|  |  |  |
| --- | --- | --- |
| **Model** | **Controller** | **View** |
| **Entities**  Chessboard   * Tile * Knight * Rook * Bishop * Barrier   Score  turn  player1  player2  activePlayer  moves  **Actions**  initialise() – Set Entities to defaults  welcome()  startGame()  selectPiece()  deselectPiece()  selectMove()  updateChessBoard() | M – welcome(),  V: displayWelcome()  **Display Welcome Action**  If V: displayWelcome->startButton.clicked(), M: startGame(playerNames, turns)  **Display Game Action**  M: startGame(playerNames, turns), V: displayGame( chessboard, score, turn, player names, activePlayer)  **Board Clicked Action**  V: boardClicked().>tile(x,y), M: selectPiece(x,y)   1. Board is clicked, is a piece already selected? 2. True, does this piece reside in this tile? (b) 3. True, Is there more than one piece? (c) 4. True, combine/split prompt. 5. False, does the tile hold any active user piece? (d) 6. True, go to b. True. 7. False, ignore click 8. False, deselect piece 9. False, is this tile a valid move? (e) 10. True, move selected piece 11. False, ignore   **Update Board Action**  M: pieceSelected(), V: updateBoard(activePlayer,selectedPiece,Moves)  M: moveSelected(), V: updateBoard(activePlayer,selectedPiece,Moves)  **Update Score Action**  M: pieceTaken(), V: updateScore(player, score)  **Update Turn Action**  M: turnOver(), V:updateTurn(turn)  **Time Out Action**  M: timeout(), V: displayTimeOut(activePlayer)  **End Game Action**  M: gameOver(), V: endGame(winnerName, score) | displayWelcome(input names, number of turns, click start)  displayGame(chessboard, score, turn, player names)  boardClicked(x,y)  displayCombine()  updateBoard(activePlayer,selectedPiece,Moves)  updateScore(player, score)  updateTurn(turn)  displayTimeOut(activePlayer)  endGame(winnerName, score) |