

Projeto Sistema de Jogo de Xadrez

Objetivo geral:

- Aplicar os conhecimentos aprendidos até o momento em Java para a construção de um projeto

First class: Position

Checklist:

- Class Position [public]
- OOP Topics:
 - o Encapsulation
 - o Constructors
 - o ToString (Object / overriding)

Starting to implement Board and Piece

Checklist:

- Classes Piece, Board [public]
- OOP Topics:
 - o Associations
 - o Encapsulation / Access Modifiers
- Data Structures Topics:
 - o Matrix

Chess layer and printing the board

```
8 - - - - -
7 - - - - -
6 - - - - -
5 - - - - -
4 - - - - -
3 - - - - -
2 - - - - -
1 - - - - -
  a b c d e f g h
```

Checklist:

- Methods: Board.Piece(row, column) and Board.Piece(position)
- Enum Chess.Color
- Class Chess.ChessPiece [public]
- Class Chess.ChessMatch [public]
- Class ChessConsole.UI
- OOP Topics: o Enumerations o Encapsulation / Access Modifiers o Inheritance o Downcasting o Static members o Layers pattern
- Data Structures Topics:
 - o Matrix

Placing pieces on the board**Checklist:**

- Method: Board.PlacePiece(piece, position)
- Classes: Rook, King [public]
- Method: ChessMatch.InitialSetup
- OOP Topics:
 - o Inheritance
 - o Overriding
 - o Polymorphism (ToString)

BoardException and defensive programming**Checklist:**

- Class BoardException [public]
- Methods: Board.PositionExists, Board.ThereIsAPiece
- Implement defensive programming in Board methods
- OOP Topics:
 - o Exceptions
 - o Constructors (a string must be informed to the exception)

ChessException and ChessPosition

Checklist:

- Class ChessException [public]
- Class ChessPosition [public]
- Refactor ChessMatch.InitialSetup
- OOP Topics:
 - o Exceptions
 - o Encapsulation
 - o Constructors (a string must be informed to the exception)
 - o Overriding
 - o Static members
 - o Layers pattern

Little improvement in board printing

Color in terminal:

- Windows: Git Bash
- Mac: Google "osx terminal color" Checklist:
- Place more pieces on the board
- Distinguish piece colors in UI.PrintPiece method

Moving pieces

Checklist:

- Method Board.RemovePiece
- Method UI.ReadChessPosition
- Method ChessMatch.PerformChessMove
 - o Method ChessMatch.MakeMove
 - o Method ChessMatch.ValidadeSourcePosition
- Write basic logic on Program.cs
- OOP Topics:
 - o Exceptions
 - o Encapsulation

Handling exceptions and clearing screen Clear screen using Java

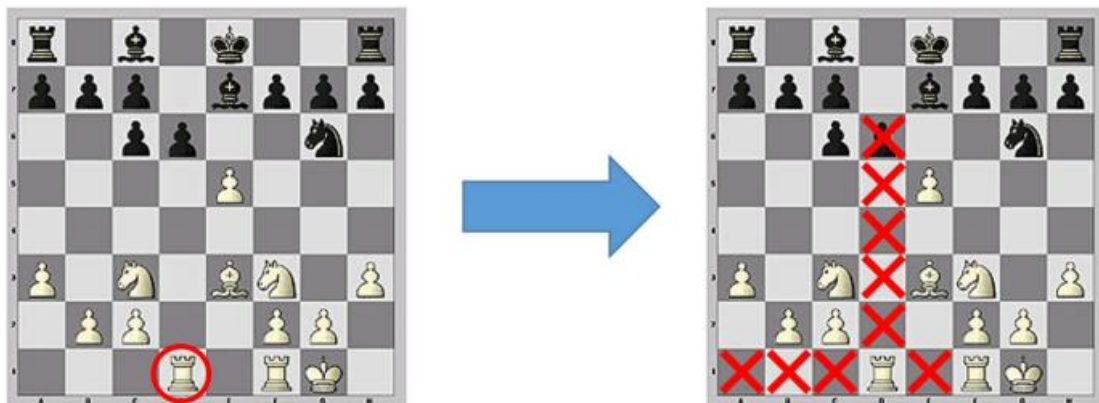
: // <https://stackoverflow.com/questions/2979383/java-clear-the-console>

```
public static void clearScreen() {  
    System.out.print("\033[H\033[2J");  
    System.out.flush();  
}
```

Checklist:

- ChessException
- InputMismatchException

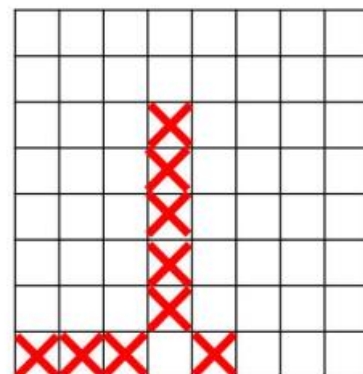
Possible moves of a piece



Input: a piece



Output: a boolean matrix of possible movements



Checklist:

- Methods in Piece:
 - o PossibleMoves [abstract]
 - o PossibleMove
 - o IsThereAnyPossibleMove
- Basic PossibleMove implementation for Rook and King
- Update ChessMatch.ValidadeSourcePosition
- OOP Topics: o Abstract method / class
 - o Exceptions

