Síntese

Relatório do trabalho proposto pelo professor Luís Romero

Relatório LDP

Laboratório de Programação



Grupo 1: Bernardo Azevedo – 23791 // Tiago Silva – 22283 // Gaspar Espinheira - 23761

Índice

[Manual de Utilização 2](#_Toc105970206)

[Diagrama de classes e relações 8](#_Toc105970207)

[Desenvolvimento e implementação 8](#_Toc105970208)

[Observações e correções 12](#_Toc105970209)

[Resultados e conclusões 12](#_Toc105970210)

[Referências 13](#_Toc105970211)

# Manual de Utilização

Primeiramente, o utilizador tem de correr o jogo. Dois jogadores abrem o jogo e é lhes apresentado o ecrã inicial do jogo, com as opções “Jogar”, “Regras”, “Conexão” e “Sair”.



Caso um jogador selecione a opção de “Sair”, a janela de jogo fecha. Caso selecione a opção de “Regras”, é apresentado ao jogador uma nova janela com as regras principais do jogo e caso o jogador selecione a opção “Conexão”, é apresentado uma janela de modo a explicar como estabelecer a conexão com o 2 jogador.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Caso o jogador selecione o botão “Jogar”, vai para outro ecrã onde jogar contra uma pessoa no mesmo computador ou então estabelecer a conexão com o outro jogador.

Uma imagem com quadrado

Descrição gerada automaticamente

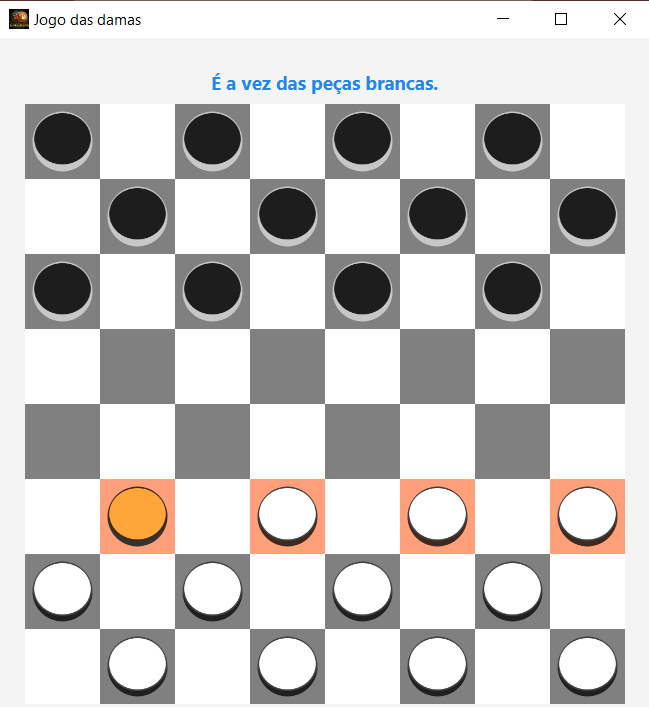
Para jogar via servidor, é necessário um dos jogadores criar um novo jogo. Para isso é necessário selecionar uma “Porta” e carregar no botão “Criar”. O outro jogador apenas terá que introduzir o seu endereço de IP e a respetiva “Porta” criada e carregar no botão “Entrar”.

