**CLASS:** Board

**IMPLEMENTATION**

*The Board object is a model of the board, keeping track of all the pieces, the nodes on the board and which player occupies them, and the existence of three-in-a-rows. It contains methods to add Pieces to the board, retrieve the piece at a specified position, and to check the validity of the current board’s configuration.*

**INHERITS**

None

**USES**

java.util.ArrayList, java.util.Random, Piece, BoardError

**VARIABLES**

RED = true : boolean

*Boolean constant to represent player 1*

BLUE = false : boolean

*Boolean constant to represent player 2*

MAXPIECES : integer

*Maximum number of pieces each side can have on the board at game start*

turn : boolean

*Indicates which player's turn it currently is*

pieces : ArrayList<Piece>

*Collection of pieces currently on the board*

position : Boolean[][][]

*A 3D array corresponding to the nodes on the physical board. The value of position[x][y][z] is the color of the Piece at that position. Empty nodes point to null*

threeInARow : Boolean[][]

*2D array corresponding to the possible locations for a 3-in-a-row. Value corresponds to which color holds the 3 in a row. First index indicates which side of the square has 3-in-a-row, starting with 0 at the top and numbered clockwise. Second index indicates inner or outer square.*

maxRed : int

*The number of pieces Red can currently have on the board. Decreases when Blue gets a Three-in-a-Row*

maxBlue : int

*The number of pieces Blue can currently have on the board. Decreases when Red gets a Three-in-a-Row*

**ACCESS** **PROGRAMS**

Board ( ) : Board

*Constructs an instance of an empty board. Randomly decides which player goes first.*

addPiece (color : boolean; coordinate : int[] ) : void

*Adds a new instance of Piece with color and coordinate to pieces. Updates the color of that position on the board*

addPiece (color : boolean; x, y, z : int) : void

*Adds a new instance of Piece with color and coordinate (x,y,z) to pieces. Updates the corresponding position with color*

getPiece (coordinate : int[] ) : Piece

*Returns the piece at the input coordinate. Returns null if no piece is there.*

validSetup ( ) : BoardError

*Checks if the state of the board is a valid configuration. Returns an instance of BoardError which contains an integer indicating. See BoardError API for more details.*

updateThreeInARow ( ) : void

*Checks all possible occurrences of Three-in-a-Rows on the board and updates threeInARow accordingly.*

getNumPieces ( ) : int

*Returns the number of pieces currently placed on the board*

getTurn ( ) : boolean

*Returns boolean value indicative of whose turn it is*

switchTurn ( ) : void

*Switches state to the next player's turn*

resetBoard ( ) : void

*Resets the state of the Board to the initial state*

**RESTRICTIONS**

*This module represents the entire board of the game. As such, only one instance of this object should be constructed.*