

Ajustando Animação - Parte 1

Transcrição

[00:00] Agora que a gente já fez a jogadora, tem uma animação muito mais condizente com o estado dela no jogo, então a gente alterou o script da jogadora para a gente atualizar a animação dela todos os frames, a gente tem mais um detalhe para arrumar, porque agora ela está parada, enquanto ela está parada a animação dela de parada toca, quando ela começa a andar, ela começa a andar na animação, e se eu paro, ela para né?

[00:22] Só que repara que a nossa jogadora só está olhando nas diagonais, ela nunca olha só para direita, ou só para cima. Se eu ficar com o mouse aqui tentando acertar, você vê que ela pula de uma animação para outra, e não é isso que a gente quer. Então, o que a gente está fazendo?

[00:34] Vamos ver lá no nosso controle de animação, o nosso script, o que a gente fez. Então, o nosso controle de animação está aqui, toda vez que a gente chama o controle de animação para mudar a animação, a gente vai chamar esse método muda animação, onde a gente passa a direção e o estado.

[00:49] Repara que a gente está fazendo a verificação na direção X e na direção Y, toda vez que chama esse método. Na direção X, se a gente pegar como exemplo, se eu passar um vetor direção onde o X for maior do que 0, quer dizer que eu estou indo para direita, então a gente retorna, a gente concatena esse valor da minha próxima animação. Se ele não tiver maior do que 0, então ele pode estar menor do que 0, então a gente faz essa verificação também. Se ele tiver menor do que 0 no X, ele está indo para esquerda.

[01:16] A gente está fazendo isso toda vez que a gente, que a gente chama esse método, mas não é isso que a gente quer, o que a gente queria é que a jogadora olhasse só para direita, quando eu estou com mouse ali para direita dela, perto do zero zero, olhando em linha reta. Ou também, que ela olhasse só para cima, quando eu estou com o mouse para cima dela ou só para esquerda quando o mouse está para esquerda dela, ou só para baixo quando o mouse está para baixo dela.

[01:39] Mas como a gente está fazendo a verificação hoje então? Vamos pegar o caso da jogadora olhando para direita, para esta verificação, do jeito que a gente está fazendo hoje, dá certo? O nosso X precisa estar maior do que 0, porque aí eu vou retornar o direita para o meu string de próxima animação, e o Y ele tem a mesma verificação, então se o Y for igual a 0, ele não retornar nem para cima, nem para baixo.

[01:59] Essa é a verificação que a gente está fazendo hoje. Só que acontece que essa verificação ela é muito fraca, porque se o meu Y, o meu vetor direção, estiver um pouquinho para cima, o Y do vetor direção estiver um pouquinho fora muito, a gente já vai ter uma animação pulando para diagonal.

[02:16] Então, se ele tiver maior do que 0, ele vai pular para diagonal, nesse caso direita para cima. Se o Y estiver menor do que 0, ele vai pular para diagonal para baixo e para direita. Então, e assim, esse maior do que 0 é realmente maior do que 0 em qualquer número, se eu tiver quatro casas decimais e um 1, isso é maior do que 0. Então, 0,0001 é maior do que 0, e aí eu vou ter a animação pulando para cima, e não é isso que a gente quer, então a gente tentar fazer essa verificação hoje em dia, de Y igual a 0, ele tem que ser exatamente igual a 0.

[02:47] Como o meu mouse tem que estar exatamente alinhado com a âncora da jogadora, a gente também pode falar que a gente quer um ângulo entre o mouse e a jogadora, de 0°. Aí só quando tiver esse ângulo de 0, é que a nossa animação vai exatamente para direita, isso é muito difícil de a gente alcançar. Então, o que a gente gostaria de fazer é ter uma folga, é ter uma área onde eu posso ter o meu mouse para a direita da jogadora, e uma folga dentro do Y para que ela só olhe para direita.

[03:15] Se ela passar essa folga, dessa área, aí sim ela vai lá na diagonal e a mesma coisa para o y, se eu estou com o meu mouse em cima dela, eu tenho que ter uma folga em relação ao meu eixo X, para ela olhar para cima, e quando passar dessa folga, quando passar essa área, aí sim ela vai olhar na diagonal.

