



UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO

ENGENHARIA DE COMPUTAÇÃO

Projeto - Redes de Computadores

Rita de Kassia Lemos Pereira

Higor de Lima Gomes

Guilherme Oliveira Silva Gomes

José Anderson da Silva

MACEIÓ - 2024

SUMÁRIO

1	PROJETO - REDES DE COMPUTADORES	2
1.1	Introdução	2
1.2	Funcionalidades	2
1.3	O que poderia ter sido implementado a mais	3
1.4	Principais dificuldades na implementação do projeto	3
1.5	Executando o projeto	3
1.6	Código fonte: servidor	4
1.7	Código fonte: client	4

1 PROJETO - REDES DE COMPUTADORES

1.1 Introdução

Projeto de uma Aplicação em Redes para Jogo de Damas: Este projeto foi apresentado como requisito avaliativo parcial para a disciplina de Redes de Computadores, ministrada no Instituto de Computação da Universidade Federal de Alagoas (UFAL) pelo Professor Almir Pereira Guimarães. O projeto visa desenvolver uma versão digital do tradicional jogo de damas, implementada através de tecnologias de redes, permitindo a interação entre jogadores diferentes.

1.2 Funcionalidades

Nosso projeto propôs a criação de uma aplicação de jogo de damas em rede, possibilitando que jogadores diferentes se conectem e disputem partidas em tempo real. Com a implementação de uma arquitetura cliente-servidor utilizando Python, sockets e o protocolo TCP (essencial pois precisamos de confiabilidade para a aplicação), conseguimos oferecer uma experiência de jogo fluida e interativa.

As principais funcionalidades incluem:

- **Conexão Cliente-Servidor:** Utilizando sockets, estabelecemos uma comunicação eficiente entre o cliente e o servidor, permitindo que dois jogadores interajam em uma partida de damas em tempo real através da rede.
- **Interface Gráfica Intuitiva:** Através da biblioteca Pygame, desenvolvemos uma interface gráfica intuitiva que exibe o tabuleiro de damas, as peças, e informações relevantes da partida, como o jogador da vez.
- **Lógica de Jogo de Damas:** Implementamos as regras tradicionais do jogo de damas, incluindo movimentação de peças e capturas, garantindo uma experiência de jogo autêntica de forma digital.
- **Turnos Alternados e Sincronização de Jogadas:** Garantimos que cada jogador possa realizar suas jogadas alternadamente, com o servidor controlando a vez e sincronizando o estado do jogo entre os dois clientes.

1.3 O que poderia ter sido implementado a mais

Para aprimorar o nosso projeto de jogo de damas em rede, poderíamos considerar algumas melhorias sutis para melhorar a experiência do usuário, como: aperfeiçoar a eficiência da comunicação, diminuir a latência e otimizar o uso da banda, balanceando confiabilidade e velocidade. Um aspecto crucial para garantir a continuidade da experiência do jogador seria a implementação de mecanismos de recuperação e reativas automáticas para lidar com falhas temporárias de rede. Esses mecanismos iria permitir que os jogadores se reconectem automaticamente e retomem suas partidas sem perdas significativas de progresso, mesmo em face de interrupções de conexão. Por fim, pensando em questões de aplicações maiores, seria interessante pensar em ferramentas de monitoramento e análise de rede, que podem oferecer informações sobre a performance e o comportamento do aplicativo em ambientes de rede reais.

1.4 Principais dificuldades na implementação do projeto

Ao trazer para a prática os conceitos estudados durante o semestre, enfrentamos várias dificuldades, especialmente ao explorar os conceitos fundamentais de sockets em Python e a questão do protocolo TCP. Utilizando funções primitivas como `socket()`, nosso desafio foi além de apenas entender teoricamente essas operações: entender como implementar e sair do "abstrato".

1.5 Executando o projeto

O repositório do projeto pode ser encontrado no **GitHub**. Primeiro, crie um ambiente virtual utilizando o seguinte comando no terminal. Isso ajudará a isolar as dependências do projeto: **`python -m venv /reds`**.

Após a criação, é necessário ativar o ambiente virtual. O processo varia ligeiramente dependendo do sistema operacional utilizado.

- No windows: **`/reds/Scripts/activate`**
- No Unix ou MacOS: **`source /reds/bin/activate`**

Com o ambiente virtual ativado, proceda com a instalação das dependências necessárias para o projeto. Neste caso, o Pygame é essencial para a execução do jogo. Execute o

seguinte comando para instalar o Pygame: **pip install pygame**.

Para colocar o jogo em funcionamento, é necessário inicializar o servidor e, em seguida, abrir duas instâncias do cliente.

Após isso, abra um terminal, navegue até o diretório do projeto onde o arquivo `server.py` está localizado e execute: **python server.py**.

Para iniciar uma partida, são necessárias duas instâncias do cliente. Abra dois novos terminais, navegue até o diretório do projeto e execute o seguinte comando em cada terminal: **python client.py**.

Siga as instruções na tela para jogar e interagir com a aplicação. Importante: o servidor precisa estar rodando para que os clientes possam se conectar e iniciar o jogo.

1.6 Código fonte: servidor

O código fonte do servidor pode ser encontrado no repositório do projeto. Clique **AQUI** para acessar.

1.7 Código fonte: client

O código fonte do client pode ser encontrado no repositório do projeto. Clique **AQUI** para acessar.