

Desenvolvimento de um Jogo Multiplayer P2P utilizando o Protocolo TCP Batalha Naval

Peer-to-Peer: Padronização de Protocolo de Aplicação

Unisinos

Objetivo do Trabalho

A turma deve desenvolver, de forma colaborativa, uma aplicação de jogo multiplayer de Batalha Naval, utilizando o protocolo TCP em uma arquitetura peer-to-peer, sem um servidor central. O objetivo é criar um jogo onde dois jogadores possam se conectar diretamente e disputar uma partida em tempo real, coordenando as ações e atualizando o estado do jogo entre si.

Descrição do Jogo: Batalha Naval

O jogo de Batalha Naval consiste em uma competição estratégica entre dois jogadores. Cada jogador posiciona seus navios em uma grade de 10x10 e tenta adivinhar as posições dos navios do oponente. Cada jogador, em sua vez, "atira" em uma coordenada da grade do oponente, tentando acertar e destruir todos os navios adversários. O jogo termina quando um dos jogadores afunda todos os navios do oponente.

Especificações do Jogo

1. Arquitetura Peer-to-Peer (P2P)

- O jogo deve ser baseado em uma comunicação direta entre os dois jogadores, sem a necessidade de um servidor central.
- Cada jogador atuará como um "nó" P2P e será responsável tanto pelo envio quanto pelo recebimento das atualizações de estado e comandos do jogo.
- Deve ser desenvolvido um protocolo de comunicação para troca de mensagens entre os pares, com base no TCP, para garantir a confiabilidade dos dados.

2. Comunicação TCP entre os Pares

- Os jogadores devem estabelecer uma conexão direta utilizando o protocolo TCP.
- A aplicação deve usar sockets TCP para enviar e receber dados.
- As mensagens trocadas devem seguir um protocolo de comunicação previamente definido, incluindo comandos como "POSICIONAR NAVIO", "ATIRAR", "RESULTADO DO TIRO" e "ATUALIZAÇÃO DO JOGO".

3. Regras do Jogo

- Cada jogador deve ter uma grade de 10x10 onde posicionará seus navios.
- A frota de cada jogador deve incluir os seguintes navios:
 - 1 porta-aviões (5 células);
 - 1 encouraçado (4 células);
 - 2 cruzadores (3 células cada);
 - 2 destróieres (2 células cada).
- Os navios podem ser posicionados horizontal ou verticalmente e não podem se sobrepor.
- O jogo deve seguir uma lógica de turnos, em que cada jogador tenta acertar a posição dos navios do oponente em sua vez.
- Um jogador vence ao destruir toda a frota do oponente.

4. Especificação do Protocolo de Comunicação

- O protocolo de mensagens entre os jogadores deve incluir:
 - Conexão Inicial: Mensagens de sincronização para estabelecer a conexão e iniciar o jogo.
 - Posicionamento de Navios: Mensagens para indicar as coordenadas dos navios de cada jogador.
 - Jogada (Tiro): Mensagens para indicar a coordenada onde o jogador deseja "atirar" na grade do oponente.
 - Resultado do Tiro: Mensagem de resposta para indicar se o tiro foi "ACERTO" ou "ÁGUA".
 - Atualização de Status: Mensagens para atualizar o estado do jogo, como "NAVIO DESTRUÍDO", "VITÓRIA" ou "DERROTA".

5. Sincronização e Resolução de Conflitos

- Como a arquitetura é P2P, os jogadores devem ser responsáveis por garantir a sincronização correta do estado do jogo.
- Em caso de desconexão, o jogo deve ser pausado até que a conexão seja restabelecida ou o jogo encerrado.
- Devem ser implementadas regras para resolução de conflitos, caso alguma mensagem receba resposta incorreta, incluindo verificação de status.

6. Interface do Jogo

- A interface do jogo pode ser baseada em texto ou gráfica, dependendo das habilidades do aluno.

