**Checkers Game Data Model Concept**

The following entities are to be used for the model:

* **Entities**:
  + Game
    - board*: CheckerBoard*
    - playe*r: 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:

