



RELATÓRIO T2 - Jogo Multiplayer Usando Go e RPC

INTEGRANTES:

Larissa Oliveira da Silva
Maria Eduarda Schöler
Nicolí de Oliveira Siqueira

1. OBJETIVO DO PROJETO

Este trabalho teve como objetivo adaptar um jogo originalmente de jogador único para um ambiente multiplayer, utilizando a linguagem Go e comunicação via RPC (Remote Procedure Call). A nova versão permite que múltiplos jogadores se conectem a um servidor central para interagir em um mundo compartilhado, com atualizações de estado sincronizadas entre os clientes.

2. ARQUITETURA DO SISTEMA

A arquitetura segue o modelo cliente-servidor:

- **Servidor de Jogo:** responsável por manter o estado global do jogo (posições dos jogadores, status de elementos como portais e chaves). O servidor não possui interface gráfica e responde às requisições RPC dos clientes.
- **Cliente do Jogo:** cada cliente possui uma interface gráfica que permite ao jogador se mover e interagir com o ambiente. Os clientes se comunicam periodicamente com o servidor para enviar comandos e buscar atualizações do estado do jogo.

3. COMUNICAÇÃO VIA RPC

A comunicação entre clientes e servidor é feita através de chamadas RPC definidas em estruturas do pacote *shared*. Algumas operações implementadas incluem:

- Registro de novo jogador (com ID)
- Atualização de movimento
- Solicitação do estado atual do jogo

Cada cliente possui uma goroutine dedicada à comunicação com o servidor, evitando bloqueios na interface principal.

4. REQUISITOS DE IDEMPOTÊNCIA

Para garantir a semântica de "exactly-once", estamos utilizando identificadores únicos para cada comando enviado (sequenceNumber). Isso evita que, em caso de falhas ou retransmissões, um mesmo comando seja processado duas vezes pelo servidor.

5. SITUAÇÃO ATUAL DO PROJETO

Até o momento, as seguintes funcionalidades já estão implementadas:

- Estrutura básica de servidor e cliente
- Conexão RPC funcional entre clientes e servidor
- Registro de jogadores e controle básico de estado
- Atualização de posições de jogadores

Funcionalidades que ainda precisam de melhorias:

- Integração completa com os elementos do mapa (portal, chaves, NPCs)
- Sincronização completa do mapa entre os clientes
- Tratamento completo de erros e reenvio de comandos com controle de idempotência

6. CONCLUSÃO

A implementação do jogo multiplayer com Go e RPC proporcionou uma aplicação prática dos conceitos de processos paralelos e comunicação entre sistemas distribuídos. A arquitetura desenvolvida permite que múltiplos jogadores interajam em um ambiente compartilhado de forma sincronizada, com um servidor centralizado responsável pelo gerenciamento do estado global.

O uso de goroutines e chamadas RPC demonstrou-se eficaz para manter a responsividade do cliente e garantir a integridade das ações no jogo. O trabalho consolidou conhecimentos importantes sobre concorrência, paralelismo e design de sistemas em rede, além de reforçar boas práticas de organização modular e comunicação segura entre processos.