

## Objetos que bloqueiam ou não o movimento

Conseguimos criar os dois tipos diferentes de elementos (chave e porta) no nosso jogo. Nosso único problema é que a porta ainda não está bloqueando o movimento da heroína. Como no momento estamos verificando a colisão com os objetos depois de atualizar o movimento do jogador, não conseguimos impedir o movimento dele pois só descobrimos que tem uma colisão depois de ter andado! Para corrigir isso, vamos ter que alterar a função `tentaMoverPara()` para verificar primeiro se tem alguma colisão com objetos e depois verificar a colisão com o mapa. Além disso, vamos ter que alterar também as funções que verificam colisão para indicar pra gente se teve ou não alguma colisão. Por exemplo, quando o jogador colide com uma porta, temos dois casos possíveis: se ele tem chave não tem colisão, caso contrário aí tem colisão. Vamos ao passo a passo de como implementar tudo isso.

- Altere a função `fazColisaoDoJogadorComAChave()` para fazer ela retornar o valor `false` para indicar que o movimento não deve ser bloqueado quando tiver colisão com a chave.
- Altere a função `fazColisaoDoJogadorComAPorta()` para fazer ela retornar o valor `true` ou `false` para indicar quando o movimento do jogador deve ser bloqueado. Se o jogador tiver chaves, ele abre a porta e não tem colisão logo precisa retornar `false`. Caso contrário, a porta se mantém fechada e o movimento tem que ser bloqueado então devolvemos `true`.
- Altere a função `verificaColisaoComObjetos()` para retornar um valor `true` ou `false` indicando se o movimento deve ou não ser bloqueado. Lembre-se que as funções de colisão com a chave e a porta já devolvem esse valor então quando chamamos cada uma delas, só precisamos devolver o mesmo valor que elas devolvem. Caso não haja colisão com objetos, aí a função tem que devolver `false`.
- Ainda nessa função, faça ele receber uma `novaPosicao` como parâmetro que representa a nova posição do jogador. Altere o código para considerar essa nova posição na chamada a função `temColisao()` ao invés de usar a posição atual do jogador.
- Altere a função `tentaMoverPara()` e crie uma nova variável local `novaPosicao` com atributos `x` e `y` para guardar a posição para onde o jogador está tentando se mover. Aqui é só pegar a posição atual do jogador e somar com os deslocamentos `x` e `y` que a gente tem no `delta`.
- Nessa mesma função, logo após criar a variável `novaPosicao`, chame a função `verificaColisaoComObjetos()` passando a `novaPosicao` como parâmetro e verifique o valor que ela retorna. Se for `true`, teve colisão e precisa bloquear o movimento então simplesmente encerre a função com o comando `return`. Caso contrário, não precisa fazer nada e o código segue para verificar a colisão com o mapa.
- Finalmente, retire a chamada a função `verificaColisaoComObjetos()` que existe na função `TIC()`.

Execute o jogo, tente primeiro abrir a porta sem ter uma chave e confira se o movimento é bloqueado. Depois, pegue a chave e tente abrir a porta novamente. Agora a porta deve se abrir e o jogador tem que conseguir seguir em frente.

