

## Eventos de animação

### Transcrição

[00:00] Bom, agora eu já terminei de atualizar todas as animações da jogadora e do zumbi, então eu vim aqui na jogadora, abri a timeline dela e fui em todas as animações atualizando as imagens, fazendo aquela busca aqui pelo jogador, na nossa pasta assets, então eu fui aqui buscando jogadora underline andar ou underline parado, a animação que eu queria, com a direção que eu queria. Então, direita, esquerda e fui atualizado, fui trazendo essas imagens para minha timeline. Da mesma maneira que a gente tinha feito com as imagens soltas.

[00:30] Agora a gente está usando um atlas só para facilitar o trabalho do processador e da nossa placa de vídeo. Mas vamos ver como ficou isso então no nosso jogo? Vamos abrir a Cocos, vamos dar um play. Olha, nossa jogadora agora está muito mais destacada do fundo, fica muito mais fácil de ver onde ela tá. A animação del parada também está muito mais fluida, está muito mais fácil de, não está fácil, ela deixa de pular uns frames, ela está mais contínua, está mais suave. E a andando também, ela está andando com mais suavidade

[01:03] E agora a gente também tem as animações dos zumbis, se a gente vier para cá, os zumbis também estão mais destacados do fundo, então eles estão maiores, tanto o nosso zumbi, quanto os nossos jogadores eles estão maiores, todas as imagens que mandaram para a gente, estão um pouco maior, dos prédios, da animação, do zumbi, da jogadora de zumbi aliás.

[01:22] E agora a gente tem a animação deles andando em todas as direções, ficou bem melhor, ficou bem mais fácil de entender o que está acontecendo. Mas a gente agora também tem, mandaram para a gente, eu já adicionei no zumbi, se a gente vier aqui, as animações de ataque, então agora a gente tem uma animação que faz o zumbi parecer que está atacando.

[01:39] Então, dentro do zumbi, se eu abrir a timeline dele aqui, deixar ele com bastante zoom, a gente consegue ver, vamos selecionar aqui, eu tenho o atacar cima, e se eu der um play nela, já vi que ele tem essa antecipação do ataque e ataca de fato. E eu tenho para todas as direções essa animação, então eu posso vir aqui na esquerda cima, que é a outra direção que ele tem. A gente também tem a antecipação e depois o ataque, e a gente precisa agora fazer com que o código chame essa animação. A gente precisa alterar o código de ataque do zumbi, para ele chamar a animação de ataque, para ele realmente executar essa animação.

[02:14] Porque a gente, se a gente voltar aqui na Cocos e for em cima de um zumbi, deixar ele atacar a jogadora, ele não vai fazer nada, ele só vai continuar da mesma maneira que estava antes. Apesar de eu ter adicionado a animação nele, então se eu ficar aqui esperando eles me atacarem, eles vão ficar só ali parados, não sabendo para qual direção andar porque eles estão bem em cima do meu ponto, e me atacando sem nenhuma animação.

[02:36] Bom, então a gente vai precisar mexer no script do zumbi. Vamos aqui abrir a nossa pasta de scripts, inimigo, que é o script do zumbi, e a gente tem aqui o atacar, o nosso método atacar, então é nele que a gente vai mexer e vai fazer com que o zumbi ataque a nossa jogadora.

[02:53] Para atacar, para fazer uma animação, a gente sempre tem que chamar o controle de animação, da mesma maneira que a gente está fazendo aqui no método de andar, o método, vou chamar o mudar a animação, dentro do controle de animação você tem o mudar a animação, passando qual que é a direção que a gente quer e qual que é o estado. O estado, o nosso é o atacar, a gente está atacando a jogadora, e a direção é a direção do nosso alvo, a gente sempre quer atacar o nosso alvo, o método atacar que a gente tem aqui, ele não sabe qual a direção do nosso alvo, então a gente pode passar para ele né?

[03:23] Eu posso pegar a direção do alvo e passar para ele por parâmetro, da mesma maneira que eu estou fazendo com o andar, então o atacar vai receber agora uma direção e dessa direção eu vou chamar o `this.controleanimação.mudaanimação`, passando a direção, que essa animação tem que executar, e o estado, o nosso estado é atacar. Então, aqui a gente deve ver agora a animação do zumbi atacando a jogadora, vamos lá, vamos voltar para Cocos, fechar aqui o prefab dele e dar um play no nosso jogo, ver como ele tá, como que essa animação vai executar.

[04:00] Então, nossa jogadora está aqui, o zumbi está ali, vamos deixar ele ver a gente e começar a perseguir, e quando ele chegar perto da nossa jogadora. Você viu isso? Ele tenta atacar ela, mas ele só, ele parece que ele só dá uma, ele muda de lugar e volta. A animação não está executando de fato, ela não começa e vai até o final. O que acontece? A gente está pedindo para ele atualizar a animação, dentro do meu método atacar, só que o método atacar, ele está dentro do meu update, e como que funciona o update?