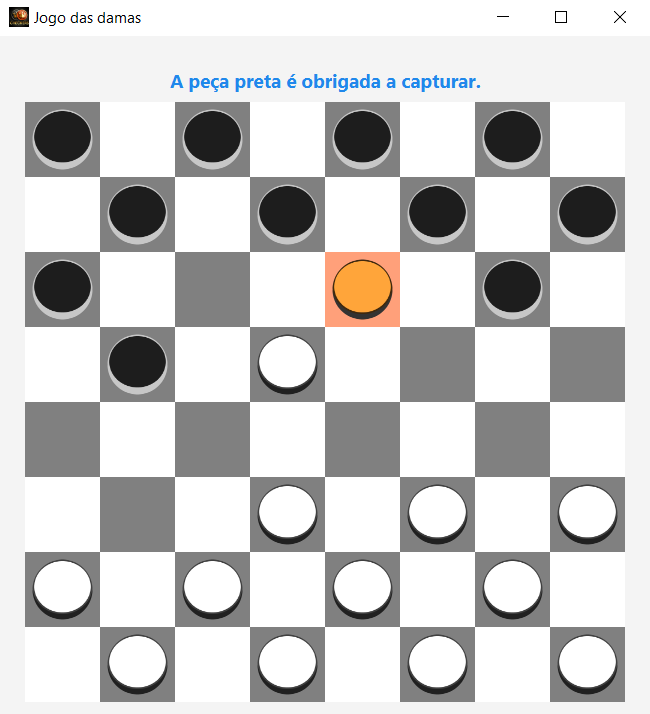


Após ser estabelecida a ligação entre os dois jogadores poderão iniciar o jogo.

No topo da tela do jogo é possível visualizar mensagens de atualização/estado do jogo. Como por exemplo, de quem é a vez de jogar, assim como quem ganhou.

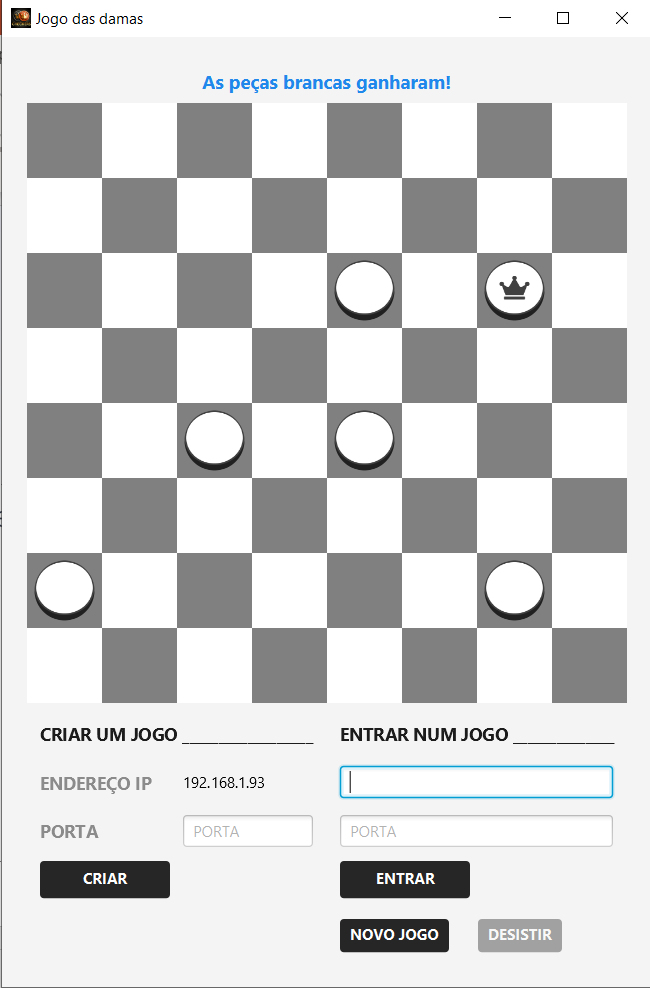
No fundo da tela encontram-se dois botões, “Novo Jogo” que apenas será possível a interação assim que o jogo termine e o botão “Desistir” que poderá ser interagido até ao fim do jogo, onde ao ser premido encerra o jogo, atualizado com uma mensagem de desistência.





Uma imagem com quadrado

Descrição gerada automaticamente



Uma imagem com quadrado

Descrição gerada automaticamente

# Diagrama de classes e relações

## 

# Desenvolvimento e implementação

**Classe *Server***

Esta classe contém o servidor do jogo e permite que os jogadores se conectem ao mesmo. O server possui um construtor em que é atribuído o tabuleiro e uma porta.

O método conectar() inicializa um socket servidor e abre a conexão a um possível socket de cliente.

O método streams() é o responsável pelo object input e object output, que acaba por ser responsável pelo envio e receção de dados dessa conexão.

O método enviarData() é o método responsável por enviar dados dessa conexão e por outro lado o método lerdata() é o responsável por receber dados dessa conexão.

O método lerData() está associado ao método verificarData() que por sua vez apresenta um switch para verificar a ação do utilizador. Esta ação poderá ser:

- “Novo\_Jogo” posteriormente inicializará um novo jogo;

- “Desistir” que permite desistir do jogo;

- “Clicar quadrado” que fará o utilizador selecionar uma peça e escolher para onde quer que a peça vá.

