

Zumbi Vagando

Transcrição

[00:00] Agora que já arrumamos a posição do nosso tiro, tem a jogadora, agora ele sempre sai da arma, e que ela não é mais soterrada pelos zumbis, tem mais um detalhe que a gente precisa arrumar em relação aos zumbis. Porque a gente tem um cenário gigante agora, e a gente tem o gerador de zumbis no cenário inteiro, só que independente da onde eles nasçam, eles vão vir na minha direção, por exemplo, esses daqui eles estão dentro do gerador que provavelmente está mais perto de mim, porque eles chegam antes. Mas tem uns aqui embaixo, que eles estão querendo vir na minha direção.

[00:27] Parece que tem muitos aqui embaixo, o gerador está gerando muitos zumbis, todos aqui embaixo. Tem outro gerador que a gente tinha deixado lá longe, ele nasceu há muito tempo atrás, então ele está bem longe de mim, e ele continua vindo na minha direção. Isso não faz muito sentido, porque olha o tamanho do nosso cenário, se gente tirar o zoom aqui com a bolinha do mouse, olha o tamanho desse cenário que a gente tem. Então, a minha jogadora fica muito pequena dentro desse cenário, e o zumbi, ele consegue ver ela, ele vai atrás dela, independentemente de onde ele nascer.

[00:56] Por exemplo, aquele gerador que estava lá longe, olha onde ele tá, ele está muito lá atrás, ele está muito longe da minha jogadora. Será que o zumbi realmente tem esse campo de visão todo? Por exemplo, se eu tivesse com um gerador aqui em cima não, os zumbis conseguiriam ver a minha jogadora? Provavelmente não, porque tem um prédio na frente, eles não deveriam conseguir ver ela.

[01:13] Então, a gente precisa fazer algum jeito dos zumbis terem um campo de visão, para eles verem a minha jogadora quando ela está perto deles, mas se ela não tiver perto, ele precisa fazer outra coisa, ele não está vendo nenhum alvo, então ele precisa vagar, ele vai andar por aí, ele vai procurar comida, ele vai fazer outras coisas. Quando a jogadora tiver perto, ele sabe que ela é um alvo, ele sabe que ela é comida para ele, então a gente consegue fazer ele perseguir ela só quando ela tiver perto.

[01:35] Para fazer isso, a gente vai precisar mexer no nosso script do inimigo, e bom, onde que a gente vai fazer isso? A gente tem aqui as propriedades, o método load, o método update, que é onde a gente tem tudo, que ele vê todo frame, e o morrer. Bom, quando ele morre, ele morre, ele não faz nada. O método load, eu só estou buscando as minhas dependências, então eu tenho que fazer no método update, já que ele verifica a posição do alvo dele, da jogadora, em qualquer frame.

[02:06] Mas está muito confuso esse método update, olha quanta coisa a gente tem aqui dentro, eu pego a direção do meu alvo, eu calculo a distância que ele tá, aí eu faço o controle de animação virar para um andar, eu configuro a animação dele para andar, eu coloco a direção dentro do componente movimentação, eu faço ele andar para frente, eu tiro o cronômetro de ataque, olha quanta coisa tem dentro de um único método. A gente pode organizar isso antes de continuar fazendo o que a gente quer, que é fazer os zumbis vagarem.

[02:34] Essas três linhas aqui, então o meu controle de animação e a minha movimentação, as três estão fazendo uma única coisa, que é fazer o meu zumbi andar, então a gente pode tirar todas essas três linhas aqui, e chamar algum método tipo this.andar, já que o meu zumbi tem que andar na direção do alvo, e a direção que a gente quer que ele ande é a direção.

[02:56] Então, como a gente calcula isso dentro do update, a gente passa ele para o parâmetro, a gente consegue receber pelo parâmetro dentro desse andar. Então, vamos criar esse método andar, ele vai ser uma função, ele recebe a direção

que ele precisa andar e aqui a gente cola as três linhas que a gente queria. Então, agora já está um pouco mais fácil de ler. Então, a gente pega a direção do meu alvo, calcula a distância e faz ele andar.

