Blokus: Iteration 1

Objectives:

* Create all unique 21 subclasses of BlokusPiece
* Get rotate and move methods of BlokusPiece to work
* Check boundaries of board
* Get placePiece method to work
  + Should not place piece on other placed pieces
* (Maybe not until next iteration) Try and get hasAdjacent and hasDiagonal to work

|  |
| --- |
| BlokusPiece |
| #pieces: BlokusSquare[]  #relativeX: int[]  #relativeY: int[]  #board: BlokusBoard  #rotatePiece: boolean |
| +<constructor>BlokusPiece(board: BlokusBoard)  +setRelatives(): void  +moveToLocation(x: int, y: int): void  +isRotatePiece(): Boolean  +rotateLeft(): void  +rotateRight(): void  +moveUp(): void  +moveDown(): void  +moveLeft(): void  +moveRight(): void  +hasAdjacent(): Boolean  +hasDiagonal(): Boolean  +placePiece(); void |

BlokusPiece is an abstract class, which holds an array of 1 or more BlokusSquares. One origin square is placed according to x and y coordinates on the board, and the other pieces are placed according to relative coordinates to the first piece. This allows rotation to take place very easily (implementation already in BlokusPiece). Checking the board coordinates should also be done in terms of the piece’s relative values. Relative coordinates are set by each of BlokusPiece’s subclasses. Each subclass also sets the value of rotatePiece, which dictates whether or not the piece is allowed to rotate (the Square, for example, shouldn’t be allowed to rotate).

Once rotation and move methods work, the next step is to get boundaries of the board to be checked when moving or rotating the piece. So, the piece should not move or rotate if it would land the piece out of bounds. More methods might need to be added to BlokusPiece for this.

Next, get the one piece to be added to the board by pressing enter, and spawn a new piece. Pieces should not be able to be added where there already exists a piece.

hasAdjacent and hasDiagonal should be used to determine if a piece can actually be placed, that is, if hasDiagonal is equal to true, and hasAdjacent is false. In the future, we’ll have to check to make sure these methods support player colors, too.

To test your code, you can create instances of BlokusPiece in BlokusGame, and call its methods. Feel free to change the design of anything if you think that will work better.