Implementing possible moves of Rook**Checklist:**

- Method ChessPiece.IsThereOpponentPiece(position) [protected]
- Implement Rook.PossibleMoves
- Method ChessMatch.ValidateTargetPosition
- OOP Topics:
 - o Polymorphism
 - o Encapsulation / access modifiers [protected]
 - o Exceptions

Printing possible moves**Checklist:**

- Method ChessMatch.PossibleMoves
- Method UI.PrintBoard [overload]
- Refactor main program logic
- OOP Topics:
 - o Overloading

Implementing possible moves of King

Checklist:

- Method King.CanMove(position) [private]
- Implement King.PossibleMoves
- OOP Topics:
 - o Encapsulation
 - o Polymorphism

Switching player each turn

Checklist:

- Class ChessMatch:
 - o Properties Turn, CurrentPlayer [private set]
 - o Method NextTurn [private]
 - o Update PerformChessMove
 - o Update ValidadeSourcePosition
- Method UI.PrintMatch
- OOP Topics: o Encapsulation
 - o Exceptions

Handling captured pieces

Checklist:

- Method UI.PrintCapturedPieces
- Update UI.PrintMatch
- Update Program logic
- Lists in ChessMatch: _piecesOnTheBoard, _capturedPieces
 - o Update constructor
 - o Update PlaceNewPiece
 - o Update MakeMove
- OOP Topics:
 - o Encapsulation
 - o Constructors

- Data Structures Topics:

- o List

Check logic

Rules:

- Check means your king is under threat by at least one opponent piece
- You can't put yourself in check

Checklist:

- Property ChessPiece.ChessPosition [get]
- Class ChessMatch:
 - o Method UndoMove
 - o Property Check [private set]
 - o Method Opponent [private]
 - o Method King(color) [private]
 - o Method TestCheck
 - o Update PerformChessMove
- Update UI.PrintMatch

Checkmate logic

Checklist:

- Class ChessMatch:
 - o Property Checkmate [private set]
 - o Method TestCheckmate [private]
 - o Update PerformChessMove
- Update UI.PrintMatch
- Update Program logic

Piece move count

Checklist:

- Class ChessPiece:
 - o Property MoveCount [private set]
 - o Method IncreaseMoveCount [internal]
 - o Method DecreaseMoveCount [internal]
- Class ChessMatch:
 - o Update MakeMove
 - o Update UndoMove
- OOP Topics:
 - o Encapsulation

Pawn

Checklist:

- Class Pawn
- Update ChessMatch.InitialSetup
- OOP Topics:
 - o Encapsulation
 - o Inheritance
 - o Polymorphism

Bishop

Checklist:

- Class Bishop
- Update ChessMatch.InitialSetup
- OOP Topics:
 - o Encapsulation
 - o Inheritance
 - o Polymorphism

Knight

Checklist:

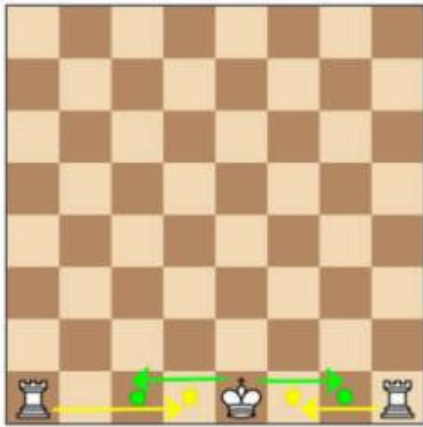
- Class Knight
- Update ChessMatch.InitialSetup
- OOP Topics:
 - o Encapsulation
 - o Inheritance
 - o Polymorphism

Queen

Checklist:

- Class Queen
- Update ChessMatch.InitialSetup
- OOP Topics:
 - o Encapsulation
 - o Inheritance
 - o Polymorphism

Special move – Castling



Checklist:

- Update King
- Update ChessMatch.MakeMove
- Update ChessMatch.UndoMove

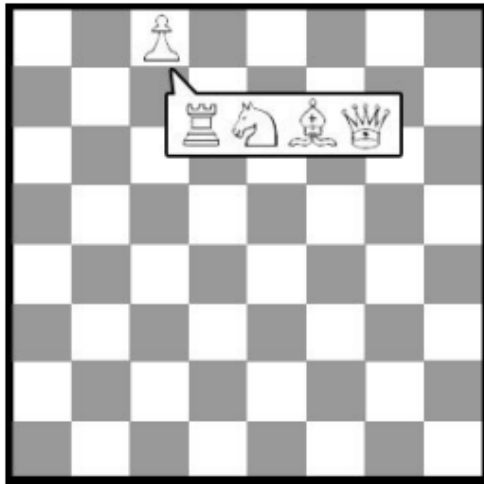
Special move - En Passant



Checklist:

- Register a pawn which can be captured by en passant on next turn
 - o Property ChessMatch.EnPassantVulnerable
 - o Update ChessMatch.PerformChessMove
- Update Pawn.PossibleMoves
- Update ChessMatch.MakeMove
- Update ChessMatch.UndoMove
- Update ChessMatch.InitialSetup

Special move – Promotion



Checklist:

- Property ChessMatch.Promoted
- Update ChessMatch.PerformChessMove
- Method ChessMatch.ReplacePromotedPiece
- Update Program log