- Deve permitir que os jogadores visualizem:
 - A sua própria grade com seus navios e os tiros recebidos.
 - A grade do oponente com os tiros realizados e seus resultados (acerto/erro).
 - Os jogadores devem poder inserir coordenadas para “atirar” no oponente e receber feedback imediato do resultado.

Protocolo da Camada de Aplicação

Nesta seção será discutido como será cada etapa do jogo.

Porta Padrão: 8080

A aplicação utilizará a porta 8080 como padrão para comunicação entre os peers. Essa escolha visa garantir uniformidade e facilitar a configuração da rede entre os participantes. Os peers devem se conectar exclusivamente a essa porta para estabelecer e manter as interações do jogo.

Ordem de Início e Troca de Dados:

O primeiro peer levantado assume automaticamente o papel de iniciar o envio de navios e será o primeiro a jogar. Essa regra define que a prioridade de início será determinada pela ordem de inicialização dos peers, garantindo uma organização clara.

- Envio de navios: O peer que inicia deve enviar sua configuração de tabuleiro primeiro.

- Primeira jogada: O mesmo peer será o primeiro a realizar uma jogada, enviando o formato de tiro (detalhado abaixo).

Formato de Envio do Tabuleiro:

O tabuleiro será enviado no formato JSON, contendo uma lista de objetos, onde cada objeto representa um tipo de embarcação e suas respectivas posições no tabuleiro. Estrutura do Objeto: "tipo": Indica o tipo de embarcação. "posicoes": Lista de coordenadas (pares [linha, coluna]) que representam as posições ocupadas pelo navio no tabuleiro.

Exemplo de Envio do Tabuleiro:

```
[  
  {  
    "tipo": "porta-avioes",  
    "posicoes": [[0, 0], [0, 1], [0, 2], [0, 3], [0, 4]]  
  },  
  {  
    "tipo": "encouracado",  
    "posicoes": [[1, 0], [1, 1], [1, 2], [1, 3]]  
  },  
  {  
    "tipo": "cruzador",  
    "posicoes": [[2, 0], [2, 1], [2, 2]]  
  },  
]
```

```
{  
  "tipo": "cruzador",  
  "posicoes": [[3, 0], [3, 1], [3, 2]]  
},  
  
{  
  "tipo": "destróier",  
  "posicoes": [[4, 0], [4, 1]]  
},  
  
{  
  "tipo": "destróier",  
  "posicoes": [[5, 0], [5, 1]]  
}  
]
```

Formato dos Tiros:

Os tiros realizados no tabuleiro devem ser enviados como strings, representando coordenadas no formato "linha + coluna", onde: A linha e a coluna variam de 0 a 9. Exemplo de tiro: "23" (atira na 2ª linha e 3ª coluna). O resultado do tiro, como acerto ou erro, será tratado internamente por cada peer. Acertos: O peer deve identificar o impacto em um navio no tabuleiro. Erros: O peer deve registrar e continuar o jogo sem alterar o estado do tabuleiro do oponente.

Validação e Tratamento de Dados:

Os dados enviados durante o jogo (tabuleiros, tiros, turnos) devem ser validados internamente para evitar erros de comunicação. Cada peer é responsável por implementar verificações para:

- Formatos de mensagens: Garantir que o envio e recepção de dados sigam as especificações deste protocolo.
- Jogadas inválidas: Rejeitar jogadas fora do formato ou fora dos limites do tabuleiro.

Persistência de Dados

A persistência de dados será implementada individualmente por cada peer. É obrigatória a capacidade de salvar:

- Ambos os tabuleiros: O tabuleiro do próprio peer e do oponente.
- Estado do jogo: Incluindo o turno atual e a sequência de jogadas realizadas. Essa persistência deve permitir a recuperação do estado do jogo em caso de falha ou reinicialização, garantindo continuidade sem a necessidade de reiniciar a partida.