CheckerBoard(int aDimension) – testCheckerBoard_Con_8x8

Input: aDimension = 8

State: N/A

Output:

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

CheckerBoard(int aDimension) – testCheckerBoard_Con_12x12

Input: aDimension = 12

State: N/A

Output:

State:

x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*	
	*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*

<table>
<tr><td></td><td><table><tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td></tr></table></td></tr>
</table>

CheckerBoard(int aDimension) – testCheckerboard_Con_16x16

<table>
<tr><td>Input: aDimension = 16

State: N/A</td><td>Output:

State:
<table>
<tr><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td></tr>
<tr><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td></tr>
<tr><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td></tr>
<tr><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td></tr>
<tr><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td></tr>
<tr><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td></tr>
<tr><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td><td>x</td><td>*</td></tr>
<tr><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td></tr>
<tr><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td><td></td><td>*</td></tr>
<tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td></tr>
<tr><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td></tr>
<tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td></tr>
<tr><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td></tr>
<tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td></tr>
<tr><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td></tr>
<tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td></tr>
</table></td></tr>
</table>

whatsAtPos(BoardPosition pos) – testWhatsAtPos_MinRow_MaxCol_Black_Tile

Input: pos = (0, 7)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘*’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

whatsAtPos(BoardPosition pos) – testWhatsAtPos_MaxRow_MaxCol_Player_Piece_o

Input: pos = (7, 7)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*

Output: ‘o’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*

*	o	*	o	*	o	*	o

*	o	*	o	*	o	*	o

whatsAtPos(BoardPosition pos) – testWhatsAtPos_MaxRow_MinCol_Black_Tile

Input: pos = (7, 0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘*‘

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

whatsAtPos(BoardPosition pos) – testWhatsAtPos_MinRow_MinCol_Player_Piece_x

Input: pos = (0, 0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘x’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

whatsAtPos(BoardPosition pos) – testWhatsAtPos_Row4_Col4_Empty_Space

Input: pos = (4, 4)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: ‘ ’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Row0_Col0_blank

Input: pos = (0, 0), player = ‘ ’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: void

State:

	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Row15_Col15_x

Input: pos = (15, 15), player = ‘x’

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o

Output: void

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	x

placePiece(BoardPosition pos, char player) – testPlacePiece_Row1_Col15_blank

Input: pos = (1, 15), player = ‘ ’

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o

Output: void

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Row15_Col1_blank

Input: pos = (15, 1), player = ‘ ’

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o

Output: void

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*		*	o	*	o	*	o	*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Row0_Col1_blank

Input: pos = (0, 1), player = ‘ ’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: void

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_Row3_Col1_x

Input: pos = (3, 1), player = ‘x‘

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: void

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*	x	*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

getPieceCounts(void) – testGetPieceCounts_Num_Of_Pieces_12

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: Gets Hashmap with the player’s corresponding pieces and the number of pieces each player has: ‘x’ = 12 and ‘o’ = 12
State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

getViableDirections(void) – testGetViableDirections

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*

Output: Gets HashMap holding the directions that each player can move.

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*

*	o	*	o	*	o	*	o

*	o	*	o	*	o	*	o

addViableDirections(char player, DirectionEnum dir) – testAddViableDirections_player_x_SW_

Input: player = ‘x’
dir = DirectionEnum.SW

State:
viableDirections Hashmap
‘x’ = emptyArrayList
‘o’ = emptyArrayList
‘X’ = emptyArrayList
‘O’ = emptyArrayList

Output: Player x can move SW

State:
viableDirections Hashmap
‘x’ = DirectionEnum.SW
‘o’ = emptyArrayList
‘X’ = emptyArrayList
‘O’ = emptyArrayList

getRowNum(void) – testGetRowNum_Size_8x8

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: 8

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

getColNum(void) – testGetColNum_Size_8x8

Input: N/A

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: 8

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

checkPlayerWin(Character player) – testCheckPlayerWin_x_defaultBoard

Input: player = ‘x’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: false

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

checkPlayerWin(Character player) – testCheckPlayerWin_o_no_X

Input: player = ‘o’

State:

	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: true

State:

	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Size_8x8_Row_0_Col_0

Input: posOfPlayer = (0,0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: piece at 0, 0 is capitalized (crowned)

State:

X	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Size_16x16_Row_15_Col_15

Input: posOfPlayer = (15,15)

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o

Output: Piece at BoardPosition 15, 15 is capitalized (crowned)

State:

x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x	*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*	x	*	x	*	x	*	x	*
*		*		*		*		*		*		*		*	
	*		*		*		*		*		*		*		*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o	*	o	*	o	*	O

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Size_8x8_blackTile_Row_3_Col_0

Input: posOfPlayer = (3,0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: Black tile at BoardPosition 3,0 is not capitalized (crowned)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_Size_8x8_Row_0_Col_4

Input: posOfPlayer = (0,4)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: Piece at BoardPosition 0,4 is capitalized (crowned)

State:

x	*	x	*	X	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece_Size_8x8_blackTile

Input: startingPos.row = 3
startingPos.col = 2
dir = DirectionEnum.SE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: The black tile at BoardPosition 3,2 has moved to 4,3. 3,2 is now blank.

