

05

Array

Transcrição

[00:00] Resumindo, nós implementamos um movimento simples de um fantasma, que é ir para a direita, e encontramos vários bugs. Por exemplo, o fantasma não pode ultrapassar a parede ou muro. Usamos a função que tínhamos criado de posição válida. O fantasma não pode entrar em contato com outro ou com meu usuário. Refatoramos a função posição válida após implementar essa lógica.

[00:28] Usamos uma array associativa para refatorar o movimento do próprio herói e não ter o case cheio de condições. As condições são agora interpretadas através da array associativa, que é um mapa, um dicionário. Palavras e termos que usamos para a mesma estrutura de dados.

[00:47] Vimos também a implementação do duck typing na prática. O que acontece é que nós como programadores estamos interessados no tipo, porque ele me diz os comportamentos distintos e existentes naquele objeto, naquele valor, e me diz o que cada um desses comportamentos faz. Não só que existe um cara com aquele nome, mas o que ele faz. Portanto, para nós, desenvolvedores, o tipo é importante. Porém, para o interpretador não importa se o tipo é A ou B, só importa que o comportamento existe. Essa é a grande vantagem.