

01

Introdução

Transcrição

[00:00] Agora nós vamos criar um jogo novo. O jogo da forca, em que escolhemos uma palavra aleatória e o jogador tem que adivinhar qual foi a palavra. Para isso, ele pode chutar letras, com até cinco chances, ou chutar a palavra em si. No total, ele tem cinco chances para acertar a palavra, isto é, ele pode tentar quatro coisas, e na última é melhor tentar a palavra.

[00:30] Primeiro, vou fazer uma função de dar boas-vindas. Uma função de dar boas-vindas vai perguntar o nome do nosso usuário e dar boas-vindas para ele. Vou salvar o meu arquivo antes de tudo.

[00:45] Vamos escrever “puts” e digitar o que queremos. O nome vai ser igual a `gets.strip`. Vou tirar o enter, qualquer outro caractere em branco que exista no final ou no começo da minha string, vou retornar meu nome, mas antes vou mostrar uma mensagem de boas-vindas. E aí temos a função. Claro que vamos invocar ela, com nome sendo igual a `da_boas_vindas`.

[01:30] Nossa próximo passo é escolher uma palavra. Então `def` vai ser `escolhe_palavra_secreta`. Vai escolher e devolver a palavra secreta, que por enquanto vai ser programador, por exemplo. Vou novamente colocar o “`puts`” e a mensagem. Nas reticências, vou colocar o número de letras da minha palavra secreta, o `size`, `#(palavra_secreta.size)`. Chama a função `size` no valor que tá referenciado pela palavra secreta, que é o programador.

[02:32] Vamos ter que `palavra_secreta = escolhe_palavra_secreta`. Nós vamos escrever também uma função para perguntar se o usuário quer jogar novamente, com sim ou não como resposta, `quero_jogar = gets.strip`, `nao_quero_jogar = quero_jogar.upcase == "N"`.

[03:12] E nosso laço principal do jogo? Como fica? Loop do end. Aqui dentro vamos jogar o jogo, e logo depois vamos perguntar se ele não quer jogar mais. Se não quiser jogar, `break`. Isso é, joga e passa o nome como parâmetro.

[03:45] Mas todo jogo que eu jogar vai usar a mesma palavra secreta? Não, cada vez escolhemos uma diferente. Ela vai ser definida dentro do nosso jogo, na hora de jogar. Isso é, vou definir uma função chamada “joga”, que recebe o nome, e lá dentro vou escolher a palavra secreta. Não só isso, tenho que falar quantas vezes eu já errei. Eu errei zero vezes até agora. Não chutei nenhuma vez e fiz zero pontos. No final de uma rodada, vou imprimir uma mensagem e falar quantos pontos ganhei até agora.

[04:40] O que faltou é testar várias vezes para ver o que o usuário digita, ver o usuário chutar uma palavra, comparar se ele acertou, se a letra tá presente na palavra. Precisamos implementar o jogo da forca. Meu usuário tem cinco vidas, cinco tentativas. Isso é, quero executar o código do meu jogo enquanto os erros forem menores do que cinco. Quando ele errar a quinta vez, acabou o jogo. Ele pode executar o jogo então cinco vezes para adivinhar a palavra secreta. O jogo funciona pedindo um chute para o usuário, e adicionamos na lista de chutes o chute que acabamos de chutar.

[05:32] Claro que temos que verificar se ele acertou ou errou. Daqui a pouco fazemos isso. Primeiro vamos perguntar qual o chute do nosso usuário. Isso é, definir uma função chamada “`pede_um_chute`”, que vai pedir um chute para o nosso usuário. Quatro linhas em branco, quantos erros você cometeu até agora, lembrando de receber o número de erros. Vou mostrar seus chutes até agora, para que você não se esqueça, e aí peço para ele entrar com uma letra ou palavra. Temos o `chute = gets.strip`.

[06:27] Eu vou retornar esse chute, mas antes tenho que verificar se ele acertou. Estamos mostrando para ele o número de erros e os chutes que ele já fez, confirmando o chute e retornando, para adicionar à lista dos chutes. Quando

pedimos um chute, precisamos passar para ele os chutes atuais e os erros atuais

[07:00] Vamos testar nosso jogo no terminal. Por enquanto, só estamos lendo o chute. Tudo que vimos até agora são conceitos da linguagem, conceitos de programação que já aprendemos. O que temos que fazer é verificar se acertamos ou erramos.