[03:33] Então, o que a gente quer, é fazer isso para todas as animações, eu coloquei aqui esse círculo pintando as áreas que são as folgas. Então, se o meu mouse tivesse dentro dessa área cinza, numa direção que apontasse para essa área cinza, eu quero que ele olhe na diagonal. Se ele tiver dentro das áreas mais claras, eu quero que ele ande nas ortogonais, então cima esquerda, direita embaixo. Para fazer isso, a gente vai ter que escolher qual é o ângulo.

[03:59] Quantos graus, já que a gente pode falar que a relação entre o mouse e a jogadora está em graus, quantos graus que a gente quer de folga. Se eu falar que eu tenho uns 20° de folga, então em relação ao eixo X, eu vou somar 20 graus, e se o meu ângulo do mouse e a jogadora tiver abaixo de 20, ela vai olhar só para direita, e a mesma coisa para o Y. Se o meu mouse tiver lá em cima, sempre é 90 graus. Então, $90 - 20$, a gente vai ter 70 graus então entre 70 e 90 graus, a minha jogadora tem que olhar para cima.

[04:34] Só que eu tenho que ter a folga para os dois lados né, então é 70 de um lado e 110 do outro, porque 90 mais 20, e assim sucessivamente. Eu vou ter para a jogadora olhando para esquerda, 180 que é o ângulo para esquerda retinho, em uma posição ao 0, e 160, então 180 menos 20, então eu vou ficar com 160 graus, e o 180 mais 20 para eu ter a mesma folga para o outro lado. Assim a gente vai fazer para todos os ângulos, só que olha quantas verificações a gente vai ter que fazer.

[05:07] Porque eu vou ter que ficar para cada animação, qual que é o ângulo que eu tenho no meu mouse, então se o meu ângulo tiver entre 0 e 20, ela vai ter que estar olhando para direita, entre 20 e 70 olhando na diagonal, entre 70 e 110 olhando para cima. Imagina o código que isso vai ficar, vai ficar um código muito comprido, vai ficar um código muito grande e muito difícil de dar manutenção.

[05:26] Então, seria ideal que a gente conseguisse reduzir essa quantidade de verificações, que a gente conseguisse fazer só um conjunto mínimo de verificações, e esse conjunto mínimo servisse para todo o círculo trigonométrico, para todos os ângulos que a gente têm. Então, o ideal na verdade, se a gente for pensar, é verificar um único quadrante, então ali eu estou pegando só a animação da jogadora indo para direita, indo na diagonal direita para cima, indo estritamente para cima, só para cima.

[05:55] Porque se a gente for reparar, esse padrão que a gente tem, de uma animação indo só no eixo X, uma animação indo no eixo X e Y, ou seja, diagonal, e uma animação indo no eixo Y unicamente, se repete em todos os quadrantes. Então, no caso do primeiro quadrante eu vou ter, entre 0 e 20 uma animação, entre 20 e 70 outra animação, e entre 70 e 90 outra animação.

[06:16] Mas eu posso repetir isso para todos os quadrantes. Então, a gente tem esse mesmo padrão sendo seguido em todos os quadrantes. Seria ótimo, seria ideal se a gente conseguisse ficar só o primeiro quadrante ao invés de todos eles. Mas para verificar todos os quadrantes, todos os ângulos, será que existe alguma coisa em comum entre todos os quadrantes, entre todos os ângulos no círculo trigonométrico?

[06:39] Então, a gente teria aqui o círculo, o que é comum a todos os ângulos? De todos os ângulos, eu consigo tirar o seno e cosseno, mas o que é seno e cosseno? São conceitos que a gente vê no colégio e a gente esquece às vezes. Então, vamos pegar de novo lá os 20 graus como exemplo, e vamos calcular o seno dele, o que é calcular o seno dele? Eu vou pegar aqui da onde está batendo o meu ângulo, dentro do círculo trigonométrico, na borda do círculo trigonométrico, e vou traçar uma linha até chegar no eixo Y, a gente tem aqui o eixo Y e X.

[07:11] Então, traçando essa linha, a gente vai ver, a gente vai depois calcular do meu ponto 0, do centro do círculo, até onde essa linha bateu, e esse valor é o valor do seno, é só isso. Então, eu vou projetar o meu ângulo ou no eixo X ou Y,

porque o cosseno a gente vai projetor no eixo X, e vou depois medir do centro do meu círculo até onde essa projeção bateu, onde essa projeção está cruzando o meu eixo.

