizarmos no curso. Se olharmos "Inspector" dela, veremos que estão ativadas as seguintes opções:

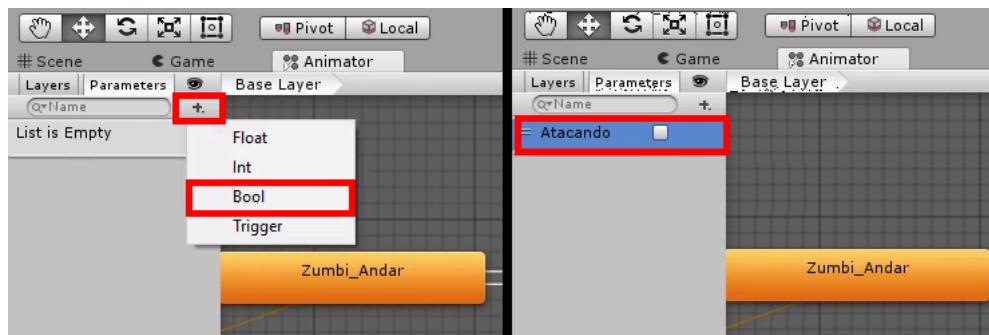
- "Loop Time";
- "Root Transform Rotation";
- "Root Transform Position (Y)";
- "Root Transform Position (XZ)".

Porque queremos que o **movimento** aconteça **somente na animação** e não que ele movimente os zumbis, modificando a cápsula ou o colisor de lugar. Queremos, simplesmente, que ele aconteça na animação.

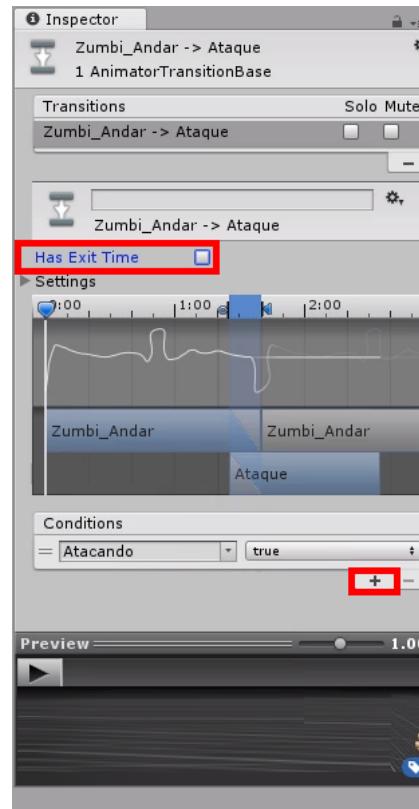
Clicaremos e arrastaremos "Ataque" para "Animator". Aplicaremos a transição de "Zumbi\_Andar" para "Ataque" e vice-versa, clicando com o botão direito do mouse em cima das animações, selecionando a opção "Make Transition" e direcionando a seta de transição de uma para a outra.



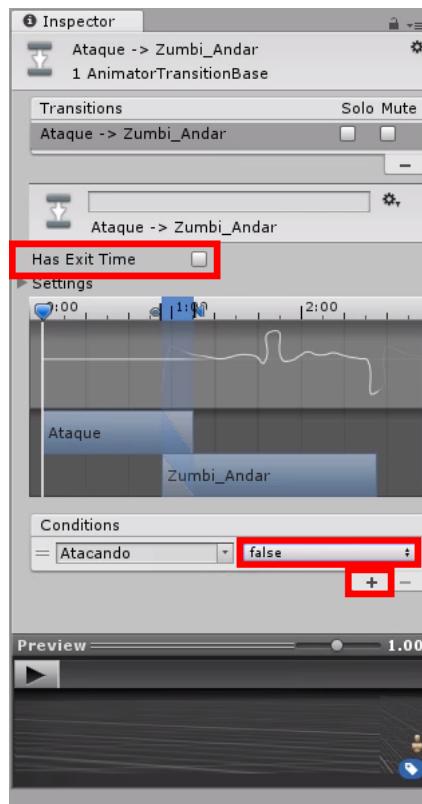
Na sequência, criaremos um parâmetro, clicando no botão com sinal de soma (+) no topo do menu lateral direito da tela. Selecionaremos um do tipo "Bool" e o nomearemos como "Atacando".



