

Kit Médico

Transcrição

[00:00] Vamos continuar a trabalhar no jogo do nosso curso. Primeiro, vamos ver o que fizemos até agora. Nossos zumbis ficam vagando até chegarmos perto, e eles vêm nos perseguir. O jogo como está hoje depende muito da habilidade do jogador. Se ele for melhor ou pior vai variar muito. Fica monótono. Talvez fosse legal incluir elementos aleatórios para melhorar o fluxo. O jogador pior pode usar esses elementos para fazer com que ele tenha um fluxo mais legal, e o jogador melhor pode utilizar para melhorar ainda mais.

[01:17] Um desses elementos é que nosso jogo tem só uma chance. E se ele pudesse ganhar vida, com um elemento aleatório? Para fazer isso, temos que ter algo palpável. Tem um prefab bem interessante para isso que é o kit de primeiros socorros. Vou jogar para o nosso cenário e arrastar para perto do jogador. Não deixe longe do personagem quando produzir porque o jogador vai ter que sair andando para pegar. E sempre que você for testar algo, a melhor forma é que seja perto, a não ser que seja necessário estar longe.

[02:38] No game, o kit é um pouco pequeno. Poderíamos aumentar. Agora, para fazer ele recuperar vida vamos usar script. Já vimos algumas formas de fazer isso com a bala. Seria exatamente o que vamos fazer aqui. Se eu encosto no objeto, eu destruo ele. Para o jogador, parece que ele pegou, mas nós sabemos que na verdade ele pegou e destruiu o objeto. Vamos precisar de uma colisão, para saber se estou tocando no objeto.

[04:00] Se eu não marcar o trigger, quando o jogador for pegar o kit médico, vai parecer que é uma parede. Com o trigger, não tenho colisão física, mas ela é computada. É o que eu quero. Vou criar um novo script, marcar que vai ser do tipo C Sharp, e colocar o nome. Já crio adicionado ao objeto. Não gosto muito dessa forma porque o script fica fora da pasta, e o projeto acaba ficando bagunçado. Eu acho mais fácil sempre criar e arrastar para o objeto, para ficar organizado.

[05:52] Já vimos que podemos usar o void com trigger, assim, quando o jogador tocar no kit médico, vai rodar o OnTriggerEnter, porque nosso objeto é um trigger. Se não for, você tem que usar o OnCollisionEnter.

[06:55] Se só o jogador vai poder ser curado pelo kit, vou perguntar se o objeto tem a tag jogador. Se tiver, vou destruir o kit médico. Mas ainda precisamos fazer com que o kit cure o personagem.

[08:20] Mantendo o fluxo do que fizemos nos cursos passados, quando fomos no controlsJogador, criamos uma interface de Imatavel no nosso método de tomar dano. Podemos criar outra interface seguindo o fluxo de organização, que vai ser o Icurável, para curar vida. Vou definir uma assinatura, que vai ser do tipo void, e vou passar a variável de quanto de vida vamos curar. Lembrando que tem que ser public, porque vamos chamar o método em outros locais.

[10:26] Esse método vai pegar a mesma coisa que o tomar dano e fazer exatamente o inverso. Vou pegar o statusJogador.Vida e somar com a quantidadeDeCura. Agora, no kit médico podemos dizer que o objetoDeColisao vale o jogador. Vou pegar esse objeto de colisão, um componente dele, que é o ControlaJogador, e vou chamar o método curarVida, passando um valor que podemos definir numa variável. Esse último é melhor definir como privado, porque não queremos que cada kit médico cure um valor. Se fizermos isso vai ser aleatoriamente.

[12:16] Quando meu kit médico colide com o nosso jogador ele cura 15 de vida, e depois é destruído. Só que ainda temos que atualizar na interface.

