

# React + Java Servlets Chess Game

## Game for M14 DUAL IES JOAN D'AUSTRIA

### Todolist

#### Backend

- Handle player join match (API Servlet)
- Handle piece transformation choice (API Servlet)
- Validate "Check" moves after each piece movement (Board logic using Piece's)
- Make sure backend only let move a piece when it's player turn (API Servlet move piece)
- Optional? Implement the ability to perform Castling.
- Save eaten Piece's in a List inside Match
- Save last move in A7 -> F6 notation (ChessCoordinates class) inside Match state

#### Frontend

- Render "select a piece when pawn reach the end"
- Restrict movement of player based on the actual turn
- Ask for Match state using 2sec Intervals or WebSocket when it is an Enemy Turn, this will update the Match render.
- Identify yourself as a player to be able to join Matches
- Render eaten pieces?
- Show last movement somewhere in the interface

### Done log

```
5103262 added production variable to webpack cfg
4ee9f79 added css styles
fd6d6b5 added components based on state, playerinfo and flip turn on backend
571294c refactor frontend and finished from Board to Match adaptation
acd8ea2 added Match entity class
f364036 added TokenGenerator util class
da9efe8 added Player entity class
010e111 changed aws ec2 deploy script
240bf7b css drawing row number and col letters
f0e6fac Added backend support to reply if the piece can trigger the transform
mechanics
59152a8 added feature support to Piece, detecting if it can transform based on his
position
1d929fd movement cancel by space key or tapping itself
4313bc4 client don't ask the server when cell is empty or when servers responds with 0
possible cells to move
b7bf748 refactored into useApi hook
```

187a768 error handling, created ErrorPanel  
9efb858 send post api for moving pieces  
c692de8 backend move to api implemented  
ed5906d increased the probability of null piece (no piece) when generating board randomly  
81ae06e added deploy script aws  
bbef917 Implemented creat board with random pieces for testing purposes.  
87a8cce Implemented moves for remaining Rook, Bishop, Queen pieces.  
ebc982b deploy war to aws instance  
cc601a2 Rook left movement implemented  
e1649f6 Implemented KING piece possibleMoves  
fe9f567 setup kingTestingField  
1ce6238 Implemented knight possibleMovements list  
ed1ad19 Implemented method to setup a knight testing field  
4681eb9 Pawn possible moves finished  
e01e43d wip pawn moves logic  
0c33ca0 Added support to handle api get available-pieces  
8f84dd5 created apiHandler wrapper for all json requests  
f4ffc9c feat: highlight available cell positions based on backend info  
b7fa88d moved ajax board-state to Board component from App  
e4acced added support available movements api (hardcoded list of slots)  
fed269a add support highlight cell  
4c7065a putting all remaining pieces in the board (backend)  
feb7a57 fix typo  
815f5ea refactor: creating image pieces with polymorfism in pieces.js  
ce9491f refactor and component extraction  
b1b6640 extracted Cell component for board rendering  
66c2460 rendering board  
9bf6e9d installed glamor  
da2c1a6 moved Piece.java  
fc6d426 Implemented special board pieces  
fe954a3 fix syntax if  
4ecfd0d refactor: renamed variable  
de2eeea codename is now a property so Gson can send it  
358c145 refactor extracting methods from creating board  
eaf467d Changed the way we create a board (UNFINISHED)  
97ef84a Added support to add Color to Pieces  
b81901c Added setter for boardSlot  
8fd11ba refactoring of Piece  
c767136 Fixed unimplemented constructor. Optimized properties.  
7d2b574 Added getters to BoardSlot  
2c2fac7 Created class ChessCoordinates  
a2927d1 Created class BoardSlot  
635c570 Prepared Frontend react with bundlers and backend servlets  
653eb84 preparing frontend and backend  
abfb0d5 initial setup repo