

01

Refatorando a Movimentação

Transcrição

[00:00] Nossa jogo já está começando a ficar bem complexo, e por causa disso a gente já tem bastante scripts, e alguns scripts, como por exemplo o controla jogador, são bastante grandes. Vamos fazer nos scripts que a gente já tem basicamente o que a gente fez no scripts de controla interface, quebrar o script em métodos. A gente também pode quebrar o scripts em mais scripts, por que que eu digo isso? Olha só, vou mostrar um caso para vocês, presta atenção nesse script de controla inimigo.

[00:28] Esse script de controla inimigo tem essa parte de movimentação. Se eu entrar no controla jogador e dar uma olhada na movimentação do controla jogador, vocês vão ver que olha só, as duas são basicamente a mesma coisa. Toda vez que a gente olha um código e tem a mesma coisa em dois lugares diferentes, é porque a gente pode extrair isso em outro local e poder utilizar a partir deste local, que é o que a gente pode fazer agora.

[00:56] É uma boa prática não escrever a mesma coisa igual em dois lugares. Então, vamos começar a refatoração dos nossos scripts, dar uma olhada neles e ver o que a gente pode modificar nos nossos códigos. Vamos começar pelo controla inimigo, que é um script menor para a gente pegar o fluxo. Já de cara eu já vejo que eu não estou usando o update, então eu posso apagar esse update aqui, salvei, nosso script já fica um pouquinho mais enxuto.

[01:22] E aí vamos começar com a parte de movimentação, então eu vou pegar essa parte de movimentação aqui, que a gente está utilizando no controla inimigo e no controle jogador, e se a gente pegar essa parte de movimentação e tirar desse script e colocar em outro script, aí no controla inimigo e no controla jogador, eu chamo do mesmo local. Os dois vão usar o mesmo script de movimentação.

[01:43] Será que a gente pode fazer isso? Com certeza., então vamos fazer exatamente isso, vamos criar um novo script aqui, novo código aqui, então create C# script e vamos dar o nome de movimento personagem, dar o nome aqui de movimento personagem. Abri esse código, vamos dar um reload aqui na solução, vou apagar tudo dele, a gente só vai ter a classe, que é o nosso script em si, vou até descer essa chave aqui, apaguei start, apaguei update.

[02:17] Salvei, vamos aqui agora com esse script que vai movimentar, como a gente vai começar pelo controla inimigo, vamos lá no nosso prefab do inimigo, personagem, zumbi, está aqui, e jogar o script para dentro dele. Então, a gente pode adicionar o componente, vou tirar essa busca aqui, do tipo scripts e eu vou jogar o movimento personagem. Pronto, está aqui já, ela está aqui abaixo do nosso rigid body.

[02:44] Esse script vai lidar com a movimentação, então eu posso copiar essa parte de movimentação aqui, copiar essa parte e jogar para esse script. Aí vamos jogar isso então dentro de um método para a gente organizar script, porque se eu jogar aqui do lado de fora, vai dar erro em tudo, então tenho que sempre que eu for colocar linha de código, dentro de métodos. Então, vamos criar o void movimentar aqui, e vou jogar esse aqui para dentro.

[03:19] Aí já deu alguns erros aqui no nosso void movimentar, deu primeiramente erro no rigid body que ele não existe, deu erro neste rigid body aqui novamente porque não existe, deu erro na direção porque ela não existe e deu erro na velocidade. Então, a gente tem que gerar essas variáveis aqui dentro. Vamos começar pelo rigid body, o movimento da personagem está no nosso inimigo e o nosso inimigo tem um rigid body, então a gente pode fazer uma variável privada, do tipo rigid body e eu vou dar o nome dela de rigid body. Por quê?

[03:49] O rigid body antigo chamava - do nosso outro script, do controla inimigo - rigid body inimigo, só que esse cara aqui, o movimenta personagem, ele vai estar em todos os personagens do nosso jogo. Ele pode estar no jogador, pode estar no inimigo, pode estar em qualquer outro zumbi que a gente quer criar, qualquer outro personagem, então não

adianta eu dar o nome rigid body inimigo porque esse cara não pode estar só no inimigo. Então, eu vou dar um nome genérico, rigid body.

[04:12] Não é bom sempre fazer isso, mas nesse caso a gente não tem uma forma de tirar esse tipo de comportamento, de chamar de um nome genérico, o ideal é você ter um nome que condiz. Inclusive a gente pode mudar de rigid body para meu rigid body, para não ficar algo assim, rigid body.

[04:31] Que rigid body? O meu rigid body, objeto que tem esse script. Criamos aqui o meu rigid body, e aí o que a gente fez antes é ir lá no start, por exemplo, no start, vou declarar start aqui, falar que o meu rigid body é igual a getcomponent rigid body, porque ele está buscando rigid body do objeto que tem esse script. Aí a gente troca o rigid body inimigo por meu rigid body, o rigid body inimigo por meu rigid body.