[03:22] O meu cronômetro já está aqui, eu preciso dele para ver quando o meu zumbi tem que atacar a jogadora, a gente não quer que ele ataque todo frame, senão ela vai perder vida muito rápido. Então, ele não tem muito o que fazer, mas dentro desse if, se minha distância já está aqui, meu cronômetro já está aqui e for menor do que zero, dentro desse if, o que eu estou fazendo é atacar o meu alvo.

[03:43] Então, eu posso também tirar essas duas linhas. Vamos copiar elas para cá, vamos colar e vamos chamar um `this.atacar`, porque é o que a gente está fazendo, se essa condição for verdadeira, eu quero atacar o meu alvo, então a gente pode criar esse método, essa função `atacar`. Já fica um pouco mais limpo, nosso método `update`. Então, pega a direção, calcula a distância, ando, tira o meu cronômetro de ataque, eu posso subir ele, deixar, fazer isso no começo.

[04:12] Se eu estou dentro da área de ataque e posso atacar, eu vou realmente atacar a minha jogadora. O andar, ele precisa da direção, como a gente calcula a direção? No método `update` a gente está passando para parâmetro, só que direção aqui no `update` é um termo muito ruim, é um nome de variável, de propriedade, um pouco ruim. Então, essa direção na verdade é a direção do nosso alvo, então uma direção do alvo é o que eu quero, e é o que eu estou passando para ele andar.

[04:41] Então, o que a gente queria fazer? Agora que a gente já organizou um pouco mais nosso método `update`, a gente quer que quando o zumbi não conseguir ver a jogadora, ele fique vagando, ele fica por aí, na cena. Então, eu vou ter aqui o método `vagar`, que é o método que ele vai chamar quando ele não estiver vendo a jogadora. Só que quando que ele não está vendo a jogadora?

[05:06] Bom, a gente não tem realmente que calcular para onde ele está olhando exatamente, se tem alguma coisa na frente ou se não tem. A gente pode simplificar um pouco as coisas e falar que se a jogadora está perto dele, em alguma distância, ele está vendo ela. Independente se aí tem um prédio, tem alguma coisa, mas se essa distância for pequena o suficiente, dificilmente ele vai ver ela através de um prédio, porque os prédios que a gente tem na nossa cena são consideravelmente grandes.

[05:29] Então, a gente consegue fazer desse jeito. A gente vai precisar de uma distância para essa verificação. Essa distância vai fazer o quê? Se minha jogadora estiver dentro dela, o zumbi vai conseguir perseguir ela. Então, a gente pode colocar outros ifs aqui dentro do nosso `update`. Se eu estou na distância de ataque, que é a menor distância que eu tenho, porque para atacar eu tenho que estar bem perto dela, eu vou atacar.

[05:57] Se não, então `else`, eu vou verificar, eu estou vendo a jogadora, eu estou pelo menos vendo ela? Então, a distância vai ser menor do que a minha distância de perseguir, porque ela está dentro do meu alcance, então `this.distanciaperseguir`, então se a minha distância atual for menor do que a distância de perseguir, aí eu vou perseguir ela, então para perseguir a jogadora eu estou andando, eu vou andar na direção do meu alvo.

[06:31] Então, a gente vai andar na direção do meu alvo. Agora se eu não estou nem perto o suficiente para atacar, nem perto o suficiente para perseguir minha jogadora, eu vou vagar. Então, o meu último, a minha última posição, meu último `else`, minha última condição é o `this.vagar`. Se eu não estou nem conseguindo atacar, nem conseguindo ver a jogadora, eu só vou vagar, eu vou andar por aí. Então, agora a gente já separou, o nosso `update` está muito melhor, porque ele está mais limpo, a gente extraiu os métodos. Então, a gente vai atacar, andar, e vagar.

[07:04] Agora como que o zumbi vai vagar? Bom, quando você está vagando, você não tem muito para onde ir, ele vai andar para um lado, aí ele para, aí ele escolhe outra direção e ele anda um pouco, ele não tem muito um comportamento definido. Como a gente já tem o método de andar fácil para a gente utilizar, vamos usar ele, vamos fazer justamente isso.

