Jogo de Xadrez

Daniel Martins Reis

July 30, 2016

Abstract

Developing a client application - server socket using a chess game with two players developed in C language Basically, it is used as a base some networking concepts to build the application, which uses up a TCP / IP connection by sending the protocols network, these being developed and interpreted by the application itself.

1 Introdução

O presente trabalho refere-se a uma aplicação cliente - servidor, sugerida na disciplina CSI 426 - Fundamentos de Redes, utilizando socket. Dessa maneira, desenvolveu-se basicamente um jogo de xadrez. Assim, tem-se na aplicação o tratamento das regras do jogo em questão, tratando-se de dois jogadores, que nesse caso são clientes que ficam concectados ao servidor.

2 Comunicação

Um Socket é um ponto final (endpoint) de um canal bidirecional de comunicação entre dois programas rodando em uma rede. Sendo assim, é a maneira mais popular de utilizar as funcionalidades de comunicação TCP/IP. Assim, tem-se um servidor que fica apenas "ouvindo" o Socket, aguardando um pedido de conexão do cliente. Dessa maneira, o cliente sabe o nome do host e a porta associada a aplicação servidora, daí solicita uma conexão, que após ser aceita pelo servidor, é criado um novo socket para atender as requisições.

Para permitir que processos se comuniquem na troca de dados, criou-se um protocolo, que nesse caso é formado pelas informações necessárias para proporcionar o funcionamento do jogo. Assim, transmite-se: gamer, linha original, coluna original, linha e coluna em que o jogador escolheu para a peça se locomover, além de permutações de valores, nesse caso somas entre esses, tornando-se assim, um meio de verificar a integridade do pacote enviado na rede.

Para utilização do protocolo, foi preciso concatenar todas as informações descritas anteriormente, dentro de um vetor de char, que após ser lançado no servidor é recebido pelo outro cliente, que aplica as operações de testes para verificação do pacote, que ao perceber que o mesmo veio correto, atualiza então a matriz do gamer. Vale ressaltar, que quando a mensagem é recebida, é preciso separar as informações que antes estavam concatenadas, sendo então preciso utilizar um método split, pois as informações enviadas, eram agrupadas e separadas

por um delimitador, nesse caso uma vírgula, e com o split, ou seja, quebra-se a string em várias, sendo elas separadas pelo delimitador em questão.

É importante ressaltar, que tanto o servidor como o cliente são capazes de receber e enviar informações. Dessa maneira, tem-se verificações para validação de jogadas em ambos, tornando o jogo consistente.

3 Código

Para representação do tabuleiro, utilizou-se uma matriz de valores inteiros, no qual o intervalo 1 a 6 representa peças do gamer 1, 7 a 12 do gamer 2 e 13, espaços em branco. Após o jogador selecionar uma peça, então explora-se todos os possíveis movimentos a partir da peça selecionada. Assim, tem-se métodos para definir quais são esses movimentos válidos, sendo eles ligados a qual tipo de peça foi selecionada. Vale ressaltar que caso a peça selecionada não apresente movimentos válidos, o jogador fica obrigado a escolher outra. Dessa maneira, basicamente marca-se a posição na matriz com o mesmo valor mas com um peso negativo, definindo-se assim a ideia de que após testar, sabe-se que o gamer só poderá se locomover para tais posições marcadas com um peso negativo. Ao fim da jogada, retoma-se os valores da matriz para peso positivo. Outro ponto importante a ser lembrado, é que quando a peça selecionada é um peão, é preciso testar também se existe uma peça do adversário na diagonal, não só apenas mover para frente. Também, a aplicação não permite ao gamer selecionar uma peça do adversário e nem selecionar uma peça que não tenha nenhum movimento válido. Nesse jogo, tem-se que cada tipo de peça possue o seu conjunto de regras associadas, como por exemplo o peão, dito anteriormente. Assim, existe um método de verificação que identifica qual peça foi selecionada, pra daí redirecionar a verificação para o seu tipo correto, sendo que cada tipo possue um método próprio, exclusivo.

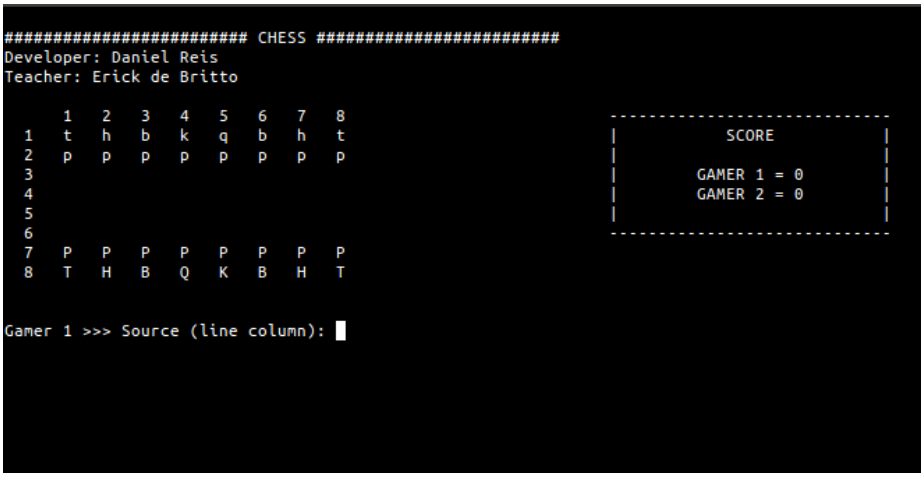

Figure 1: Interface.

Na interface, como mostrada na figura acima, utiliza-se o próprio terminal, sendo que as peças em letra maiúscula representam o gamer 1, e as minúsculas o gamer 2. Assim, também mostra-se as posições com peso negativo com um

traço antes do nome da peça, simplesmente para poder mostrar ao usuário quais são os seus movimentos válidos com peça selecionada a partir da localização da mesma.

4 Considerações

Ao desenvolver esse game utilizando redes é possível entender como realmente funciona uma aplicação cliente - servidor usando sockets, possibilitando também abstrair melhor alguns conceitos definidos na disciplina. Dessa maneira, vale ressaltar que por ser uma simples aplicação, sendo apenas um jogo, optou-se por não criptografar os dados, até porque os mesmos não tem importância caso sejam capturados, pois nesse âmbito, percebe-se a real importância de criptografar os dados, que por mais que não se teja conectado ao servidor, ao enviar as informações pela rede pode ser que exista alguém no modo escuta ali, ou seja, capturando todos pacotes que transitam na mesma, e logicamente, tentando os interpretar. Nota-se também que pacotes podem perder integridade ao serem enviados na rede, mostrando-se assim a real necessidade de testar se os mesmos chegaram, sendo de forma correta ou não. Nessa aplicação, por utilizar pacotes tão pequenos, não percebe-se esse fenômeno.