

Redes de Computadores

Relatório do projeto AB2 | 2020.2

Kamila de Almeida Benevides - kab@ic.ufal.br

Samuel Lima da Silva - sls@ic.ufal.br

Principais funcionalidades da aplicação

O trabalho desenvolvido foi utilizar os conceitos de redes na implementação do jogo da velha. Esse jogo consiste de duas pessoas que jogam alternadamente, preenchendo cada um dos espaços vazios. Cada espaço é preenchido com X ou O. Vence o jogador que conseguir formar primeiro uma linha com três símbolos iguais, seja ela na horizontal, vertical ou diagonal.

A implementação foi desenvolvida na linguagem *Python*, e com biblioteca *pygame*.

Descrição do protocolo

O jogo consiste de dois jogadores, também chamados de clientes. O jogo só inicia quando os dois clientes são encontrados. O servidor e cliente trocam mensagens que contém comandos que ambos devem executar ou ficar cientes. Abaixo está a lista de comandos:

- INQUEUE: O cliente foi colocado na fila de espera. O servidor está buscando outro cliente para iniciar a partida.
- START: Os dois clientes foram encontrados e o jogo vai se iniciar.
- YOURTURN: O cliente que recebe esse comando é o que deve fazer uma jogada.
- OTHERTURN: O cliente que recebe esse comando é o que deve esperar a jogada.
- WINNER: O cliente que recebe esse comando venceu o jogo.
- LOOSER: O cliente que recebe esse comando perdeu o jogo.
- DRAW: A partida deu empate.
- OKAY: O servidor executou algum comando com sucesso.
- REDO: O cliente que fez a jogada gerou algum erro no servidor e deve fazer uma jogada válida novamente.
- MOVE idx_i idx_j: O cliente que vai fazer a jogada envia esse comando ao servidor para indicar qual lugar do tabuleiro marcou. O servidor envia esse comando aos dois clientes para indicar qual lugar do tabuleiro foi marcado pelo cliente que fez a jogada. idx_i e idx_j são os índices da linha e coluna, respectivamente, marcadas.

As mensagens são cadeias de caracteres. Cada comando na mensagem é separado por uma quebra de linha. O único comando que admite espaços em branco é MOVE. Os espaços desse comando são usados para separar idx_i e idx_j. Abaixo um exemplo de mensagem no formato adequado:

MOVE 1 3

OKAY

Abaixo estão descritos cada ação que pode ser executada e as trocas de mensagens envolvidas. As ações em cada item são executadas na ordem que aparecem.

1. Cliente se conecta ao servidor: O servidor recebe a conexão. Verifica se existe algum jogo na fila de espera. Se existe jogo em espera, atribui o jogador à aquele jogo e inicia o jogo e retorna ao cliente uma mensagem com comandos START e OKAY. Se não existe jogo em

espera, o servidor cria um novo jogo, associa o jogador à aquele jogo, coloca o jogo em espera e envia ao cliente uma mensagem com comando INQUEUE.

2. Início do jogo: O servidor atribui ao primeiro cliente encontrado a jogada do primeiro turno. Início do turno: O servidor envia uma mensagem com comandos OKAY e YOURTURN para o cliente que vai fazer a jogada. O servidor envia uma mensagem com comandos OKAY e OTHERTURN para o cliente que vai esperar a jogada. Então, o servidor fica esperando pela mensagem MOVE do cliente que vai fazer a jogada.
3. Cliente fez sua jogada: Quando o servidor recebe a mensagem com comando MOVE do cliente que fez a jogada, ele verifica se a jogada foi válida e se o cliente ganhou a partida ou deu empate.
4. Fim de turno: O servidor envia aos dois clientes uma mensagem com comandos OKAY e MOVE. O comando MOVE contém os índices da posição escolhida pelo jogador que fez a jogada naquele turno. O servidor inverte os papéis dos clientes. Quem esperava pela jogada agora faz a jogada e quem faz a jogada agora espera. Então, o servidor inicia um novo turno.
5. Fim do jogo: Se no passo 3 o cliente tiver ganhado a partida, o servidor adiciona na mensagem que vai ser enviada no item 4 o comando WINNER, ao cliente que fez a jogada, e adiciona para o cliente que estava esperando a jogada o comando LOSER. Se deu empate, o servidor adiciona nas mensagens dos dois clientes no item 4 o comando DRAW. Então, o servidor finaliza o jogo e encerra a conexão com os dois clientes.
6. Cliente gerou algum erro no servidor: O servidor envia aos dois jogadores uma mensagem com comando REDO e inicia o próximo turno. Os erros que o servidor consegue detectar são: cliente marcou alguma posição inválida no tabuleiro, índice fora do range permitido, posição já marcada, índice não numérico ; cliente deveria ter enviado uma mensagem com comando MOVE e não enviou ou a mensagem não veio no formato adequado.

O servidor foi projetado de forma que possa executar N jogos em paralelo com, consequentemente, $2 \cdot N$ jogadores conectados ao servidor. O número N obviamente depende dos recursos computacionais disponíveis na máquina.

O que poderia ter sido implementado a mais

Poderia ter sido adicionado uma funcionalidade de verificação de prontidão. Quando é encontrado dois jogadores para um jogo, a partida se inicia imediatamente. Nenhum dos dois jogadores têm tempo suficiente para confirmar que estão preparados para iniciar a partida. Poderia ter sido implementado um tratamento com timeout para identificar quando um cliente ainda permanece conectado ao servidor.

Dificuldades encontradas no desenvolvimento do projeto

Poderia ter melhorado alguns mecanismos do código do cliente. Quando o cliente está esperando a resposta do servidor, o sistema operacional entende que o processo do cliente parou. Realmente aconteceu isso, porém era esperado.