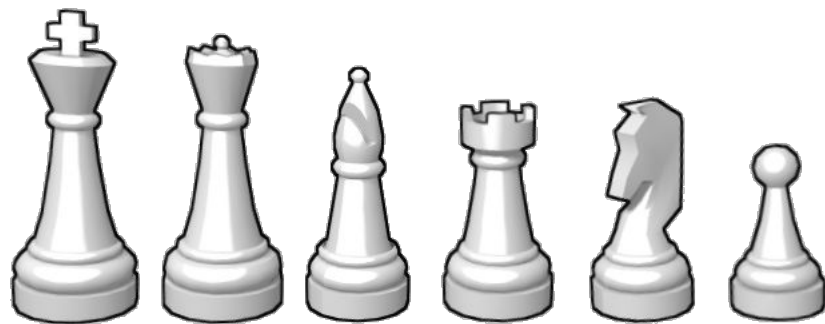


TP Final - Xadrez

Artur Vaz - Davi Beltrao - Giovanni Tagliaferri

O Jogo

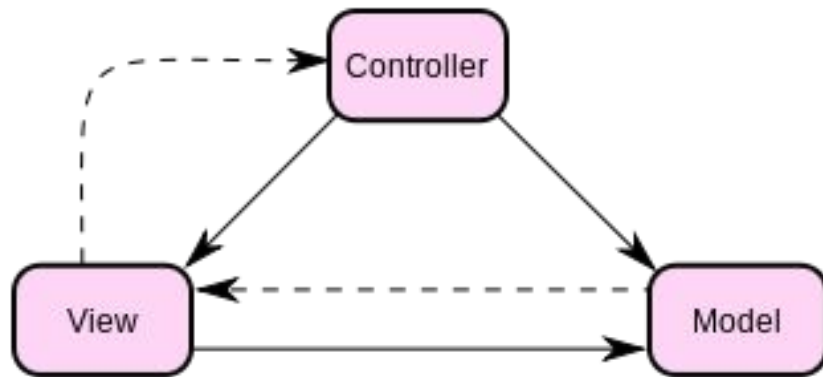
- Surgiu no século XV na Europa, baseado em um jogo vindo da Índia;
- Tabuleiro de tamanho 8x8;
- 16 peças para cada jogador;
- Das 16 peças totais possuem 6 tipos;
- Cada tipo de peça tem um tipo de movimento;
- Aproximadamente 10^{123} jogadas possíveis;
- Ganha quem conseguir derrotar o Rei inimigo.



Implementação

Foi usado o padrão Model-View-Controller(MVC) para separar a implementação da visualização da interação.

- Model -> Estrutura da aplicação.
- View -> Visualização da aplicação.
- Controller -> Controle dos inputs.



Model

- Implementação da lógica e estrutura de dados do jogo;
- Tabuleiro representado como uma matriz de posições;
- Singleton;
- Possui uma referência ao View;
- Responsável por:
 - Guardar o estado atual do tabuleiro e das peças;
 - Verificar se a jogada é válida;
 - Verificar se a peça está em cheque e se o jogo terminou;
 - Fazer a conexão no modo multiplayer;
 - Atualizar o estado do tabuleiro entre os jogadores conectados;
 - Decidir o movimento no modo Player vs AI;
 - Enviar mensagens para o View para atualização da Interface.

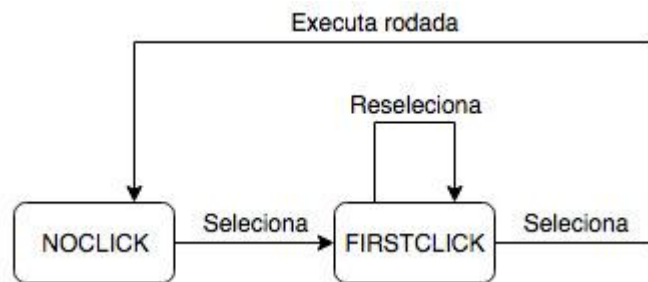
View

- Interface do usuário;
- Implementado utilizando principalmente a biblioteca Swing do Java;
- Possui uma thread que manda repintar o painel de 10 em 10 milissegundos;
- Possui uma referência ao controle para que possamos conectar os elementos da interface aos eventos que ocorrem nele;

Controller

O jogo foi implementado com interação exclusiva do mouse, dessa forma a classe Controller implementa a interface `MouseListener`, que contém métodos que ajudam na captura da posição clicada no tabuleiro.

A ideia por trás do controller é enviar a intenção do usuário para o model, foi usado estados que garantem a captura da posição inicial e final da peça.



Piece

- Possui informações sobre:
 - Jogador que a pertence;
 - Posição;
 - Jogadas possíveis.
- Cada tipo de peça estende a classe Piece;
- Há override na função de gerar as jogadas possíveis para cada tipo, levando em conta suas regras de movimentação;

Swing

- Biblioteca de interface do Java;
- A parte de visualização depende diretamente do Swing;
- A parte do controle depende parcialmente do Swing e do AWT;
- O que utilizamos:
 - JPanel;
 - Grid Layout;
 - Card Layout;
 - JLabels e Imagens;
 - JOptionPane.

Sockets

- Existe um ServerSocket e um Socket em cada instância do Model;
- Server só é utilizado no jogador host;
- O processamento da jogada é realizado de forma local, é transferido para o outro socket a jogada assim que o Model julga a sua validade;
- É enviado inteiros em pares, o primeiro par é referente a peça que vai ser movida, o segundo par é a posição para o qual foi movido;
- É enviado uma dupla de pares especiais no final do turno para avisar ao inimigo que o turno passou;
- A comunicação é feita em uma thread a parte, para poder atualizar em tempo real o tabuleiro;

Diagrama de classes

