

REDES DE COMPUTADORES

TRABALHO PRÁTICO I - DOCUMENTAÇÃO

Wilson Moreira Tavares

2014068334

1. INTRODUÇÃO

Este trabalho prático envolve os conceitos trabalhados em sala, sobretudo o protocolo TCP e a biblioteca Socket. Foi implementado um jogo batalha naval que é jogado por um cliente e um servidor, com as informações sendo trafegadas através da rede.

Um socket de rede é o ponto-final de um fluxo de comunicação entre 2 aplicativos através de uma rede. Em cada ponta da comunicação implementada há um aplicativo, cliente e servidor, que compartilham informações. Um socket de rede em um computador é definido como a combinação de:

- um endereço IP
- o número de uma porta

2. JOGO

Foi implementado o jogo batalha naval, a ser jogado por um único cliente contra o servidor. A aplicação cliente é controlada pelo usuário, que escolhe os campos a serem atacados e pode interromper o jogo a qualquer momento utilizando o comando Q+[ENTER]. O servidor, por sua vez, joga de maneira completamente autônoma assim que é instanciado, possuindo uma inteligência artificial simples implementada.

Cada jogador possui um pedaço do oceano, onde sua frota está contida. A frota de ambos é composta pelas seguintes embarcações:

- porta-aviões (tamanho cinco, quantidade 1)
- navios-tanque (tamanho quatro, quantidade 2)
- contratorpedeiros (tamanho três, quantidade 3)
- submarinos (tamanho dois, quantidade 4)

O jogo consiste em uma troca de disparos alternados entre os jogadores, terminando termina quando um dos lados consegue acertar 30 campos adversários, que é a quantidade total ocupada pelas embarcações citadas acima. É exibida na tela de cada aplicação seu respectivo resultado (“Você vendeu!” ou “Você perdeu...”).

Existem instruções para o jogo no arquivo “readme.txt”, como foi definido em especificação.

3. IMPLEMENTAÇÃO

Como foi dito anteriormente, cada jogador possui um pedaço do oceano onde sua frota está contida. Esse pedaço do oceano é representado por uma matriz 10x10 de inteiros com suas respectivas frotas. A presença de uma embarcação em um campo da matriz é representada pela presença do valor 1, enquanto a ausência é representada pela presença do valor 0.

A matriz da aplicação cliente é criada através da leitura de um arquivo no formato .txt simples, chamado "entrada_cliente.txt" e contido no diretório "entrada/". A matriz da aplicação servidor é criada de forma aleatória, respeitando as restrições de sobreposição. Cada componente das frotas pode estar disposto na horizontal ou na vertical.

Cada aplicação possui uma segunda matriz, que representa o que é conhecido do campo adversário. Essa matriz é iniciada totalmente zerada. Ao disparar um tiro contra um campo da matriz, seu valor é atualizado com o valor -1 se não havia uma embarcação no local do tiro e 1 se o tiro atingiu alguma embarcação.

Na aplicação servidor todo o processo é automático, desde a criação da matriz à escolha do local onde os tiros são dados. As matrizes são impressas automaticamente a cada 5 tiros. Quando um tiro acertou o alvo, tenta-se atacar o campo vizinho, na seguinte ordem de prioridade:

- direita
- esquerda
- cima
- baixo

Na aplicação cliente a criação da matriz é feita através da leitura do arquivo e a localização dos tiros é informada via teclado, com tratamento para as entradas que fogem do escopo previsto.

A implementação seguiu os padrões definidos pela especificação. A aplicação servidor faz uso de IPv6, assim como a aplicação cliente. Caso seja passado um endereço IPv4 para a aplicação cliente, é identificado que o endereço é IPv4 e o mesmo é convertido para IPv6.

As mensagens enviadas possuem um padrão de três caracteres, onde:

- o primeiro caractere representa a linha a ser atacada
- o segundo caractere representa a coluna a ser atacada
- o terceiro caractere representa a resposta ao último tiro

Existem duas situações onde a mensagem possui tamanho um:

- O comando Q+[ENTER] encerra o jogo, enviando uma mensagem com apenas o caractere 'Q'
- Quando há um vencedor, uma mensagem com apenas o caractere 'L' é enviada ao adversário

4. TESTES

Foram realizados testes com as seguintes situações:

- vitória do servidor
- vitória do cliente
- campos digitados errados
- campos digitados repetidos
- comandos inexistentes
- abandono do jogo pelo cliente
- impressão de matriz pelo cliente

```
wilson@beat: ~/github-workspace/tp-redes
Criando socket...
Configurando socket...
Atribuindo endereço e porta...
Aguardando cliente...

Sua base naval:

  [0][1][2][3][4][5][6][7][8][9]
[A] 0 1 1 1 1 1 1 1 1 1
[B] 0 0 1 1 1 0 0 0 0 0
[C] 1 1 1 1 1 0 0 0 0 0
[D] 0 1 0 0 1 0 0 0 0 0
[E] 0 1 1 1 0 0 0 0 0 0
[F] 0 1 0 0 0 0 0 0 0 0
[G] 1 1 0 0 0 0 0 0 0 0
[H] 0 0 1 1 1 0 0 0 0 0
[I] 0 0 0 0 0 0 0 0 0 0
[J] 0 1 1 0 0 0 0 0 0 0

-----
Campo atacado: A5
```

Tela inicial do jogo no servidor

```
wilson@beat: ~/github-workspace/tp-redes
Criando socket...
Configurando socket...
Conectando ao servidor...
Concluído!

Sua base naval:

  [0][1][2][3][4][5][6][7][8][9]
[A] 1 1 1 1 0 0 0 0 1 1
[B] 1 1 1 0 0 0 0 0 0 1
[C] 0 0 0 0 0 0 0 0 0 1
[D] 0 0 0 0 0 0 0 0 0 0
[E] 0 1 1 1 0 0 0 0 0 0
[F] 0 1 1 1 0 0 0 0 0 0
[G] 0 0 1 0 0 0 0 0 0 0
[H] 1 1 1 0 0 0 0 0 0 0
[I] 0 0 0 0 0 0 0 0 0 0
[J] 1 1 1 1 0 1 1 1 1 1

Campo atacado pelo adversario: A5

-----
Digite um comando ou o campo que deseja atacar: 
```

Tela inicial do jogo no cliente

Em todos os cenários o software se manteve robusto e soube contornar as situações descritas. Não houveram tempos de espera, lentidões para gerar ou para escolher o campo aleatório por parte do servidor.

A única brecha conhecida é a geração de matriz através de arquivos pelo cliente, que não consegue contornar um arquivo desconfigurado. Há uma verificação sobre o número de uns presente na matriz, mas não há verificações sobre a forma das embarcações da frota serem respeitadas, por ser muito custoso.

5. CONCLUSÃO

A implementação deste trabalho prático foi de extrema importância para afixar e praticar os conhecimentos adquiridos em sala. Os maiores problemas que tive foram mais no que diz respeito à lógica do jogo e outros problemas menores relacionados ao meu longo tempo sem usar a linguagem C.

Acredito que a inteligência do servidor pode ser melhorada, de forma a guardar o último acerto e tentar expandi-lo em todas as direções, não só uma delas. Além disso, a criação da matriz no cliente pode ser melhorada. A leitura de um arquivo é feita, e não há confirmação de que os dados estão estruturados como o jogo exige, passou por minha mente utilizar do método aleatório do servidor, que funciona muito bem e resolveria este problema, porém achei melhor manter como estava sugerido nas especificações.