

Melhoria do código da colisão com objetos

Chegou a hora de melhorar novamente o nosso código antes que ele cresça demais e a gente não consiga mais entender o que cada parte está fazendo. Nossa vilã dessa vez é a função **verificaColisaoComObjetos()**. Repare que a gente tem um monte de `if` onde a única coisa que muda é o tipo de objeto e a função que chamamos. Para a chave, tem que chamar a função de colisão com a chave. Para a porta, tem que chamar a função de colisão com a porta. Muito parecido, certo? Em Lua podemos guardar valores em variáveis mas também conseguimos guardar funções! Isso significa que a gente consegue guardar dentro de cada objeto qual a função de colisão que ele usa. Assim a gente consegue usar essa informação para melhorar o nosso código e deixá-lo mais genérico e mais fácil de adicionar novos objetos. Vamos a nossa tarefa:

- Em cada objeto que criamos até agora (chave, porta e inimigo), crie um atributo a mais chamado `funcaoDeColisao` e guarde nesse atributo a função que tem que ser chamada para a colisão daquele objeto. Por exemplo, no objeto chave, uma parte do código que a gente tem na função **criaChave()** ficaria assim: `local chave = { funcaoDeColisao = fazColisaoDoJogadorComAChave, ... }`, mantendo os outros dados que a gente já tinha na chave. Lembre-se de fazer isso com todos os objetos que criamos até agora.
- Altere a função **verificaColisaoComObjetos()** para que ela chame a função de colisão do objeto quando houver colisão. Agora você pode tirar o `if` que verifica o tipo do objeto e simplesmente chamar `objeto.funcaoDeColisao(indice)` para fazer a colisão com o objeto.

Como em todas as refatorações que fizemos até agora, rode o jogo e teste novamente as colisões pra saber se a gente não quebrou nada com as alterações.