

03

Refatoração

Transcrição

[00:00] Chegou a hora de utilizarmos nosso herói dentro do nosso código. No encontra jogador é o primeiro momento do nosso foge onde utilizamos o herói, onde encontramos ele. Se eu encontrei, não vou mais retornar um array de duas posições que representa o herói. Tenho uma estrutura que representa ele. O herói.new é o meu jogador. Configuro tanto o jogador.linha, que é nossa linha para valer, quanto a coluna. Ao invés de retornar esse array, retornamos nosso herói. O valor referenciado pela variável jogador. Uma estrutura que tem os dois dados lá dentro, a linha e a coluna.

[01:16] Já que estamos usando a nossa classe herói, precisamos requerer ela. No momento em que refatoramos, alteramos o código do encontra jogador para deixar de usar aquele array de dois valores e passamos a usar um objeto do tipo classe herói, estamos quebrando a compatibilidade de quem chama esse método, essa função.

[01:51] Se o jogador perdeu, ele vai verificar se é nulo ou não. Não tem problema. Mas também dentro do joga verificamos quem é nosso herói e fazemos alguma coisa com ele. O herói é utilizado na hora de calcular uma nova posição e na posição do mapa. O herói pega a posição da linha e da coluna e usa essa posição para zerar a posição do mapa. Nossa herói antes era um array de duas posições. Agora ele é um objeto do tipo herói. Ele tem linha e coluna com atributos, porque na classe herói definimos attr acesso linha e coluna.

[02:50] No calcula nova posição, também passamos o herói. Duplicamos nosso objeto com os valores de lá de dentro, pego o movimento adequado e mudamos a posição zero, que não é mais zero, é linha e coluna. É o mesmo objeto que recebi como argumento? Não, porque duplicamos ele. Quando copiamos os valores da linha e da coluna temos outro objeto que foi referenciado pela variável herói e é nele que alteramos a linha e coluna e devolvemos, como fazíamos antes com nossa array, só que agora temos um objeto em duas posições. Alteramos tudo.

[03:56] Recapitulando. No encontra jogador, ao invés de devolver um array, criamos uma variável que instância um herói novo e esse herói é referenciado por essa variável. Certa linha, certa coluna dele, estamos felizes. Com isso quebramos a compatibilidade. Todo mundo que chamava encontra jogador, ou seja, dois lugares do nosso jogo, usavam um array, e agora estamos verificando se o valor é nulo ou não. No joga, fazíamos duas coisas com o herói. Calculávamos nova posição e usávamos a posição dele para colocar um vazio no mapa. Antes era zero e um, agora é uma estrutura que tem coluna e linha.

[04:50] Repare que mudamos de array para outro tipo de objeto e a única coisa que tivemos que fazer é pegar os valores que acessávamos de uma maneira e passamos a acessar de outra maneira. Essa foi nossa refatoração. Teoricamente, não nos preocupamos com o tipo, mas quando refatoramos e mudamos o tipo completamente, todo mundo que dependia daquele tipo teve que ser refatorado também.