[07:25] A gente escolhe uma direção aleatória para o zumbi, faz ele vagar por um tempo, e depois que esse tempo zera, se ele não está perto da jogadora ainda, ele escolhe outra direção, ele sorteia outra direção. Bom, então dentro do meu vagar, a gente vai ter aqui que no final eu vou querer andar, então ele vai andar para alguma direção, direção vagar, essa direção ela é global né? Porque eu vou precisar alterar ela em vários momentos, ela não pode ficar só dentro do nosso método, então `this.vagar`, é uma direção vagar, é uma variável que vai ser global.

[08:00] E quando que eu vou sortear essa variável? Eu vou sortear essa variável se o meu cronômetro vagar, ou se o meu tempo restante para vagar - então `this`., vamos criar uma variável privada, tempo restante para vagar - se esse tempo restante para vagar for menor que 0, ou seja, se eu tenho um cronômetro e ele zerou, aí sim eu vou escolher uma nova direção. Então, `this.direçãovagar`, vai ser igual a uma nova direção. A gente já viu que para sortear uma direção aleatória, a gente pode usar um novo vetor, sorteando um `math.random - 0.5`, então uma direção negativa ou positiva no X e Y.

[08:52] Então, eu posso vir, fazer um `new cc.vec2` que é o novo vetor, e para o X e Y dele, eu vou criar uma posição aleatória. Então, `math.random`, aliás, uma posição não, um valor aleatório, `math.random`, que aí eu vou sortear um número entre 0 e 1. Só que eu não quero só os números positivos, eu quero positivos e negativos, então eu vou fazer - 0.5.

[09:22] Então, eu vou ter entre -0.5 e 0.5, uma posição, um valor em X. A mesma coisa para o Y, então vai ser o `math.random - 0.5`, esses são os valores que a gente quer para nossa direção vagar. Só que eu não tenho ainda o meu tempo restante para vagar, a gente não criou isso. A gente já criou um cronômetro, a gente sabe como ele funciona. Se eu tenho um cronômetro, um cronômetro de ataque, assim que eu atacar, eu tenho que zerar esse cronômetro. Então, quando eu sortear uma nova direção, eu tenho que zerar esse cronômetro do meu tempo restante para vagar.

[09:52] Então, o `this.timeparavagar` vai ser igual alguma variável que o meu game designer vai decidir, porque ele aí ele consegue decidir quanto tempo o zumbi vai ficar andando em cada direção, se eu tiver um tempo muito longo, ele vai andar infinitamente. Se eu tiver um tempo muito curto, ele vai ficar mudando de direção muito rápido, ele vai parecer meio frenético. Então, isso a gente consegue controlar depois no meu inspetor da Cocos, vai ficar mais fácil.

[10:24] Então, `this.timeparavagar` vai ser igual ao `this.timevagar`, é uma variável pública porque a gente vai conseguir alterar no inspetor. Então, eu zerei o meu cronômetro, assim que ele virou 0 eu resetei o meu cronômetro. Só que eu não estou tirando o delta time dele, então a gente pode fazer a mesma coisa que a gente está fazendo aqui no cronômetro de ataque.

[10:51] A gente vai subtrair o delta time, então `this.time restante para vagar` vai ser menor ou igual o meu delta time, que é a diferença de tempo entre um frame e outro. A gente criou muitas variáveis aqui, a gente precisa trazer elas para as propriedades agora. Então, eu tenho o meu distância de perseguir, é uma variável pública, então eu vou colocar aqui junto com as variáveis públicas, ela vai ser um float porque eu quero um número, a minha distância é sempre um número.

[11:27] O meu distância que está aqui embaixo, eu vou trazer ele também para as variáveis públicas, separar em dois blocos que nem a gente fez da jogadora. Então, eu tenho a minha distância de perseguir, minha distância de ataque, o meu tempo para vagar, que é o tempo que ele vai ficar em uma única direção enquanto ele estiver vagando, `cc.float`, e eu tenho o tempo restante para vagar que a gente criou aqui embaixo.

