

Redes de Computadores

Trabalho Final

Campo Minado Cooperativo

Luis Eduardo Rasch & Gabriel Moura

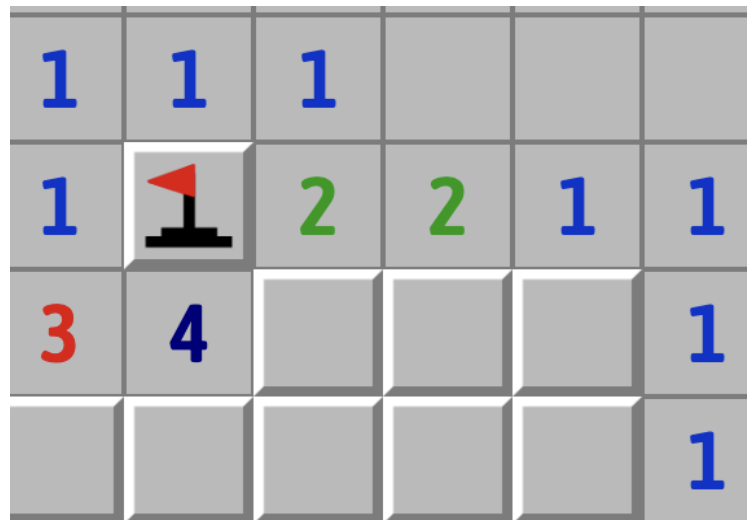


Imagem meramente ilustrativa

1. Objetivo e Função da Aplicação e Protocolo Desenvolvido

O protocolo de aplicação desenvolvido tem como objetivo principal permitir a implementação de um jogo campo minado cooperativo em turnos, onde dois jogadores podem participar simultaneamente de uma mesma partida. A aplicação visa proporcionar uma experiência colaborativa, mantendo a sincronização do estado do tabuleiro entre os participantes conectados durante a jogatina.

A função do protocolo é garantir a comunicação confiável entre os participantes, utilizando uma arquitetura peer-to-peer com coordenação centralizada. Nesse modelo, ambos os computadores são tratados como peers, mas um deles assume o papel de host (servidor lógico) responsável por validar e coordenar todas as ações do jogo, enquanto o outro peer atua como cliente conectado diretamente ao host. Essa abordagem assegura que todas as jogadas sejam processadas de forma consistente e propagadas corretamente para todos os participantes.

2. Características do Protocolo Desenvolvido

- Arquitetura: Peer-to-Peer (P2P) híbrida com coordenação centralizada
- Modelo de Conexão: Sem estado
- Persistência: Não persistente
- Modo de Comunicação: Push
- Controle: Na banda
- Protocolo de Transporte: UDP

A implementação P2P híbrida permite comunicação direta entre participantes mantendo coordenação centralizada para consistência do jogo.

3 e 4. Listagem dos Tipos de Mensagens Possíveis entre os Peers

Tipo de Mensagem	Descrição	Enviado por
handshake	Solicita conexão inicial ao peer host	Peer cliente
handshake_responde	Confirma conexão e envia coordenadas das minas	Peer host
player_ready	Sinaliza início do turno do peer destinatário	Qualquer peer
flag_updated	Comunica mudanças no estado das bandeiras	Qualquer peer
game_lose	Comunica que o peer perdeu o jogo	Qualquer peer
game_win	Comunica que o peer ganhou o jogo	Qualquer peer
disconnect	Notifica desconexão do peer	Qualquer peer

5. Formato de Cada Tipo de Mensagem e Respectivos Campos de Cabeçalho

```
var msg := {  
  >| "type": type,  
  >| "from": get_parent().address,  
  >| "to" : peer,  
  >| "payload": null  
}
```

O campo payload funciona como um contêiner flexível que transporta dados específicos conforme o tipo da mensagem. Para mensagens simples de controle (como handshake, player_ready, game_lose), o payload é definido como null, indicando que não há dados adicionais necessários. Para mensagens complexas que requerem transferência de informações do jogo, o payload contém objetos estruturados ou arrays com os dados pertinentes.

6. Especificação dos Valores Possíveis para Cada Campo de Cabeçalho

Campo	Tipo	Valores Possíveis
type	string	handshake, handshake_response, player_ready, flag_updated, game_lose, game_win, disconnect
from	string	IP do peer remetente
to	string	IP do peer destinatário
payload	object/null	Dados específicos da mensagem ou null

Campos de Payload Específicos:

Campo	Tipo	Valores Possíveis
mine_coords	array	Array de objetos com coordenadas das minas: [{"x": int, "y": int}]
flags	array	Array de objetos com estado das bandeiras: [{"x": int, "y": int, "flagged": boolean}]

Obrigado 👍

~~Há momentos em que a maior sabedoria~~
~~é parecer não saber nada~~