[07:32] Então, no caso de 20 graus, ele tem 0.34 de seno, então ele tem entre o centro do círculo e onde bateu a projeção do seno, deu 0.34 e ele tem 0.93 de cosseno, ou seja, entre o centro do círculo, que é o meu ponto 0, e onde bateu a projeção do ângulo 20 graus, a gente tem 0.93 de distância. Mas por que 0.93 e 0.34, da onde que vem esses números?

[07:59] Quando a gente fala de círculo trigonométrico, a gente está trabalhando com ângulos, entre trigonometria, a gente sempre está trabalhando com círculo que se chama de círculo trigonométrico ou círculo unitário, porque ele tem 1 de raio. Então, qualquer distância, qualquer ponto que eu pegar dentro do meu círculo, nas bordas aliás, em comparação com meio vai ter 1 de distância. Então, do centro do meu círculo para qualquer ponto, eu vou ter sempre 1, ou seja, o máximo que eu consigo chegar é um valor entre 0 e 1, medindo sempre do centro.

[08:27] E a gente vai ter, no nosso seno e cosseno, podem ser negativos ou positivos, porque como eu estou mentindo do centro, a gente o padrão de quando eu estou indo para direita no eixo X, quando eu estou medindo o cosseno, eu vou medir entre 0 e 1. E se eu estou indo para esquerda do eixo X, em relação ao meu 0, eu vou medir em -1. A mesma coisa para cima para baixo, se eu estou indo para cima é 1, se eu estou indo para baixo é -1, ou seja, todos os senos e cossenos que a gente vai calcular dentro do círculo trigonométrico, tem valor entre -1 e 1.

[08:55] Então, voltando lá para os nossos 20 graus. Se a gente vai usar seno e cosseno, se é uma propriedade que todos os ângulos tem, como que a gente pode usar isso para ajudar a gente a fazer uma única verificação, verificar só aquele primeiro quadrante? Bom, eu tenho essa diferença entre 0 e 20 graus em relação ao eixo x, mas o que acontece se eu espelhar esse ângulo, se eu inverter ele em relação ao eixo Y?

[09:21] Ou seja, meu centro do círculo está aqui, a minha ponta está aqui e eu vou girar ele para vir para o outro lado, então eu vou espelhar ele, e se eu espelhar, a gente vai chegar naquele 160 graus, porque é $180 - 20$ graus, então eu estou espelhando o meu ângulo de 20 graus. Olha que interessante, 160 graus e 20 graus, tem 0.34 de seno, ou seja, eles têm o mesmo seno, e eles tem o valor absoluto do cosseno igual. Os dois tem 0.93 de cosseno, só que 160 graus está em menos ponto 93 de cosseno, porque ele está do lado esquerdo do nosso círculo, ele está para esquerda do ponto 0.

[10:01] O que acontece se eu pegar os mesmos 20 graus e agora espelhar para baixo, ao invés de inverter ele pelo eixo Y, eu vou inverter ele pelo eixo X. Então, eu vou espelhar ele para baixo. Se eu fizer isso, eu vou ter ali do 0 ou 360, lembrando que o círculo, na volta inteira são 360 graus, então a gente pode medir ou 0 ou 360 nesse ponto aqui. E $360 - 20$ eu vou ter 340, o que aconteceu agora, quando eu espelhei o meu ângulo pelo eixo X? Eu vou ter o mesmo cosseno, 340 e 20 graus tem mesmo cosseno, exatamente o mesmo, eles tem 0.93.

[10:41] Só que agora o seno que espelhou, o seno de 20 é 0.34, e o seno de 340 é -0.34. A gente inverteu o seno, quando a gente espelhou no eixo X. E se eu inverter agora no eixo Y e no X ao mesmo tempo? A gente vai ter o ângulo lá no terceiro quadrante. Se a gente contar aqui, a gente tem, o primeiro quadrante é esse, o segundo quadrante é esse e esse é o terceiro quadrante.

[11:04] E o quarto quadrante, onde a gente está vendo ali o 340 graus. Então, no terceiro quadrante, quando a gente espelha o nosso ângulo no eixo X e no Y, a gente vai ter $180 + 20$, ou seja, tem a mesma diferença de 20 graus, e eu espelhei agora o cosseno e o seno, em valor absoluto eles são iguais ao seno e cosseno de 20., só que eles estão negativos.

