

## Armazenamento

### Transcrição

[00:00] Que código horrível esse de posições válidas. Eu concordo. Vamos melhorar. Sempre dá para melhorar cada vez mais. Repare que temos as posições que são variações de somas da nossa posição. Já vimos isso em algum lugar. Só que diferente de antes, quando calculávamos a posição do herói, agora queremos passar pelos quatro caras possíveis.

[00:52] Vou armazenar todos os movimentos possíveis. Para baixo é o mais um zero. Para direita, é o zero mais um. Para cima, menos um zero. E para a esquerda, zero menos um. São todos os movimentos possíveis. Para cada um dos movimentos, vamos usar o each.

[01:36] Temos agora que calcular a nova posição. É a posição zero mais o movimento, zero, vírgula, posição um, mais movimento um. Se essa posição é válida no novo mapa, colocamos a nova posição. Senão não colocamos e vamos para o próximo movimento. No final, temos todas as posições. Estamos passando por todas as posições possíveis e verificando se elas são válidas.

[02:33] Vamos extrair a soma de vetor também. Vou passar o movimento e a posição. Depois, defino soma vetor, que recebe vetor um e vetor dois. Ele vai devolver um novo vetor, que é o vetor um, zero, mais vetor dois, zero, vetor um, mais vetor dois um.

[03:02] Poderíamos fazer de outras maneiras. Esse é o suficiente para somar meu vetor de uma dimensão, dois elementos.

[03:22] Os fantasmas estão andando, tudo funcionando, e eu refatorei posições válidas a partir de, tendo uma lista de movimentos possíveis. Poderíamos fazer o fantasma se movimentar que nem um cavalo em um jogo de xadrez.

[03:46] O cavalo sempre se movimenta duas casas em uma direção e uma na outra. Ou duas casas na outra direção e uma na mesma. Duas casas numa direção, uma na outra. Duas casas numa direção, uma na outra. E aí você coloca todas as combinações e o fantasma passa a se movimentar da mesma forma. Seria bem desafiador.

[04:30] Com a mesma técnica que usamos antes, consigo implementar o movimento dos jogadores de uma peça de xadrez. No nosso caso, não vamos fazer isso.