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*				*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece_Size_8x8_player2

Input: startingPos.row = 5
startingPos.col = 1
dir = DirectionEnum.NE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*

Output: The piece at BoardPosition 5,1 has moved to 4,2. 5,1 is now blank.

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*	o	*		*		*
*		*	o	*	o	*	o
o	*	o	*	o	*	o	*

*	o	*	o	*	o	*	o

*	o	*	o	*	o	*	o

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece_Size_8x8_player1

Input: startingPos.row = 2
startingPos.col = 0
dir = DirectionEnum.SE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: The piece at BoardPosition 2,0 has moved to 3,1. 2,0 is now blank.

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
	*	x	*	x	*	x	*
*	x	*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_Row_2_Col_0_SE

Input: pos = (2, 0), dir = SE

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*	o	*		*		*	
	*		*		*		*
*		*	o	*	o	*	o

Output:

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
	*	x	*	x	*	x	*
*		*		*		*	
	*	x	*		*		*
*		*	o	*	o	*	o

o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

O piece count = 11

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_Row_5_Col_3_SW_crown

Input: pos = (5, 3), dir = SW

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	x	*	o	*	o
o	*	o	*	o	*	o	*
*		*	o	*	o	*	o

Output:

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*		*	o	*	o
o	*		*	o	*	o	*
*	X	*		*	o	*	o

O piece count = 9
X crown piece

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_Row_2_Col_2_NW_crown

Input: pos = (2, 0), dir = SE

State:

	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	o	*	x	*	x	*
*		*		*		*	

Output:

State:

O	*	x	*	x	*	x	*
*		*	x	*	x	*	x
x	*		*	x	*	x	*
*		*		*		*	

	*		*		*		*
*		*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

	*		*		*		*
*		*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

X piece count = 9

playerLostPieces(int numPieces, char player, HashMap pieceCounts) – testPlayerLostPieces_Remove_1_x

Input: numPieces = 1 player = ‘x’ State: pieceCounts HashMap ‘x’ = 12 ‘o’ = 12	Output: The number of pieces for player ‘x’ have decreased by 1 and the new value of the remaining pieces becomes stored in the pieceCounts HashMap State: pieceCounts HashMap ‘x’ = 11 ‘o’ = 12

scanSurroundingPositions(BoardPosition startingPos) – testScanSurroundingPositions_Row_7_Col_7

Input: startingPos = (7,7)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*

Output: DirectionEnum.SE = null
DirectionEnum.SW = null
DirectionEnum.NE = null
DirectionEnum.NW = ‘o’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*

*	o	*	o	*	o	*	o

*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_Row_0_Col_0

Input: startingPos = (0,0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: DirectionEnum.SE = ‘x’
DirectionEnum.SW = null
DirectionEnum.NE = null
DirectionEnum.NW = null

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_Row_1_Col_3

Input: startingPos = (1,3)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output:DirectionEnum.SE = ‘x’
DirectionEnum.SW = ‘x’
DirectionEnum.NE = ‘x’
DirectionEnum.NW = ‘x’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_Row_0_Col_7

Input: startingPos = (0,7)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output:DirectionEnum.SE = ‘null’
DirectionEnum.SW = ‘*’
DirectionEnum.NE = ‘null‘
DirectionEnum.NW = ‘null‘

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

scanSurroundingPositions(BoardPosition startingPos) – testScanSurroundingPositions_Row_2_Col_0

Input: startingPos = (2,0)

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: DirectionEnum.SE = ‘ ‘
DirectionEnum.SW = ‘null’
DirectionEnum.NE = ‘x’
DirectionEnum.NW = ‘null’

State:

x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

getDirection(DirectionEnum dir) – testGetDirection_SE

Input: dir = DirectionEnum.SE

State: N/A

Output: BoardPosition b.row = 1
BoardPosition b.col = 1

State: N/A

What tests did each team member write? Just tell me the names of the functions (unless for some reason multiple team members wrote functions for the same method. In that case, tell me which tests specifically by giving me the test names)

[member 1]
Jarrett Wilson
	GetColNum (1)
	Checkerboard(3)
	scanSurroundingPositions(5)
	addViableDirections(1)

[member 2] Will Lovin	GetRowNum (1) movePiece (3) crownPiece (4) getDirection (1) getViableDirections (1)

[member 3]
Curren Patel
	GetPieceCounts (1)
	whatsAtPos (5)
	checkPlayerWin(2)
	playerLostPieces(1)

[member 4]
Michael Funchess
	jumpPiece(3)
	placePiece(6)