[11:27] Então, a gente consegue espelhar todos os ângulos que a gente tem em relação a qualquer quadrante. Como que a gente vai fazer isso? Bom, se a gente consegue espelhar, eu consigo falar que o cosseno de 20 graus é igual ao cosseno de 60, se eu ignorar o sinal, ou seja, se eu pegar o módulo dele. Essa barrinha aqui que está em volta do 160 graus, é o operador de módulo na matemática, ele faz com que eu ignore o sinal, ou seja, o cosseno de 20 graus, que é 0.93, é igual

o módulo do cosseno de 160. Ou seja, eu pego o cosseno de 160 e ignoro o sinal, que vai me retornar 0.93, a gente consegue usar isso para espelhar o 160 lá no primeiro quadrante.

[12:08] O seno a mesma coisa, eu consigo falar que o seno de 20 graus, é igual ao módulo do seno de 340 graus. Porque como eu espelhei ele para baixo, em relação ao eixo X, eu espelhei 20 graus em relação ao eixo X, eu preciso inverter o seno, porque foi o seno que virou negativo, o cosseno vai ser sempre igual. Então, com toda essa matemática, com todo esse seno e cosseno, com círculo trigonométrico, como que a gente vai usar essas propriedades, que a gente viu que a gente consegue espelhar os ângulos, para fazer o nosso código funcionar com menos verificações possíveis?

[12:40] Bom, eu tinha lá o nosso círculo trigonométrico, e eu quero verificar só o primeiro quadrante. Apesar de ter todos os ângulos ali que é o ângulo que a gente tinha definido, 20, 70, 110 e 160. Então, o que eu quero é verificar só o primeiro quadrante. E se eu pegar um ângulo aleatório, de qualquer ponto do meu círculo trigonométrico, como que eu faço para espelhar ele para o primeiro quadrante?

[13:02] Vamos pegar aqui o ângulo de 237, é um ângulo qualquer, funcionaria com qualquer ângulo, então 237. Se eu pegar o seno 237 e o cosseno de 237, eu vou ter -0.83 no cosseno e -0.54 no seno. Agora, se ele está invertido no seno e no cosseno, eles dois tem um valor negativo, eu preciso transformar isso em valor positivo, para que eu tenha um ângulo lá no primeiro quadrante. Então, fazendo isso, eu vou ter um seno de 0.54 e um cosseno de 0.83.

[13:34] E qual que é o ângulo que tem esse seno e esse cosseno? A gente vai ver que 237 graus lá no terceiro quadrante, eu posso espelhar ele, ele que equivale a 57 graus no primeiro quadrante, e aí eu consigo fazer a verificação só de onde eu quero. Mas vamos deixar isso de lado e vamos pensar, eu preciso saber o ângulo exato? Não, eu não preciso saber exatamente o ângulo que eu quero, eu só preciso saber quando que eu tenho que começar a verificar o eixo X e o eixo Y na minha animação. Porque lembra, a gente estava verificando, no nosso código a gente tinha lá, se direção.x for maior que 0 ou direção.x for menor que 0, aí eu retornava uma string, no caso direita ou esquerda.

[14:16] Então, a partir de que momento que eu tenho que fazer essa verificação se eu tenho, se agora eu tenho essa folga desse ângulo? Eu não preciso fazer sempre né? É mais fácil fazer essa verificação só quando eu precisar. Para isso, a gente tem o círculo trigonométrico, e como que eu vou fazer uma verificação? Bom, a partir de que momento que eu quero verificar a minha direção.x, o nosso vetor direção.x?

[14:39] Então, a partir de que momento que eu quero que minha jogadora só olhe para cima ou para baixo, ou seja, que eu saiba quando ela está olhando para cima e para baixo? Bom, vamos lá lembrar dos nossos ângulos, eu tenho dois ângulos que a gente definiu como sendo a folga para a nossa animação, ou seja, entre 0 e 20, a minha jogadora está olhando só para direita, entre 20 e 70 ela está olhando na diagonal. Então, direita e esquerda ou cima e baixo, porque a gente está pegando qualquer ângulo aqui e comparando com com esses valores aqui, e entre 70 e 90 graus, ela vai estar olhando só no eixo Y.

[15:11] Então, eu posso falar que vou verificar meu eixo Y, ou seja, só vou verificar se a animação da jogadora está indo para cima ou para baixo, a partir do momento que aquele ângulo que eu selecionei, do meu círculo trigonométrico, e espelhei no primeiro quadrante, for maior do que 20. Na verdade, isso porque entre 0 e 20, a minha jogadora só vai estar olhando para direita ou para esquerda, eu não quero verificar o eixo Y aqui.