[04:36] Ele está rodando todo o frame, e no caso ele vai chamar o método atacar, quando a minha distância for menor que a distância de ataque, e o cronômetro de ataque não tiver zerado, quando eu puder atacar. Só que assim que eu ataco, assim que eu chamo o método atacar, eu volto o cronômetro de ataque para o tempo inicial dele.

[04:56] Então, ele vai passar uma vez no método atacar. Então, durante um frame ele vai chamar esse método, a animação vai mudar e no frame seguinte ele vai pular esse if porque o meu cronômetro já não é menor que zero, ele vai entrar no if ou de, a distância vai ser menor que a distância de perseguir, ou ele vai vagar.

[05:15] Nas duas chances, nesses dois casos, ele vai chamar o método andar. Porque se a minha distância for menor que a distância de perseguir, eu vou andar direto, se eu estou vagando, a gente tem aqui, a gente no final do método vagar, a gente chama o método andar. E o método andar faz a animação mudar para o estado andar, então a gente não pode fazer, a gente tem que evitar que esse método andar seja chamado. Uma das maneiras de fazer isso é evitar de chamar o meu cronômetro.

[05:44] Eu não quero saber se o meu cronômetro está menor do que 0 ou não, eu quero saber se eu estou na distância de ataque, eu ataco. A gente pode voltar lá e testar de novo. Agora a gente não vai ver mais, a gente não tem mais o cronômetro de ataque, se eu estou perto o suficiente eu ataco, e ponto. Então, vamos chegar perto do zumbi, deixar, vamos deixar só um vir aqui perto de mim, aqui três me perseguindo.

[06:08] Você viu que a animação já mudou um pouco, só que agora a gente vê que ele está com o braço para baixo. Ele está começando a animar, só que a minha vida, a vida da jogadora foi muito rápido, porque ela não tem mais aquele tempo. Todo frame ele está tirando dano dela.

[06:23] Então, a gente tem que ter um jeito de começar a animação de ataque e fazer com que ele ataque uma única vez a jogadora, como que a gente vai fazer isso? Bom, eu vou voltar aqui no meu código e a gente tem que pensar também, aliás, vamos voltar na Cocos, vamos voltar lá no zumbi.

[06:41] A gente tem que pensar que quando a gente viu aquela animação dele, ele tinha uma antecipação, ele jogava o braço dele lá para trás, para depois atacar a jogadora. E aí a gente só quer que a vida da jogadora diminua, quando realmente a mão dele tiver em cima dela, porque se ele está na antecipação, se a mão ele está lá atrás, ele não está atacando a jogadora, a gente ainda pode desviar, então essa antecipação é muito importante.

[07:06] Se eu pegar aqui uma animação de ataque, essa antecipação é muito importante, a gente tem que evitar, a gente tem que considerar ela na hora de atacar. Bom, além disso, eu preciso dar um jeito de travar o loop de update, porque se eu iniciei o ataque, se eu comecei a animação de atacar, eu não preciso continuar mais, ele já vai atacar a jogadora. Como que a gente pode fazer isso?

[07:30] Bom, ao invés de atacar assim que eu começo, eu vou iniciar o ataque. Então, `this.iniciarataque`, porque o que o iniciar ataque vai fazer? Ele vai chamar a animação de ataque e vai falar para o meu zumbi, ele vai falar para todo mundo do zumbi, olha, eu estou atacando, eu não quero fazer mais nada além de atacar. Então, se ele vai chamar a animação, ele vai precisar da direção do alvo e o meu atacar não vai precisar mais dela, porque essa linha do atacar vai para outro método.

[08:01] Então, eu tenho aqui o `iniciar ataque`, vai receber, aliás, ele é um método, então dois pontos `function`, ele vai receber uma direção, que é a direção que eu estou atacando, e o que ele vai fazer? Ele vai pegar essa linha aqui, que é a de controle de animação, e vai falar, olha, eu iniciei a minha animação de ataque, e eu vou configurar agora, falar para o meu zumbi, para esse script do zumbi que eu estou atacando.

[08:31] Então, `this.atacando` vai ser igual a verdadeiro, e o que a gente vai fazer com isso? Vamos colocar uma vírgula aqui que vai precisar disso, senão o nosso código não compila, e `iniciar ataque`. Então, se minha distância for menor que a distância de ataque, eu vou iniciar o ataque, isso significa que eu vou começar a minha animação, eu não vou atacar de uma vez. Então, eu vou iniciar o meu animação, e se eu estou atacando, eu não quero fazer mais nada disso.

[08:58] Então, eu não preciso, eu posso fazer um `if` aqui, se eu estou atacando, então se `this.atacando` for igual a `true`, eu não quero fazer nada disso, ou seja, se eu não estou atacando, eu quero fazer meu método `update`. Tudo isso aqui vai para dentro desse `if`, então assim que meu loop de `update` começar, ele vai falar, eu estou atacando, o zumbi está atacando?

[09:22] Se ele não tiver atacando, ok, roda o loop de `update`. Se ele estiver atacando, ou seja, se eu estou dentro daquela animação de atacar, eu vou parar o meu `update`. Então, eu vou ter que criar essa propriedade `atacando`, que aliás vai ser uma propriedade privada também, ela não vai ser uma propriedade pública porque só interessa para o zumbi, então `cc.booleana`. Ela é uma propriedade booleana, então ela recebe `true` ou `false`.