[05:05] Dois erros já solucionados, só que aqui a gente tem um problema, esse cara o movimenta personagem, eu estou utilizando ele para movimentar os nossos personagens gerais no jogo. E se eu quiser no start do controla inimigo, movimentar um personagem, aí o que eu tenho que fazer?

[05:22] Se esse código aqui tiver no start, eu não sei em que ordem que a Unity vai rodar isso, eu vou rodar esse start primeiro ou o outro start? Aí se eu quiser movimentar o meu personagem no controla inimigo aqui e esse start aqui não rodou ainda, vai dar um erro porque eu ainda não tenho rigid body. Então, a gente vai colocar isso no método que roda antes do start, que é o awake. Salva isso aqui, deixa a Unity computar, vai dar erros aqui, está dando erro no nosso console, vamos continuar solucionando.

[05:53] A direção e a velocidade, eu não tenho aqui mesmo, então eu vou trazer pelos parâmetros do nosso método, então eu vou criar por vector3 direção e o float velocidade. A velocidade eu vou colocar com v minúsculo, porque essa velocidade não é mais uma variável pública nem nada do tipo. Prontinho, agora no nosso controle inimigo eu posso utilizar esse script.

[06:20] Então vou criar um privat do tipo movimenta personagem, porque vai valer esse cara aqui, que eu vou chamar de meu movimento personagem. Ou então algo melhor, movimenta inimigo, pronto, que era o que estava antes no nosso rigid body. Tem aqui o movimenta inimigo, vamos aqui no start agora dar um getcomponent para o movimenta personagem.

[06:59] Já tenho um movimenta inimigo aqui, agora a gente pode substituir aquela três linhas de movimentação para um, já estou com uma variável que vale esse script aqui, então posso chamar ela movimenta inimigo ponto, aí eu podia chamar o método movimentar. Não apareceu aqui por que? Porque o método não é público, então public movimentar.

[07:21] Aí pronto, agora movimentar e eu passo os meus parâmetros, eu vou passar a direção e vou passar a velocidade. Lembrando que mesmo que essas variáveis direção e velocidade tenham o mesmo nome, aqui podia ser qualquer coisa, desde que aqui dentro eu chamassem essa qualquer coisa. Lembra que a programação é espertinha? Ele sabe, você mandou um vector3, e eu preciso de um vector 3, está aqui no parâmetro, eu troco esse valor da direção pelo vector 3 que você mandou.

[07:51] Eu preciso de um float, você mandou um float? Troca esse valor. Tanto que aqui o V é maiúsculo e lá o V é minúsculo. Pronto, agora a gente já arrumou isso, vamos fazer a mesma coisa aqui com a rotação agora, que a rotação a gente também está com esse problema. Então, eu vou extraer essas duas linhas, Ctrl + C nelas, vamos criar aqui agora o public void rotacionar e vamos colar as linhas de rotação que são essas duas.

[08:20] Eu preciso de uma direção e eu preciso de um rigid body. O rigid body eu já tenho, que é o meu rigid body e a direção eu ainda não tenho, então vector 3, direção que eu vou passar por parâmetro. Agora o que a gente pode fazer? Apagar isso aqui e chamar movimento inimigo ponto rotacionar passando a direção que é uma variável que a gente criou aqui em cima.

[08:47] Passando a direção, ele vai jogar essa direção para o nosso, para essa variável, e vai fazer a mesma coisa com a rotação. Então, eu criei um script que tem 2 métodos, o rotacionar e o movimentar, e chamei eles no controla inimigo. Você viu que já ficou bem sucinto. O rotacionar já está aqui e o movimentar eu já quebrei aquelas linhas que a gente tinha em uma coisa só.

[09:10] Agora a gente tem o script movimentar que eu posso utilizar em qualquer personagem do nosso jogo. Utilizando os parâmetros e passando parâmetros pelo método. Essa forma de fazer é bem legal, eu poderia por exemplo aqui nesse script, simplesmente ter um void update, um void fixed update e fazer a movimentação em outro script, né, nem ter a movimentação aqui no controla jogador.

[09:33] Eu gosto dessa forma que a gente fez porque o seguinte, eu trato os meus scripts como se fossem componentes da Unity. Porque olha só, eu tenho o meu rigid body aqui, que a Unity me forneceu, e dentro do controla inimigo eu estou chamando o rigid body. Inclusive a gente nem precisa mais dele porque a gente não está utilizando ele mais para nada.

[09:55] Vamos apagar aqui. Dentro do meu controla inimigo eu tava puxando o rigid body para uma variável e usando os métodos do rigid body. Então, eu estou fazendo a mesma coisa, dentro do meu controla inimigo eu estou puxando o movimento personagem e utilizando os métodos do movimento personagem.

[10:12] estou fazendo como se esse script aqui fosse um componente da Unity. Esse cara é um componente da Unity que eu estou utilizando em um código meu, estou fazendo basicamente o que a Unity faz para a gente. Vamos dar um play e ver se está tudo certo com os nossos zumbis, eles ok, rotacionando e me perseguindo. Aí dessa forma a gente deu uma refatorada na movimentação do nossos personagens.