



Gerar uma posição Aleatória

Transcrição

[00:00] Com os nossos scripts aqui organizados e o nosso cenário já grandão, o que eu posso fazer aqui? Eu posso, por exemplo, pegar os geradores de zumbi antes só ficavam aqui atrás e jogar eles divididos pelo cenário. Então, eu vou duplicar um qualquer aqui, por exemplo, esse último, Ctrl + D e vou arrastar eles por aí, jogar eles pelo nosso cenário.

[00:23] Coloquei mais geradores pelo cenário, vou jogar a game aqui para baixo, para a gente ver a game e a cena ao mesmo tempo, e vou dar um play aqui para a gente ver o que acontece. Dei um play aqui, a barra de vida bugou, a gente já soluciona isso e eu vou dar um pause aqui agora.

[00:39] Olha o que está acontecendo, os zumbis estão saindo dos geradores de posições completamente diferentes, já que cada gerador está em um lugar, e estão indo diretamente para o personagem, como se eles soubessem exatamente onde que o personagem está.

[00:53] Isso não é tão bom. Parece que o computador é mais inteligente do que a gente, ele fica perseguindo a gente o tempo todo. O ideal seria que o zumbi ficasse parado, ou fazendo qualquer coisa e quando ele ver o personagem aí sim ele persegue, o que não é o que está acontecendo aqui. Vamos fazer isso agora?

[01:12] Só vamos resolver o bug da vida rapidamente, que é por que? Porque o meu status aqui, a vida está zerada, e no start ele fala que a vida é igual vida Inicial. Aí sim ela recebe um valor, só que start está rodando depois do start da interface. para esse cara rodar antes, vamos colocar ele no awake, aí salva, vamos esperar a Unity computr aqui para a gente, vamos dar um play e solucionado.

[01:40] Vamos esperar o zumbi atacar, tudo ok. Agora vamos pegar o nosso zumbi, e fazer ele ficar fazendo qualquer coisa, até ele ver o nosso personagem. Vamos abrir aqui o script do zumbi, que é o controla inimigo. E aí aqui eu tenho o seguinte, olha, meu zumbi, se a sua distância for maior do que 2.5, você me persegue, se não, se for menor do que isso, você me ataca.

[02:06] Eu quero fazer o seguinte, olha zumbi, se a sua distância, e eu já tenho essa variável de distância, que é aqui a distância entre eu e o jogador, o zumbi e o jogador. Se a distância entre o zumbi e o jogador foi maior do que 15, eu quero fazer outra coisa, eu quero por exemplo ficar parado, ou vagar, ou fazer alguma coisa, ficar vagando por aí, fazendo qualquer menos perseguir o jogador.

[02:35] Só que se eu testar esse cara e depois esse cara, eles estão sendo testados ao mesmo tempo, e os dois vão ser verdadeiros, porque a distância pode ser maior que 15 e pode ser maior do que 2.5, por exemplo, 20 de distância é maior do que 15 e maior do que 2.5. E se eu quiser testar esse cara, e se esse cara não conseguir ser testado, eu quero passar para esse cara, porque a gente já viu o if else para fazer isso.

[02:59] Eu posso fazer uma cascata desse, e partir para esse e partir para esse? Posso, é só colocar um else aqui, aí eu crio um um else if. Aí eu testo se a distância é maior do que 15, se não for, se for por exemplo 14, aí ele passa para a linha de baixo, aí ele testa se a distância é maior do que 2.5., se não for, ele cai na linha de baixo. Aí eu posso fazer isso indefinidamente, eu posso ter else if, else if, elase if, e por aí vai.

[03:23] Aqui eu vou fazer o nosso zumbi agora fazer qualquer coisa, enquanto ele não vê o jogador, que é essa distância maior do que 15, ou seja, ele só vai ver o jogador quando ele tiver numa distância menor do que 15. Eu joguei um número aqui, daqui a pouco a gente testa se esse número é bom. Eu vou criar um método para guardar, o que ele vai fazer?

[03:43] Eu vou fazer ele vagar por aí, já criei a chamada do método aqui, basta eu então criar o método. Vai ser o void vagar e o nosso zumbi vai ficar vagando por aí, o que significa ele ficar vagando? Ele vai nascer, aí ele vai achar uma posição perto dele qualquer, e vai para lá, ele vai ficar andando sozinho por aí, por algumas posições meio aleatórias, aí se o personagem chegar perto dele, ele vê o personagem e persegue.

[04:12] Então vamos vagar o nosso zumbi, a primeira coisa que a gente tem que fazer então é gerar uma posição aleatória, para ele poder perseguir. Então, vou criar uma variável aqui, vai ser o `privat vector 3`, posição `vector 3`, posição aleatória, e eu vou gerar uma posição aleatória aqui dentro do vagar. para gerar uma posição aleatória, eu vou criar um outro método, por que?

[04:43] Porque é bom os métodos simplesmente terem uma função, o vagar faz o que? Vaga o zumbi. Ele não tem que criar posição aleatória, nem nada do tipo, ele vai fazer o nosso zumbi vagar. Criar a posição aleatória pode ser um outro método, então eu vou criar o void aleatorizar uma posição.

[05:04] Agora eu vou fazer o seguinte, aqui dentro desse cara, eu vou aleatorizar uma posição, vai ser, uma posição é o `vector 3`, e como que eu faço para gerar uma posição aleatória a partir de onde o meu zumbi está? O meu zumbi está nesse local aqui, suponhamos, vou dar um `play` aqui para gerar algum zumbis.