[15:34] Eu quero verificar 20 e 70, porque aí ela vai estar na diagonal, então eu preciso saber se ela está indo para cima ou para baixo, e para esquerda ou para direita, e entre 70 e 90 eu só quero verificar o eixo Y. Então, o que todos os ângulos acima de 20 graus tem em comum? Todos eles têm um seno maior do que o seno de 20, porque se a gente for subindo aqui no nosso ângulos, e a gente está projetando sempre os ângulos no eixo Y, todos vão estar acima dessa linha que a gente tracejou aqui, que é do seno de 20.

[16:03] Então, o seno de 20 graus, ou o seno do ângulo que a gente escolher para ser essa folga, vai ser um seno mínimo, e aí a gente vai pegar todos os ângulos, ou seja, qualquer ângulo que o meu mouse formar com a jogadora, eu vou

espelhar lá no primeiro quadrante, e vou verificar se o seno dele for maior do que esse seno mínimo.

[16:19] Então, a gente tem toda essa área aqui, entre 20 e 90, que é onde eu quero fazer minha verificação em Y. No código vai ficar uma coisa mais ou menos assim, se o módulo do seno de X, então, se eu pegar o seno do meu ângulo, tirar um módulo, porque eu quero ele lá no primeiro quadrante, se ele for maior do que o seno mínimo, aí eu vou fazer a verificação de qual animação que tenho que tocar na direção. y. E quando que eu tenho que fazer a verificação da animação na direção.x? Quando que eu tenho que verificar se ele está indo para esquerda ou para direita?

[16:50] Eu quero verificar se a minha jogadora está indo para esquerda ou para direita, quando os meus ângulos forem menores do que 70, porque de novo, entre 0 e 20, ela está indo só para direita ou só para esquerda. Entre 20 e 70 ela está indo para esquerda e para direita ou para cima e para baixo, porque ela está na diagonal. Acima de 70 graus ela está só para cima, então o eixo X não me interessa. Então, de novo, o que todos os ângulos abaixo de 70 graus têm em comum?

[17:17] Todos os ângulos abaixo de 70 graus, tem um cosseno maior do que o cosseno de 70, porque de novo, se a gente pegar os ângulos e for vindo aqui e for projetado eles no nosso eixo X, a gente vai ver que todos os ângulos abaixo 70, tem um cosseno maior do que o cosseno de 70. Então, no final, o nosso cosseno de 70 vai ser o cosseno mínimo. A gente vai ter o cosseno mínimo de X, dá folga para o eixo Y, que é o seno mínimo, e o cosseno mínimo, que é a nossa folga no eixo X.

[17:47] Então, a gente escolheu 70 graus como sendo a folga para o eixo Y, e a gente vai ver aqui esse cosseno mínimo, para verificar esse eixo. E aí a gente também tem entre 0 e 70 graus. Todo esse pedaço do nosso círculo, desse primeiro quadrante, é onde a gente vai fazer a verificação no direção.x, se a nossa jogadora está indo para direita ou para esquerda. No final, com essas duas verificações, no código a gente vai ter, se o cosseno de X, se o módulo do cosseno de X, então eu pego o cosseno do meu ângulo, tiro o sinal dele porque eu quero ele lá no primeiro quadrante, se ele for maior que o cosseno mínimo, aí eu vou fazer a verificação da direção.x

[18:26] E aí no final eu vou ter alguma coisa assim, eu vou tentar fazer a verificação do meu ângulo a partir do seno mínimo e do cosseno mínimo. Então, se eu pegar um ângulo aqui do segundo quadrante e espelhar o seno dele, a gente vai ter um seno maior do que o seno mínimo nesse ângulo. Se eu pegar esse mesmo ângulo e espelhar o cosseno, a gente vai ter alguma coisa maior do que o cosseno mínimo.

[18:51] Então, o nosso ângulo é como se ele tivesse sendo verificado lá no primeiro quadrante, a gente tem muito menos verificação. Isso vai funcionar para qualquer ângulo. Se pegar aqui alguma coisa do terceiro quadrante e tirar os sinais do seno e cosseno, eu vou ter um ângulo equivalente a ele aqui no primeiro quadrante. É isso que a gente vai fazer em código daqui a pouquinho.