[09:52] Eu começo o meu jogo com o `atacando` igual a falso, então `this.atacando` vai ser igual a falso no começo do jogo. Eu mudei ela, ela não é `atacando` sem o underline. Ela tem um underline logo antes, então `this.atacando` e o meu `iniciar ataque`, `iniciar ataque` e `this.atacando` vai ser falso também.

[10:15] Então, agora a gente vai ver que quando o meu jogador, o meu zumbi estiver perto da jogadora, ele tem que começar a fazer a animação. Vamos testar de novo, então aqui a gente deu um play na Cocos, vamos lá testar, vamos puxar um dos zumbis. Vamos tentar puxar só um deles de novo, vamos matar esse. Quando ele chegar perto de mim, ele tem que, ele atacou e ele está me atacando. Mas repara que ele ficou parado, eu estou longe dele e ele continua atacando, e minha vida não diminuiu, a vida da jogadora ainda não foi afetada. O que está acontecendo?

[10:53] Bom, voltando lá no nosso código, eu chamei o método `iniciar ataque`. Se minha distância for menor que a distância de ataque, eu chamei o `iniciar ataque`. Esse `iniciar ataque` vai começar a animação e falar para todo mundo, olha, eu estou atacando, mas eu nunca chamei o método `atacar`, que é quem dá realmente o dono.

[11:10] Aliás, a partir do momento que eu ataquei a jogadora, que eu já emiti o evento de sofrer dano para ela, eu também posso falar que o meu `atacando` vai ser falso, aqui eu já terminei o meu ataque. Então, o meu `atacando` vai ser falso, e a gente não está usando mais o cronômetro de ataque, então eu posso apagar essa variável.

[11:30] Mas agora como que a gente vai fazer para chamar o nosso método de atacar? A gente vai precisar fazer isso através da animação, e existe um jeito dentro da Cocos que você faz isso. Se a gente abrir a Cocos aqui, de novo vamos voltar lá no zumbi, a gente vai ter que abrir a animação de ataque dele e a gente vai fazer um evento de animação.

[11:56] Ou seja, quando a gente entrar em algum determinado frame, algum ponto daquela animação, é a própria animação, a própria Cocos vai disparar um método do nossa classe, do nosso objeto. Então, por exemplo, se eu pegar o

atacar para direita ou atacar direita cima, vamos dar um zoom aqui, eu posso trazer o meu cursor, a minha linha de visualização da animação para o ponto que eu quero.

[12:24] Então, eu só quero chamar o meu atacar, quando o braço do zumbi estiver lá na frente, e eu vou clicar nesse símbolo aqui que parece um cursor de quando você está escrevendo, e ele vai falar para adicionar um evento de animação. Se eu clicar nele, a gente vê que tem esse quadradinho branco que apareceu bem em cima da minha timeline, ele é um pouco pequeno, mas se você clicar nele duas vezes, ele vai abrir uma segunda janela, onde você vai falar para Cocos, qual que é o método que você quer chamar dentro desse objeto, quando a animação chegar nesse ponto?

[12:57] O método que a gente quer chamar, se a gente voltar no código, isso daqui sai da frente, ele não está sumindo, mas tudo bem. O nosso método é o atacar, então a gente quer chamar o método atacar, assim que a Cocos, assim que aquela animação de ataque chegar naquele ponto. Então, a gente pode fechar, e aí a gente tem aqui o evento de animação.

[13:15] A gente vai fazer isso para todos, então atacar baixo, ele tem a mesma coisa, eu preciso trazer meu cursor para cá, clicar para adicionar o evento de animação, abrir a janela de evento de animação e falar que eu vou chamar o método atacar. Então, eu vou fazer isso para todo mundo e a gente vai ver o resultado. Pronto, eu já adicionei todos os eventos de animação, então eu fui lá em todas as animações e coloquei o evento, a gente pediu para chamar o método atacar e agora a gente pode testar.

[14:09] Então, assim que o meu inimigo estiver perto o suficiente do meu jogador, ele vai começar a animação de ataque. Quando a animação dele chegar naquele ponto onde o braço está para frente, ele deve reduzir a vida da minha jogadora e liberar o update do zumbi de novo. Então, o atacando dele vai virar falsa, vamos lá buscar um zumbi. Trazer ele para perto de mim, então ele vai chegar perto de mim, ele deu ataque e me tirou o dano.

[14:43] Então, minha vida já reduziu, e assim que eu afastei dele, ele vai voltar a me perseguir. Então, agora a gente já tem um zumbi com animação de ataque, que só ataca mesmo, só tira o dano quando a animação dele chega no ponto específico que a gente quer, através de um evento de animação. O legal disso é que a gente não precisou mais usar o cronômetro, a própria animação dá o tempo, dá a velocidade de ataque do zumbi, então se a animação fosse mais rápida, ele ia atacar mais rápido. Se ela fosse mais devagar, ela ia atacar mais devagar.