O método cancelarConexão() fecha ambos os sockets, o servidor e o do cliente.

**Classe *Client***

A classe Client está responsável por inicializar o socket do cliente

**Classe *NetworkInterface***

A classe NetworkInterface é um conjunto dos três métodos principais entre a conexão dos dois utilizadores (Começar um jogo, Desistir do jogo, Ação de selecionar e mover uma peça).

**Classe *DadosTabuleiro***

A classe DadosTabuleiro apresenta variáveis principais da construção do tabuleiro, como por exemplo, tamanho do tabuleiro, tamanho das peças, tamanho dos quadrados do tabuleiro.

**Classe *Interface***

A classe Interface está responsável por atribuir o tabuleiro á devida posição na interface gráfica, assim como os métodos criar um jogo/entrar num jogo.

O método criarJogo() está associado á inicialização de um socket servidor e através de um thread aguarda que o outro utilizador se conecte.

O método entrarJogo() está associado á inicialização de um socket cliente que por sua vez se conecta ao socket do servidor.

Assim que a conexão for feita é atribuído o número dos jogadores é inicializado o jogo e num dos jogadores é invertido o tabuleiro.

**Classe *Jogo***

A classe Jogo é a classe onde é realizada toda a lógica do jogo usando as outras classe já criadas.

O método começarNovoJogo() verifica se o jogo está a decorrer ou se está finalizado, habilitando/desabilitando respetivamente o botão.

O método desistirJogo() verifica se o jogo está a decorrer e faz a verificação que peça é que carregou no botão desistir.

O método clickQuadradao() verifica que peça foi selecionada e verifica as jogadas possíveis dessa mesma peça.

O método realizarMovimento() verifica o tipo de movimento a ser realizado demonstrando a mensagem no ecrã.

O método desenhaTabuleiro() como o nome indica fica responsável pelo desenho do tabuleiro usando as classes DadosTabuleiro, fazendo a atribuição da cor do tabuleiro e das imagens para as peças.

O método handle() verifica onde o utilizador clicou, caso clique no botão novo jogo ele começa um novo jogo, por outro lado caso ele clique numa peça o método regista as coordenadas do clique assim como as coordenadas de para onde quer que a peça se mova. Este método verifica também se a peça clicada é a do utilizador que está a jogar.

**Classe *Main***

A classe Main é o controlador da interface do menu inicial

**Classe *MovimentoPeça***

A classe MovimentoPeça regista as coordenadas, de que linha e coluna para uma dada linha e coluna.

**Classe *Tabuleiro***

A classe Tabuleiro define o tabuleiro do jogo.

É definido o array com as diferentes peças do jogo.

O método moverPeça() verifica de onde para onde a peça está a ser movida, atualizando o tabuleiro com o devido movimento.

O método podeCapturar() verifica se pode ou não capturar.

O método podeMover() verifica se pode ou não mover.

O método guardaMovimentosCapturas() guarda os movimentos efetuados, sejam eles movimentos ou capturas.

**Classe *Menu***

A classe Menu é o controlador para o Menu Inicial

**Classe *Instruções***

A classe Instruções é o controlador para a janela das Instruções da conexão.

**Classe *Regras***

A classe Regras é o controlador para a janela das Regras

# Observações e correções

Tendo em conta as apresentações anteriormente realizadas, consideramos que houve uma alteração drástica, pois não tínhamos o conhecimento devido para o desenvolvimento deste projeto, relativamente à estrutura de classes.

Relativamente à 2ºapresentação realizada (Protótipo de Baixo Nível – Interface e Interação) consideramos que conseguimos representar a interface da maneira planeada inicialmente.

# Resultados e conclusões

Para concluir, consideramos que este projeto foi essencial na consolidação de conhecimentos obtidos nas aulas. Através do *Scene Builder* conseguimos criar a interface do jogo  *Checkers*. Foi possível conectar o jogo a um servidor, onde os jogadores podem jogar e enviar ao adversário a jogada executada.

Durante a realização do projeto, tivemos bastantes dificuldades, mas, consideramos que os requisitos foram cumpridos na sua maioria.

# Referências