Chess Board Controller Requirements Spec2

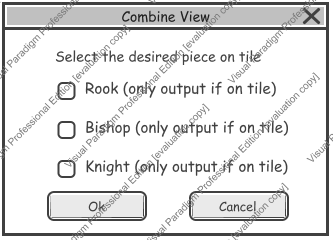
Image4.png

Dolly

# Image0.png Chess Board Controller

## Scenarios

| Scenario |
| --- |
| 1. if Image1.pngChess Board View - chessBoardViewInitialised() == False |
| 1.1. Image1.pngChess Game Model-getChessGameModel(&chessGame) |
| 1.2. Image1.pngChess Board View-displayChessBoardView(chessGame) |
| 1.3. exit Image1.pngChess Board Controller, go to Image1.pngChess Game 3 |
| 1. end if |
| 2. Image1.pngTimer-startTime() |
| 3. Image1.pngChess Board View - (listener) chessBoardView.boardClicked(clickedTile) |
| 4. Image1.pngTimer - (listener) timeOut(); |
| Extension: |
| 1.a. Display chess board for the first time |
| 3.a. handle chess board clicks |
| 1. if Image1.pngChess Game Model - getSelected(clickedTile) == true // the tile is already selected, deselect tile |
| 1.1. Image1.pngChess Game Model - deselectTile(clickedTile) |
| 2. else if Image1.pngChess Game Model - getValidMove(clickedTile) == true // if tile is a valid move for a selected piece |
| 2.1. Image1.pngChess Game Model - makeMove(): move // move - tile piece is moving to, tile piece is moving from, pieces moving, score |
| 2.2. Image1.pngChess Board View - updateChessBoardView(move) |
| 2.3. if Image1.pngChess Game Model - getActivePlayer() == player2 |
| 2.3.1. Image1.pngChess Game Model - decrementTurn() |
| 2.3.2. Image1.pngChess Board View - updateTurnView() |
| 2.3. end if |
| 2.4. Image1.pngChess Game Model - toggleActivePlayer() |
| 3. else if Image1.pngChess Game Model - getTileHasPlayerPiece(clickedTile, activePlayer) == true // tile has a piece of the active player, select and draw moves |
| 3.1. Image1.pngChess Game Model - setSelected(clickedTile): move<arraylist> |
| 3.2. for(i = 0; i < move.size(); i++) |
| 3.3.    Image1.pngChess Board View - updateChessBoard(move) |
| 3.4. end For |
| 3. end if |
| 4. go to 2 |
| 4.a. handle time outs |
| 1. Image1.pngTime Out View - displayTimeOutView() // game will simply wait for user acknowledge by clicking ok |
| 2. if Image1.pngChess Game Model - testSelectedTile() == true // a tile is selected, clear selection and any moves |
| 2.1. Image1.pngChess Game Model - getSelectedTile():tile |
| 2.2. Image1.pngChess Board View - clearTile(tile) |
| 2.3. Image1.pngChess Game Model - clearSelectedTile() |
| 2.4. Image1.pngChess Game Model - getMoves():moves<arrayList> |
| 2.5. for(i = 0 ; i < moves.size, i++) |
| 2.6. Image1.pngChess Board View - clearTile(moves.tile) |
| 2.7. end For |
| 2.8. Image1.pngChess Game Model - clearMoves():moves<arrayList> |
| 2. end if |
| 3. if Image1.pngChess Game Model - getActivePlayer() == player2 |
| 3.1. if Image1.pngChess Game Model - getTurn() == 0 |
| 3.1.1. Exit Image1.pngChess Board Controller go to Image1.pngChess Game 6 |
| 3.1. end if |
| 3.2. Image1.pngChess Game Model - decrementTurn() |
| 3.3. Image1.pngChess Board View - updateTurnView() |
| 3. end if |
| 4. Image1.pngChess Game Model - toggleActivePlayer() |
| 5. go to 2 |



3. else if Chess Game Model - getTileHasPlayerPiece(clickedTile, activePlayer) == true // tile has a piece of the active player, select and draw moves



1. Time Out View - displayTimeOutView() // game will simply wait for user acknowledge by clicking ok