Documentação - TP1

1. Introdução

O problema proposto foi implementar um jogo de Campo Minado. As regras do jogo são semelhantes ao original, na qual o jogador deve revelar todas as posições no campo que não contenham nenhuma bomba. Caso revele uma bomba, o jogo é encerrado.

Para a implementação do jogo, criou-se um cliente e um servidor. O servidor é responsável por guardar a resposta do tabuleiro e o estado atual do jogo. O cliente é responsável por enviar as jogadas ao servidor e armazenar o histórico de jogadas. A comunicação entre ambos segue o protocolo TCP, sendo compatível com IPv4 e IPv6.

2. Funcionamento

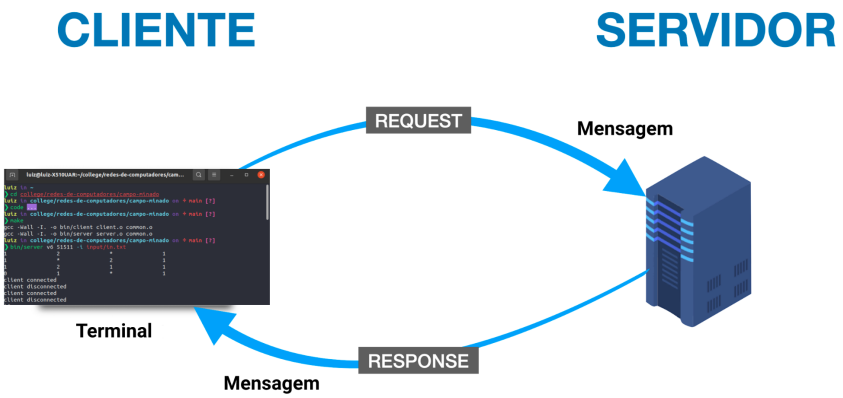
Essa seção visa descrever, de maneira geral, o funcionamento do programa, tanto em relação ao servidor quanto ao cliente.

O servidor do jogo é responsável por armazenar o resultado e o estado atual do jogo. Ele fica aberto a receber apenas uma conexão vinda de um cliente e envia o resultado das ações desse seguindo o protocolo TCP. Ele suporta apenas uma conexão por vez, e um jogo por conexão. Para o cliente conectar no servidor, ele deve acessar o IP e a porta especificados na inicialização do servidor, e após isso, enviar uma mensagem do tipo START.

O resultado do jogo é proveniente de um arquivo que é carregado pelo servidor no momento de inicialização. Sendo assim, o servidor só é capaz de gerar um jogo por inicialização.

O fim da conexão entre cliente e servidor só acontece quando o cliente envia uma mensagem do tipo EXIT, indicando para o servidor fechar a conexão. Caso o cliente queira continuar jogando, ele pode enviar um comando RESET, para reiniciar o estado do jogo para o estado inicial.

O fluxo de comunicação, bem como a estrutura da mensagem que transita entre o cliente e servidor, são descritas abaixo.




```c
typedef struct Mensagem {
    int type;
    int coordinates[2];
    int board[ROWS][COLS];
} Mensagem;
```

3. Discussão

O desenvolvimento de um Campo Minado remoto, permitindo a interação entre um cliente e um servidor através de *sockets* na linguagem C, é um desafio empolgante, mas está longe de ser uma tarefa trivial. Durante o processo de criação desse jogo, diversos empecilhos surgiram, aumentando a complexidade de desenvolvimento do jogo. Neste texto, discutiremos alguns desses desafios que podem ser encontrados ao criar um Campo Minado remoto.

1. Comunicação em Rede:

A comunicação em rede é o pilar fundamental desse projeto, e qualquer problema nesse aspecto pode comprometer seriamente o funcionamento do jogo. A sincronização precisa entre o servidor e o cliente é essencial para garantir que ambos tenham a mesma visão do tabuleiro em todos os momentos. Problemas de latência, perda de pacotes ou desconexões inesperadas podem interromper a jogabilidade e exigir estratégias complexas de tratamento de erros. Apesar de tais pontos não terem sido o enfoque desse trabalho, em projetos reais, eles são muito importantes para manter a robustez do projeto.

2. Sincronização de Estado:

Garantir que o servidor e o cliente mantenham o mesmo estado do jogo é crucial. Problemas de sincronização podem ocorrer se a ordem das mensagens for

alterada ou se informações forem perdidas durante a transmissão. Uma estratégia eficaz de sincronização e controle de versão do tabuleiro é essencial para evitar inconsistências.

3. Tratamento de Erros:

Erros são inevitáveis em qualquer aplicativo de rede. Lidar com erros de forma eficaz e fornecer mensagens de erro claras para os usuários é vital para manter a experiência de jogo positiva. Isso inclui tratar desconexões abruptas, problemas de conexão e erros de protocolo.

4. Fluxo do jogo:

Projetar o fluxo de um projeto que envolve o uso de sockets para comunicação em rede pode ser uma tarefa desafiadora. Isso ocorre porque a arquitetura baseada em sockets introduz complexidades adicionais devido à natureza assíncrona e distribuída da comunicação entre o servidor e os clientes. Nesse projeto não foi diferente. Manter as regras definidas no protocolo e garantir que tanto o cliente como o servidor recebam e enviem as mensagens requeridas foi uma tarefa complexa.

4. Conclusão

O desenvolvimento de um Campo Minado remoto é um desafio que vai além da simples criação de um jogo. Requer conhecimento na área de comunicação em rede, lógica de jogo e tratamento de erros. Superar os empecilhos ao longo do processo de desenvolvimento é essencial para entregar uma experiência de jogo fluida e satisfatória para os jogadores, enquanto proporciona um valioso aprendizado sobre programação e redes.