Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Telecomunicações e Informática

Unidade Curricular de

Sistemas Distribuídos

Ano Lectivo de 2016/2017

Serviço de Jogo Pedra-Papel-Tesoura

Gilberto Morim, A65214

Junho, 2017

Índice

1. Introdução

No contexto da unidade curricular Sistemas Distribuídos, foi-nos proposta a elaboração de um Serviço do Jogo de Pedra-Papel-Tesoura, no qual o cliente interage com o servidor. O objectivo deste projecto é fazer um *match* entre dois jogadores que pretendem jogar, servindo concorrentemente vários pares de jogadores. Este trabalho foi implementado usando a linguagem Java pondo em prática os exercícios que foram dados nos guiões das aulas, utilizando ferramentas como threads para efetuar várias funções concorrentemente ou sockets para fazer com que seja possível jogar em sistemas terminais diferentes ligados em rede.

2. Desenvolvimento

2.1. Contextualização

O objetivo do trabalho prático é simular um serviço de jogos de Pedra-Papel-Tesoura, que permita as seguintes operações por parte dos seus clientes:

• Iniciar uma sessão entre dois jogadores que pretendam jogar

• Receber as jogadas de cada jogador

• Devolver aos jogadores o resultado

2.2. Estrutura do Programa

- **Cliente**

  O programa cliente implementa três classes em Java: a classe Cliente, cuja função é iniciar o programa pelo qual o cliente irá interagir, abrir a socket de comunicação entre o cliente e o servidor, e iniciar duas threads que implementam as duas outras classes: ThreadedReadFromServer (responsável por ler todas as mensagens que o servidor envia ao cliente, e imprimi-las para o ecrã do cliente – mais especificamente, lê o resultado do jogo que o servidor envia, e apresenta-o ao cliente), e a outra classe, ThreadedSendToServer (responsável por ler as mensagens introduzidas pelo cliente e enviá-las para a socket de comunicação, e detetar quando o cliente escreve “exit” para fechar a socket)

- **Servidor**

  O programa servidor implementa também três classes: a classe Server, cuja função é iniciar a instância do servidor, e abrir a ServerSocket num determinado porto, à espera de ligações dos clientes, abrindo uma socket em cada uma das ligações. Assim que são detetadas duas ligações, é inicializada, numa nova thread, uma instância da classe ClientHandler, que representa o jogo em si. Esta classe tem acesso às sockets de ambos os clientes, e aguarda que ambos os clientes escrevam a sua opção, e envia as opções lidas para uma instância da classe RockPaperScissor, que vai comparar as jogadas e determinar o vencedor e perdedor, informação que é enviada para o respetivo jogador.

3. Conclusões

A realização deste trabalho decorreu sem grandes percalços, mas foi bastante importante para limar a maioria dos conceitos aprendidos nas aulas práticas da UC, e perceber como todos eles funcionam.

Na minha opinião pessoal, o projeto teve uma dificuldade acessível, mas suficiente para requerer domínio sobre as novas ferramentas aprendidas ao longo do semestre, e para as aplicar e interligar. Acho que seria adequado, no entanto, colocar um requisito mais explícito que envolvesse a sincronização e acessos de threads a zonas críticas.