Para que o zumbi passe da animação de andar para o ataque, clicaremos na seta que sai de "Zumbi\_Andar" e, em "Inspector", desmarcaremos a seleção "Has Exit Time" e adicionaremos "Atacando" a "Conditions".



Clicaremos na seta que sai de "Ataque" e desmarcaremos no "Inspector" dela a opção "Has Exit Time" também, adicionaremos a condição "Atacando" e a configuraremos como "False".



Lembrem-se que estamos desmarcando a opção "Has Exit Time" para que a **transição** ocorra **instantaneamente**, sem intervalo.

Feitas as transições, iremos ao *script* do inimigo ( *ControlaInimigo.cs* ) para configurar o ataque. Anteriormente, estabelecemos no seguinte trecho de  `FixedUpdate()` que, quando estiver a determinada distância da heroína, ele deve andar em direção a ela.

```
if(distancia > 2.5)
{
    Vector3 direcao = Jogador.transform.position - transform.position;
    GetComponent<Rigidbody>().MovePosition
        (GetComponent<Rigidbody>().position +
    direcao.normalized * Velocidade * Time.deltaTime);

    Quaternion novaRotacao = Quaternion.LookRotation(direcao);
    GetComponent<Rigidbody>().MoveRotation(novaRotacao);

}
```

Abaixo do `if`, que estabelece uma ação para quando os zumbis estão a determinada distância da heroína, criaremos um `else` para quando estiverem perto dela. No entanto, queremos que, durante o ataque, ele vire para a heroína e ataque-a olhando para ela. Para isso, moveremos o seguinte trecho para cima de `if` :

```
Quaternion novaRotacao = Quaternion.LookRotation(direcao);
GetComponent<Rigidbody>().MoveRotation(novaRotacao);
```

Será apontado um erro relacionado a variável `direcao`, pois a declaração dela permaneceu em `if`. Sendo assim, transferiremos o trecho no qual a declaramos para cima de `Quaternion`. Para simplificar esse procedimento, podemos utilizar os atalhos:

- "Ctrl + X" para cortar;
- "Ctrl + V" para colar.

De volta ao `else`, faremos os zumbis atacarem, utilizando `GetComponent<>` para acessar `Animator` e passar o valor de variável Booleana (`SetBool()`), verdadeiro ou falso. No caso, o parâmetro "Atacando" e o valor que queremos passar, após vírgula ( , ), que é `true` — de verdadeiro — considerando que queremos que o zumbi ataque a heroína quando estiver perto dela.

Copiaremos — selecionando e utilizando o atalho "Ctrl + C" — e colaremos o trecho de `GetComponent<>` de `else` em `if`, no qual trocaremos `true` por `false`.

Assim, estabelecemos por meio de `SetBool` que, quando os inimigos estiverem:

- **longe** da heroína (`if`), "Atacando" é **falso** (`false`);
- **perto** (`else`, oposto de `if`), "Atacando" é **verdadeiro** (`true`).

Salvaremos e minimizaremos o trecho que editamos de `ControlaInimigo.cs` da seguinte forma:

```
Vector3 direcao = Jogador.transform.position - transform.position;

Quaternion novaRotacao = Quaternion.LookRotation(direcao);
GetComponent<Rigidbody>().MoveRotation(novaRotacao);

if (distancia > 2.5)
{
    GetComponent<Rigidbody>().MovePosition
        (GetComponent<Rigidbody>().position +
    direcao.normalized * Velocidade * Time.deltaTime);

    GetComponent<Animator>().SetBool("Atacando", false);
}

else
{
    GetComponent<Animator>().SetBool("Atacando", true);
}
```

De volta à Unity, ativaremos "Play" e veremos os zumbis atacando a heroína.