[11:53] Meu tempo restante para vagar vai ser uma variável privada, que também é um número. Então, no cronômetro de ataque, a gente resetava ele, a gente iniciava ele com o tempo de ataque, assim que a gente dava o onload, no tempo para vagar a gente vai fazer a mesma coisa, então `this.time restante para vagar`, vai ser igual a `this.time vagar`. Então, a gente vai iniciar a nossa variável aqui no método on load.

[12:27] O tempo de ataque a gente não precisa iniciar, aliás, a gente já iniciou no inspetor. O distância perseguir, a gente vai iniciar lá no inspetor. Tempo de ataque a mesma coisa, dano a mesma coisa, o que vai faltar aqui é nossa direção de andar, nossa direção de vagar. A gente criou aqui no método vagar, tem a direção vagar que eu estou sorteando ela, mas ela precisa ser iniciada com algum valor. Eu não posso iniciar ela como nulo, porque assim que ele cair nesse if, nesse método vagar, ele pode dar um um erro, ele pode estar sem o valor definido, e a gente vai passar esse valor Indefinido para um andar e vai disparar um erro.

[13:02] Então, vamos criar nossa direção vagar, lá em cima no nosso painel de propriedade, então cc.vec2, e vamos fazer com que ele comece na direção vagar, vai começar como um vetor para cima. Então, this. direção vagar vai ser igual a um cc.vec2.up, que é uma constante, é um jeito da gente falar que é um vetor, com 0 no X e 1 para o Y. É um jeito fácil da Cocos criar esses atalhos, porque é muito comum você usar esses vetores que são 0 e 1, ou no X ou no Y, e vice versa.

[13:40] Então, agora a gente já tem o nosso comportamento de vagar. Lembrando, a gente está no update, a gente vai verificar a distância de ataque, verificar a distância de perseguir, e se nenhuma dessas der verdadeiro, aí sim a gente vai vagar. Então, podemos ir lá na Cocos, vamos colocar no nosso prefab o zumbi, então cliquei duas vezes no prefab, ele abre a aba de prefab aqui para a gente editar.

[14:03] Vamos nosso inimigo, a gente tem o alvo, dano, tempo de ataque, tempo de perseguir, vamos falar que ele vai perseguir, aliás distância de perseguir. Então, se a jogador estiver a uns 200 pixels dele, ele começa a perseguir ela, a minha distância de ataque é 50 e o tempo que ele vai vagar, enquanto ele não tiver com a jogadora, vamos falar que é 3 segundos, então a gente pode salvar esse prefab e vamos testar. Vamos dar um play aqui e testar o que está acontecendo.

[14:29] A gente pode abrir nosso console para ver se vai ter algum erro, a gente mexeu em muito código, então pode ser que aconteça algum erro, mas aparentemente não, porque os zumbis já devem ter nascido. Olha, ele está aqui, inclusive a gente está longe o suficiente, que a gente vai ver ele vagando. Ele escolhe uma direção, ele anda nela, aí ele muda. Mas eles estão andando, olha, agora ele mudou e agora ele começou a me perseguir.

[14:52] Você vê que quando eu cheguei perto dele, ele está me perseguindo, como eu sou mais rápido que ele, se eu for para longe dele, aí ele mudou de direção já, ele voltou para a direção de vagar dele. Se eu chegar perto, ele vai vir me perseguir e vai me atacar, aí eu começo a perder vida. Então, nossos zumbis agora, eles não têm aquela visão longe que tinha antes, eles estão muito mais perto. Aliás, eu deixei um bem perto, porque a gente conseguia ver o que estava acontecendo, então 200 aqui talvez seja perto demais.

[15:19] A gente pode aumentar um pouco, deixar talvez com 350, e assim que chegar perto ele vai me perseguir, se eu tiver longe ele vai ficar andando pelo cenário, e ajudando também, como eu só tenho três geradores, a popular aquele cenário. Porque de novo, a gente tem um cenário gigante e por enquanto a gente só tem três geradores.

[15:38] Mas então os nossos zumbis agora vão começar a popular esse cenário andando, eles vão escolher direções aleatórias. Pode ser que a gente venha aqui para esse cantinho, e algum zumbi esteja lá, porque ele acabou coincidindo de cair naquela posição, de andar até aquela posição. Então, muito legal, agora a gente pode passar para as próximas coisas.