Checkers Game Data Model

CheckerPiece Class

- this class represents each individual piece in the game
- Variables
  - int xCoordinate
    - int for x board coordinate
  - int yCoordinate
    - int for y board coordinate
  - Color color
    - either black or red
  - Player player
    - reference to Player object this piece belongs to
  - boolean isKing
    - true if piece has been kinged, false otherwise

Checkerboard Class

- this class represents the board itself and includes methods to move pieces and get information about the game
- Variables
  - CheckerPiece[][] gameBoardCoordinateGrid
    - a 2D array of type CheckerPiece that holds references to each piece currently in the game
  - int numberRedPieces
    - int of current number of red pieces in play
  - int numberBlackPieces
    - int of current number of black pieces in play
- Methods
  - movePiece(CheckerPiece piece, int newX, int newY)
    - this method takes a CheckerPiece and a new coordinate pair, and attempts to move that piece to a new location. If it is a valid location, then it removes the current reference on the coordinate grid, and sets it in the new location in the 2D array. This method also checks to see if any pieces were jumped in the process of moving, and if so, removes their reference from the grid. If the coordinate is valid and is at the end of the board on the opponent side, the piece is kinged. Kinged pieces are allowed extra movement options that normal pieces are not, and this function will differentiate between pieces.
  - playerDidWin()
    - checks to see if one player has won based on number of red vs black pieces. If one has reached zero, then the winner is the other player. Meant to be called after a piece is moved.
  - getNumberRedPieces()

        - returns number of red pieces
    - getNumberBlackPieces()
        - returns number of red pieces

Player Class

- Variables
    - String username
        - the players name
    - enum PieceColor
        - an enumerated value that has two possible values, RED, or BLACK. This allows us to keep track of the players color
    - boolean isPlayerControlled
        - used to tell the controller if this player should be managed by a user or by the computer

GameController Class

- this class creates the board and adds the pieces, and handles the game state
- Variables
    - Checkerboard checkerboard
        - a checkerboard object that represents the current game's checkerboard, created when a new game is started
    - Player player1
    - Player player2
    - Player currentPlayer
        - player whose turn it is currently, when set, game should only allow this player's pieces to be moved
- Methods
    - generateGame()
        - this method will create a new Checkerboard, and populate it with 24 pieces, 12 for each player. The pieces will be set in their initial starting coordinates according to the rules of checkers. Player 1 will be assigned to the red pieces, and player 2 will be assigned the black pieces. Player 1 will be set as the current player initially.
    - We will need several more methods in this controller class to build the actual UI, and attach listeners to specific UI items. These functions will handle what needs to be called in the model, such as changing a piece's location.