

Jogo da adivinhação

VISÃO GERAL

Foi feito um programa chamado de “Jogo da adivinhação” entre o cliente e o servidor onde, o servidor será responsável por gerar um número aleatório de 0 a 100 e o cliente deverá tentar descobrir ele com um determinado número de tentativas e toda vez que o cliente tentar um determinado número o servidor vai responder de acordo com o número dado, se o número for o correto, vai aparecer a mensagem “Parabéns! Você acertou o número”, caso for o número errado, o mesmo vai dar dicas se o número é maior ou menor, caso seja um número muito alto ou muito baixo vai imprimir “Número inválido. Informe um valor de 0 a 100”, entretanto se o jogador esgotar as chances vai aparecer a mensagem que as chances acabaram e vai voltar para o menu inicial se o cliente decidir sair do servidor ele irá voltar para a tela inicial do lado do cliente.

O jogo segue a seguinte ideia: a máquina sorteia um número aleatório de 0 a 100 e o jogador tem 5 chances para acertar o número sorteado. O jogo utiliza sockets TCP em Python para estabelecer a conexão entre o cliente e o servidor. Quando a conexão é estabelecida, o servidor cria uma thread para cada jogador, garantindo que cada usuário jogue individualmente, sem interação com outros jogadores conectados.

LÓGICA DO CÓDIGO

Para a execução desse programa existem dois códigos `server.py` e `client.py`, os dois códigos são responsáveis por implementar o jogo de forma simples onde o cliente deve adivinhar um número aleatório escolhido pelo servidor. A relação entre os códigos se dá da seguinte forma:

1. O servidor inicia a execução, criando um socket TCP e esperando por conexões de clientes na porta 50000.

2. Quando um cliente se conecta ao servidor, o servidor cria uma nova thread para lidar com esse cliente e gera um número aleatório entre 0 e 100.
3. O cliente inicia a execução, exibindo um menu inicial que permite ao usuário escolher entre logar no servidor e sair do programa.
4. Se o usuário escolhe logar no servidor, o cliente tenta se conectar ao servidor usando o endereço IP e a porta informados pelo usuário. Se a conexão é bem-sucedida, o cliente entra em um loop onde solicita ao usuário que digite um número entre 0 e 100 e envia esse número para o servidor.
5. O servidor recebe o número enviado pelo cliente e compara com o número aleatório que foi gerado. Se os números são iguais, o servidor envia uma mensagem de parabéns para o cliente e encerra a conexão. Se os números são diferentes, o servidor envia uma mensagem informando se o número aleatório é maior ou menor que o número enviado pelo cliente, e diminui a quantidade de chances restantes. O servidor continua nesse loop até que o número seja adivinhado corretamente ou as chances acabem.
6. O cliente recebe a mensagem enviada pelo servidor e exibe para o usuário. Se o número adivinhado é igual ao número aleatório gerado pelo servidor, o cliente encerra a conexão. Se as chances acabam e o número não foi adivinhado, o cliente encerra a conexão.
7. Após encerrar a conexão com o servidor, o cliente exibe uma mensagem de despedida e volta para o menu inicial.

EXPLICAÇÃO DE FUNÇÕES

Nos códigos, as funções estão organizadas de forma que conecte um código ao outro, com cada função tendo uma responsabilidade específica e bem definida.

No código do servidor, a função **handle_client** é responsável por lidar com a conexão de um cliente, gerar um número aleatório e processar as mensagens recebidas pelo cliente, verificando se elas são válidas e se o número escolhido pelo cliente é igual ao número gerado aleatoriamente.

A função **main** do código do servidor é responsável por obter um endereço IP válido e uma porta para o servidor, e iniciar o servidor, aguardando por conexões de

clientes. A função **main** chama a função **handle_client** em uma nova thread para cada cliente que se conecta ao servidor.

Já no código do cliente, a função **client** é responsável por se conectar ao servidor, enviar mensagens e receber respostas do servidor, e encerrar a conexão quando o jogo terminar. A função **menu** do cliente exibe um menu com duas opções: logar no servidor ou sair do programa. Se a opção de logar no servidor for escolhida, a função **main** é chamada para iniciar o processo de conexão com o servidor.

Portanto, podemos dizer que a função **main** do servidor e a função **client** do cliente estão conectadas, já que a primeira chama a segunda para iniciar a comunicação entre cliente e servidor. Além disso, a função **handle_client** do servidor é chamada em uma nova thread a cada nova conexão de cliente, e a função **client** do código do cliente é executada enquanto o usuário não escolher sair do programa no menu exibido pela função **menu**.

No contexto da teoria de redes, podemos dizer que esse programa é um exemplo de aplicação que utiliza o protocolo TCP/IP, uma vez que utiliza sockets TCP para estabelecer a comunicação entre o cliente e o servidor. O servidor fica em um endereço IP e porta especificados, aguardando conexões de clientes. Quando um cliente se conecta, é estabelecida uma conexão TCP entre o cliente e o servidor, permitindo que mensagens sejam trocadas entre eles.