



Atividade 5: Trabalho Final

Aluno: Guilherme Luis Domingues

RA: 155619

Instituto de Computação

Universidade Estadual de Campinas

Campinas, 08 de Janeiro de 2020.

Sumário

1	Estrutura geral do trabalho	2
2	Fases do desenvolvimento	3
3	Dificuldades encontradas	4
4	Executando o programa	5

1 Estrutura geral do trabalho

A arquitetura geral do trabalho ficou da seguinte maneira:

- Servidor
 - Recebe conexão TCP com os clientes
 - Controla quais cliente estão jogando e quais estão disponíveis
 - Ao final da partida, recebe o resultado da mesma e incrementa no ranking
- Cliente
 - Recebe conexão UDP do seu oponente
 - Detém a regra do jogo
 - Ao final, retornar o resultado para o servidor

Desta forma, a visão geral do projeto fica:

1. Cliente se conecta ao servidor via TCP. O servidor gera um ID e retorna para o mesmo;
2. Cliente recebe lista de jogadores disponíveis para jogar;
3. Cliente requisita o início de jogo, dentre os disponíveis;
4. Ambos os jogadores recebem o endereço de seu oponente: Jogador 1 se conecta via UDP ao Jogador 2 e vice-versa;
5. Ao final da partida, o vencedor retorna o status para o servidor;
6. Ambos jogadores voltam para a lista de disponíveis para jogar.

2 Fases do desenvolvimento

Primeiramente, iniciei o desenvolvimento buscando a comunicação TCP entre o Cliente e o Servidor. Uma vez esta comunicação feita, começamos a elaborar a lógica descrita acima. O código base desta etapa foi o do Exercício 4, o qual utilizava Pool/Select para segurar mais de um cliente ao mesmo tempo.

Finalizada a comunicação, partimos então para a criação da lista de jogadores em aguardo. Todo jogador que se conecta ao servidor vai automaticamente para essa lista, ficando visível para os demais jogadores e recebendo a lista. Para iniciar o um jogo, o jogador deve enviar para o servidor o ID do adversário. A princípio não é necessário o oponente aceitar o jogo. uma vez escolhido, a conexão entre eles deve ser estabelecida.

Com os jogadores já conectados um ao outro, a lógica do jogo deve começar a ser utilizada. A rodada 1 começa com quem solicitou o jogo. O jogador deve entrar uma posição do tabuleiro, representado pela sequência de A até I, como mostra a Tabela 1. Assim que o Jogador 1 inputar seu primeiro movimento e, uma vez que o mesmo é válido, a jogada foi confirmada. O Jogador 1 envia a sua jogada para o Jogador 2, e perde seu turno. O Jogador 2, por sua vez, inputa a jogada, ela é validada e então enviada novamente para o Jogador 1. Assim consequentemente até existir um vencedor ou dar velha (empate).

A	B	C
D	E	F
G	H	I

Tabela 1: Representação do Tabuleiro.

3 Dificuldades encontradas

Apesar de ter a estrutura pensada, acabei encontrando alguns problemas durante a implementação do exercício.

A primeira delas é lidar com diferentes retornos do servidor ao cliente utilizando Pool. Pelo desenvolvimento planejado, uma vez que o cliente conecta ao servidor ele deve receber a lista dos jogadores disponíveis para jogar. Esse envio está sendo feito apenas depois que o cliente envia 2 mensagens para o servidor e não consegui encontrar o porque.

Além disso, não consegui entender como um cliente se conectaria a outro. Por pesquisas, vi que cada cliente funcionaria como um P2P, ou seja, cada cliente acaba sendo um servidor também. O que pensei em fazer, em passo a passo, era:

- Receber o endereço do oponente;
- Desconectar ambos os jogadores do servidor;
- Abrir um Socket UDP do Jogador 1 para o Jogador 2;
- Iniciar a lógica do jogo, seguindo até terminar;
- Ao finalizar o jogo, abrir novamente a conexão TCP do Jogador com o servidor;
- Com esta conexão aberta, enviar o resultado para o servidor;
- Servidor recebe este resultado, armazena no arquivo e retorna os jogadores para a lista de jogadores disponíveis;

