Matt Tschannen

Checkers Game Data Model Concept

Our data model needs to be able to adapt to various board sizes, so the squares on the board will be initialized from 0 to the total area minus one. Movement will be calculated accordingly for each dimension by adding or subtracting two numbers in available directions. On a typical 8x8 board each square will have a number from 0 to 63. On the standard boars, pieces can move to a space that is not occupied by adding or subtracting 7 or 9, depending on if they are moving left or right, and backwards or forwards. Moves can be taken again if there is an opposition piece available to take. Like with assignment one, the board will be from another class that is an extension of the rectangle class. This class will have properties to correspond to the needed gameplay states. This class is where information like if a piece is on a space, what color the piece is and what kind of piece(kinged or normal) it is. When a square with a piece is clicked on, the pieces should be identified by becoming somehow highlighted or recolored. Moves will take place when a highlighted rectangle is clicked on.

Methods:

- initializeBoard()
  - Builds the board and initializes all squares to the beginning state.
- initializePieces()
  - Will be executed in initializeBoard(). This method will place pieces in the starting positions.
- onPieceClicked()
  - If a piece is clicked on, this method will find the available moves and highlight them accordingly. This method will return and highlight available moves when a piece is clicked on for both regular pieces and king pieces. Calls findAvailableMoves() on the square clicked.
- findAvailableMoves()
  - Will be executed in onPieceClicked(). Takes in the number for the corresponding square with a piece. Determines if there is a piece in the adjacent squares, if there is, check the next row to see if a capture is available. If a capture is available, check for another capture. If any of these steps fail, no move available.
- onMoveSelected()
  - Once a move is selected, the move will be executed and the board will be updated.
- checkForEndGame()
  - Checks the board to determine if one of the player's pieces are completely gone or there are no available moves left.