Noah Panicola
2-12-18
Checker Game Data Model
Object Oriented Design

Description

The objects below will be used to maintain and manipulate the data of the Checkerboard application. There will be a Checkerboard object that contains an ArrayList of Cell objects. The cell will contain a CheckerPiece, or null if there is nothing there. The CheckerBoard will determine if a piece (selected by the active user) has any available moves using the array indices. If it does have any moves it will highlight the cell the user can move to. A double jump will need to be done in two moves. The user must click on the first cell to jump, then the second. We will loop through each piece after every move to see if the game needs to be ended. If there are no pieces left or no available moves, the Player with the most CheckerPieces left will get the win. We will give the option to save the game and user information in text files.

Objects

1. Player
    - Properties
        - name → String
        - username → String
        - wins → Int
        - losses → Int
        - color → Color
        - hasNextTurn → Boolean
    - Methods
        - register() → add player to text file with 0 wins and 0 losses
        - setName(String name) → sets the name of the user
        - addWin() → adds a win to the user by username in a text file
        - addLoss() → search text file for username and add a loss
        - setColor() → sets the color
2. CheckerBoard
    - Properties
        - numRows → Int
        - numCols → Int
        - activePiece → CheckerPiece
        - cells → ArrayList <Cell>

- Methods
    - updateBoard() → redraw the board after changes have been made
    - setActivePiece() → set the active checker piece
    - getLegalMoves() → use the active piece to find the legal moves
    - setupBoard() → puts the pieces in the right spot to start a new game
    - saveBoard() → saves the board to a text file

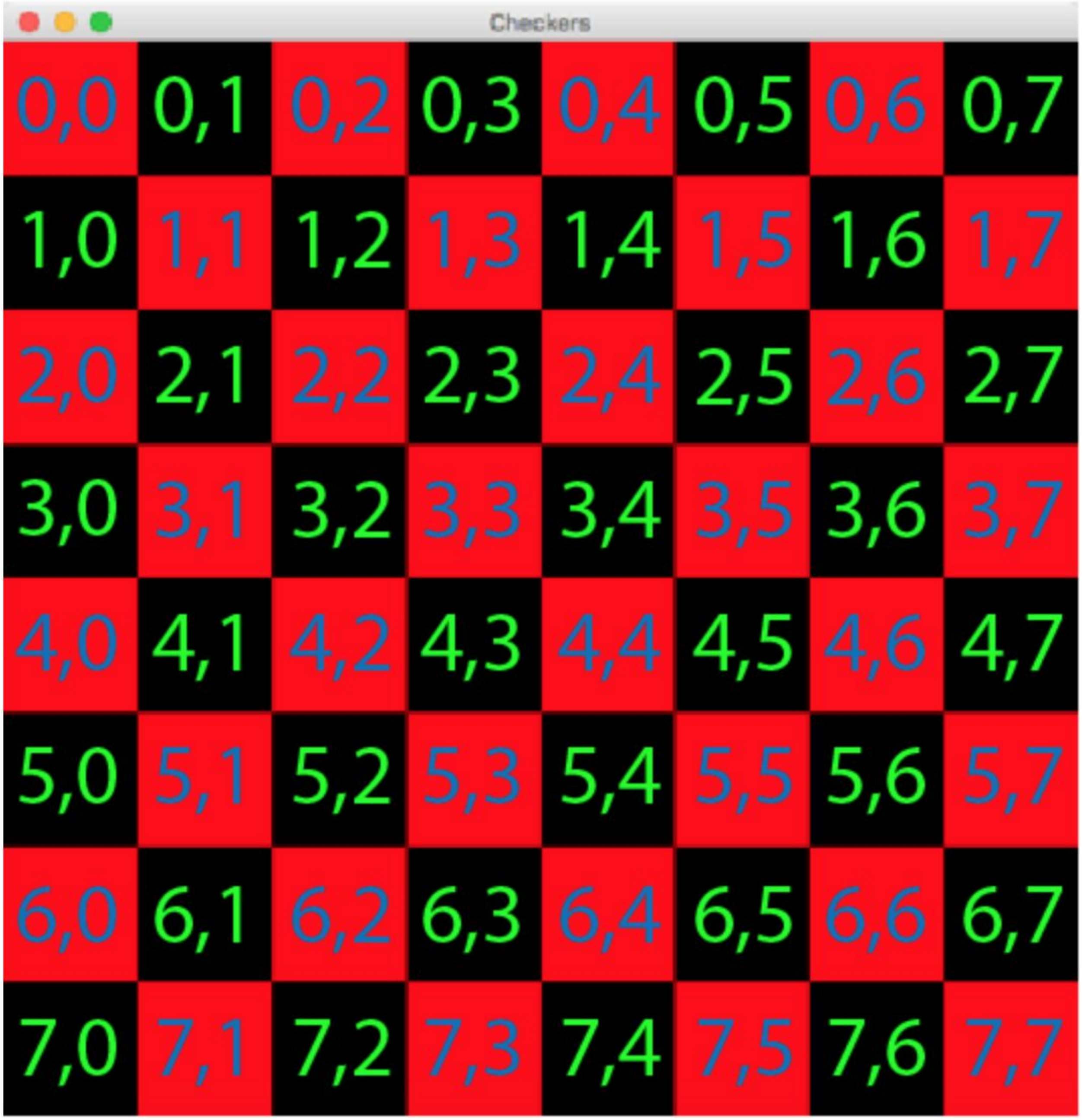
3. CheckerPiece
    - Properties
        - xIndex → Int
        - yIndex → Int
        - isSelected → Boolean
        - Color → Color
        - isKing → Boolean
        - isVisible → Boolean

    - Methods
        - getLegalMove(Cell cell) → checks one spot to see if the piece can move there
        - remove() → removes the jumped piece from the board

4. Cell
    - Properties
        - xIndex → Int
        - yIndex → Int
        - checkerPiece → CheckerPiece
        - Color → Color

User Interface

- Uses ArrayLists of Cells with X and Y indexes to keep track of data



0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7