Checkers Game Data Model Concept

The following entities are to be used for the model:

- **Entities**:
    - Game
        - board*: CheckerBoard*
        - player*: Player*
        - pieces*: Array<CheckerPiece>*
        - turn: bool
        - lastMove*: Move*
        - status: Enum
            - Black wins
            - White wins
            - Draw

        - initGame() { // initializes the game }
        - saveGame() { // saves the game }
        - loadGame(game) { // loads from a previously saved game}
        - whoseTurn { // if 0, return Black. If 1, return White}

    - Location
        - location {x, y}:
        - empty: bool
        - piece*: CheckerPiece*

        - isEmpty() {

          // Returns state of this location
          return location.Empty == true
          }
        - getPiece() { // returns the piece at this spot, if any }

    - Player
        - name: String
        - type: Enum {Computer / Human}
        - color: Enum {Black / White}

    - *Array<Checker Pieces>*
    - move*: Move*

    - getName() { // returns player name }

    - getType() { // returns type: Computer or Human }
- CheckerPiece
    - Location
    - Black
    - White
    - king: bool

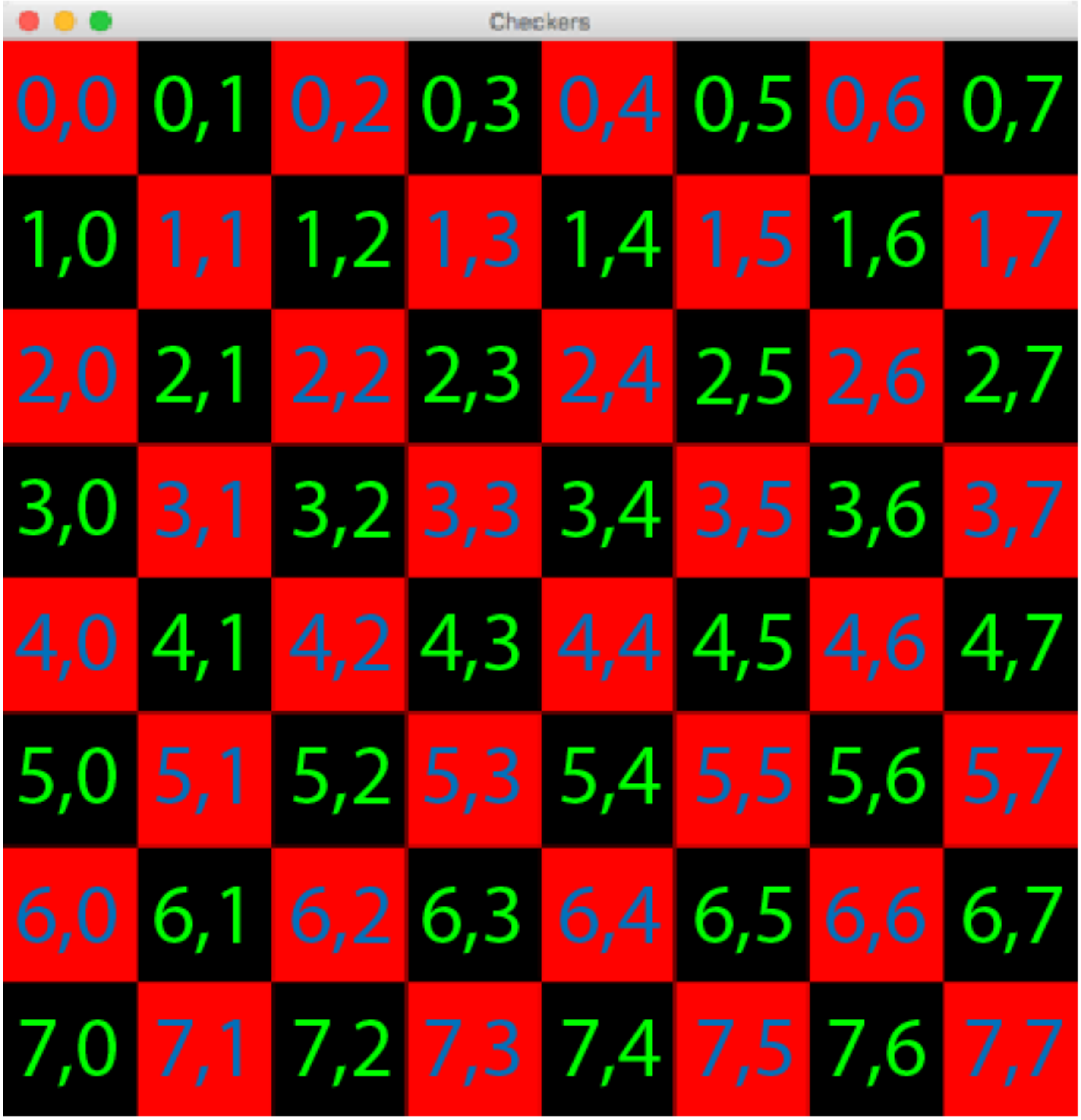
    - getLocation { // get piece location }
    - isKing { // return king boolean }
    - getType { // black/bKing, white/wKing }
- Move
    - *byPlayer: Player*
    - sourceLocation: *Location*
    - destinationLocation: *Location*

    - isLegal(CheckerPiece piece) {
      ```
              if (destinationLocation.isEmpty() {
                      return true
              }
      }
      ```
    - getPlayer() { // returns player of this move }
    - getSource() { // returns source location }
    - getDestination() { // returns destinatination location }

// Getters are to be implemented for all in general, including the ones depicted here.

The model relies on coordinates for the cells/squares, as shown below:

Dark Player



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

Light Player