[05:22] Pronto, o meu zumbi está nesse local aqui. Como que eu faço para gerar uma posição aleatória baseada nesse local dele, eu tenho que gerar uma posição é somar com o transform ponto position dele. Então, eu vou gerar uma posição aqui, como que eu vou gerar uma posição aleatória? Eu vou fazer o seguinte, eu vou usar o random ponto inside Unity esfera. O que esse cara faz?

[05:51] Ele vai pegar uma esfera de raio 1, ou seja, eu tenho uma esfera aqui que vale 1, do centro até a borda, e vai pegar. Eu vou randomizar dentro dessa esfera, eu vou aleatorizar dentro dessa esfera, uma posição qualquer, ou seja, eu posso ir para cima, para um lado, para o outro, para baixo. Dentro de um raio qualquer ali, de uma esfera de 1, eu posso gerar uma posição, ele pega e gera e pronto.

[06:14] Então tem uma esfera que ele vai gerar uma posição dentro dela qualquer para gente, só que 1 distância é muito pouco. O zumbi vai andar 1 passo? Não, vamos colocar uns 10 aqui, uns 10 de distância. Ele pega uma esfera que é maior. Agora tem raio 10, do centro até a borda vale 10, e dentro dela ele vai gerar uma posição, sei lá, aqui, essa é a posição dentro dessa esfera.

[06:39] Gerei minha posição aleatória, agora eu tenho que somar a minha posição com o transform ponto position do personagem, para eu gerar uma posição aleatória baseada em onde o nosso personagem está. Eu vou cancelar essa posição no `Y`, porque a esfera pode ir para cima e pode ir para baixo. Eu não quero que ele fique para cima e para baixo, eu quero que ele fique sempre no chão.

[07:03] Então eu vou cancelar o `y` dessa posição que a gente gerou aqui, que vai ser igual, o `y` vai ser igual a transform ponto position ponto `y`. Vai ser onde eu estou agora nesse momento, no `Y`, então gerei aqui uma posição aleatória cancelando `Y`. O que eu posso fazer agora? Eu posso gerar aqui uma variável lá, posição aleatória, e nela passar a posição que eu gerei aqui dentro.

[07:34] Só que olha, essa variável aqui, ela é uma variável que só funciona aqui dentro, como que eu puxo essa variável aqui para fora? Eu vou mostrar para vocês como fazer métodos com retorno, que são muito interessantes e muito importantes aqui na parte de criação de games. É o seguinte, eu tenho essa posição e eu quero passar ela para fora, eu uso um retorno. Então, eu escrevo a palavra `return` e eu vou sair para fora a posição, vou mandar para fora a posição.

[07:59] A posição é o que? É um `vector 3`, ele deu erro aqui, por que? Porque olha, ele fala que o método é o seguinte, olha, esse método é do tipo void, ou seja, ele não deveria retornar nada.

[08:12] Então, a gente tem que mudar o tipo do método, a gente vai colocar vector 3, por que? Porque eu não estou dando return no vector 3, que é a minha posição? Então, o tipo do método tem que vector 3. Se tivesse retornando um inteiro, o tipo do método ia ser int, se eu tivesse retornando float, o tipo do método ia ser float. Como que eu uso isso agora?

[08:31] Eu posso rodar esse método como se ele fosse uma variável. Eu posso vir aqui e falar, olha, posição aleatória é igual ao método aleatorizar posição, aí que esse cara vai fazer? Rodar o método e me devolver um vector3 que eu vou guardar nesse cara aqui, pronto.

[08:49] Agora eu já aleatizei a minha posição. Vamos fazer o zumbi ir para essa posição aleatória? Que que eu vou fazer então? Aqui eu tenho a direção que o meu zumbi trafega por aí, que ele vai, essa é a direção que ele vai. Eu vou declarar essa variável lá em cima., então vai ter um private vector 3 direção, para poder utilizar essa mesma variável aqui dentro do vagar.

[09:12] Então eu vou tirar esse vector três daqui, porque eu já declarei ela lá em cima, então eu posso chamar só a direção, e essa linha aqui da direção, eu vou colocar aqui dentro do Elsys, por que? Porque eu quero perseguir o jogador só quando a minha distância for maior do que 2.5.

[09:30] Quando ela for maior do que 15, a direção não pode perseguir jogador, ela tem que perseguir quem? A posição aleatória. Agora vou fazer o seguinte. Olha zumbi, você vai pegar e você vai, gerou a posição aleatória e a direção vai ser a posição aleatória menos o meu transform ponto position, é como se a gente tivesse perseguindo alguém mas não é mais o jogador, é uma posição aleatória, beleza?

[09:56] E eu vou copiar a mesma linha de movimentar aqui para dentro, pronto. Agora ele está vagando, ele vai aleatorizar uma posição e aí ele vai pegar essa direção e vai movimentar o nosso zumbi. Vamos salvar isso e vamos ver o que acontece? Vamos olhar os zumbis que estão mais longe, por exemplo, esses de cá. Vamos dar um play aqui.

[10:25] Os zumbis estão todos doidos aí, por que? Porque eu estou gerando uma posição aleatória no update, o tempo. Estou gerando uma posição aleatória, então não sabe para onde que ele tem que ir, que é algo que a gente vai corrigir daqui a pouco.