Aluno: Leonardo D’avila Teixeira Soares

Documentação do Sistema de Jogo de Exploração de Labirintos

Objetivo

O objetivo deste trabalho é criar uma versão de um jogo de Exploração de Labirintos que permitirá a interação entre um cliente e um servidor usando sockets em linguagem C.

Funcionamento do Jogo

No início do jogo, o cliente envia um comando para iniciar a partida e recebe do servidor quais movimentos o jogador pode fazer. O cliente pode então enviar para qual direção quer ir, e o servidor vai atualizar a posição do jogador (caso o movimento seja válido) e retornar quais novos movimentos ele pode fazer. O cliente pode também requisitar ver o mapa, ação que faz com que o servidor retorne o labirinto para o usuário com todas as regiões ocultas, exceto pelas que o jogador já sabe o que são (como caminhos livres ou muros).

O jogo é vencido quando o cliente encontra a saída. O servidor informa ao cliente que o jogo foi vencido, e revela o labirinto com todas as células reveladas (mesmo se o jogador não descobriu todas). A partida é encerrada com sucesso.

Contudo, o cliente pode enviar um comando de encerrar o jogo a qualquer momento, onde após enviar a mensagem vai ser desconectado, e o servidor vai resetar o estado do jogo.

Estrutura do Código

Arquivo common.h

Este arquivo contém as definições de constantes, estruturas e funções comuns utilizadas tanto pelo cliente quanto pelo servidor.

Arquivo common.c

Este arquivo implementa as funções definidas em common.h, incluindo funções de utilidade para manipulação de endereços e comunicação via sockets.

Arquivo server.c

Este arquivo implementa a lógica do servidor, que inclui:

- Inicialização do servidor e espera por conexões de clientes.

- Processamento de comandos recebidos do cliente.
- Atualização do estado do jogo e envio de respostas apropriadas ao cliente.

Arquivo client.c

Este arquivo implementa a lógica do cliente, que inclui:

- Conexão ao servidor.
- Envio de comandos para iniciar o jogo, mover-se no labirinto, visualizar o mapa, e encerrar o jogo.
- Recebimento e processamento das respostas do servidor.

Estrutura de Dados

```
struct action {
    int type;
    int moves[100];
    int board[TAM_MAX_BOARD][TAM_MAX_BOARD];
};
```

Esta estrutura é utilizada para enviar e receber mensagens entre o cliente e o servidor. Ela contém:

- **type**: Tipo de ação (iniciar jogo, mover, visualizar mapa, etc.).
- **moves**: Lista de movimentos possíveis.
- **board**: Estado atual do labirinto.

Funções Principais

Funções no common.c

- **logexit**: Exibe uma mensagem de erro e encerra o programa.
- **addrparse**: Converte uma string de endereço e porta em uma estrutura sockaddr_storage.
- **addrtostr**: Converte uma estrutura sockaddr em uma string de endereço.
- **server_sockaddr_init**: Inicializa uma estrutura sockaddr_storage para o servidor.
- **mapearComando**: Mapeia um comando recebido para um valor do enum Comando.
- **validaMovimento**: Verifica se um movimento é válido.
- **verificaAoRedor**: Verifica os movimentos possíveis ao redor de uma posição no labirinto.

- **obterDimensoes**: Obtém as dimensões do labirinto a partir de um arquivo.
- **inicializarBoard**: Inicializa o tabuleiro do labirinto.

Funções no server.c

- **usage**: Exibe a mensagem de uso do servidor.
- **main**: Função principal que inicializa o servidor e processa os comandos do cliente.

Funções no client.c

- **usage**: Exibe a mensagem de uso do cliente.
- **main**: Função principal que conecta ao servidor, envia comandos e processa as respostas.

Como Executar

Compile os arquivos common.c, server.c, e client.c:

Executar **make**

Inicie o servidor:

./server v4 51511 -i input/in.txt

Inicie o cliente:

./client 127.0.0.1 51511

Comandos do Cliente

- **start**: Inicia o jogo.
- **up**: Move para cima.
- **right**: Move para a direita.
- **down**: Move para baixo.
- **left**: Move para a esquerda.
- **map**: Solicita o mapa do labirinto.
- **reset**: Reinicia o jogo.
- **exit**: Encerra o jogo.

Desafios Encontrados

Durante o desenvolvimento deste sistema de jogo de exploração de labirintos, alguns desafios foram enfrentados, tanto no aspecto técnico quanto na organização do trabalho. São eles:

- Compreender os detalhes da criação, configuração e manipulação de sockets em linguagem C.
- Organizar a lógica do código de forma clara e eficiente foi outro desafio. O sistema precisava ser modularizado adequadamente para garantir que as funções e responsabilidades estivessem separadas de maneira lógica entre os arquivos common.c, server.c e client.c.
- A conciliação do desenvolvimento deste projeto com outras responsabilidades acadêmicas e profissionais foi um dos maiores desafios. Organizar o tempo de maneira eficiente foi essencial para garantir que o progresso do projeto não fosse prejudicado.
- A depuração e os testes do sistema também representaram um grande desafio, pois uma alteração no código que corrigia um fluxo funcional muitas vezes resultava em consequências inesperadas em outras partes, que só eram identificadas após algum tempo.

Conclusão

Este sistema permite a interação entre um cliente e um servidor para jogar um jogo de Exploração de Labirintos. O cliente pode enviar comandos para mover-se no labirinto, visualizar o mapa, e encerrar o jogo. O servidor processa esses comandos, atualiza o estado do jogo, e envia as respostas apropriadas ao cliente.