CheckerBoard(int aDimension) – testCheckerBoard_Con_10x10

Input:aDimension = 10 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
	*		*		*		*		*
*		*		*		*		*	
o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o	*	o

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_16x16

Input:aDimension = 16

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	*	x	*	x
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

whatsAtPos(BoardPosition pos) – testWhatsAtPos_1_1_x

Input:

pos = (1, 1)

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:

[state of the board is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_minRow_2_x

Input:

pos = (0, 2)

State:

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

Output: ‘x’

State: [state of the board is unchanged]

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

whatsAtPos(BoardPosition pos) – testWhatsAtPos_3_maxColumn

Input:

pos = (3, 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: [state of the board is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_maxRow_6_*

Input:

pos = (7,6)

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: [state of the board is unchanged]

whatsAtPos(BoardPosition pos) – testWhatsAtPos_6_minCol_o

Input:

pos = (6,0)

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:

o

State: [state of the board is unchanged]

placePiece(BoardPosition pos, char player) – testPlacePiece_6_6_x

Input:

pos = (6, 6)

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: N/A

State:

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

placePiece(BoardPosition pos, char player) – testPlacePiece_4_1_o

Input:

pos = (4, 1)

player = ‘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

Output: 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	*	o

placePiece(BoardPosition pos, char player) – testPlacePiece_minRow_5_x

Input:

pos = (0, 5)

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: N/A

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

placePiece(BoardPosition pos, char player) – testPlacePiece_2_minColumn_o

Input:

pos = (2, 0)

player = ‘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

Output: N/A

State:

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

placePiece(BoardPosition pos, char player) – testPlacePiece_4_maxColumn_x

Input:

pos = (4, 7)

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: N/A

State:

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

placePiece(BoardPosition pos, char player) – testPlacePiece_maxRow_2_x

Input:

pos = (7, 2)

player = ‘o’

State:

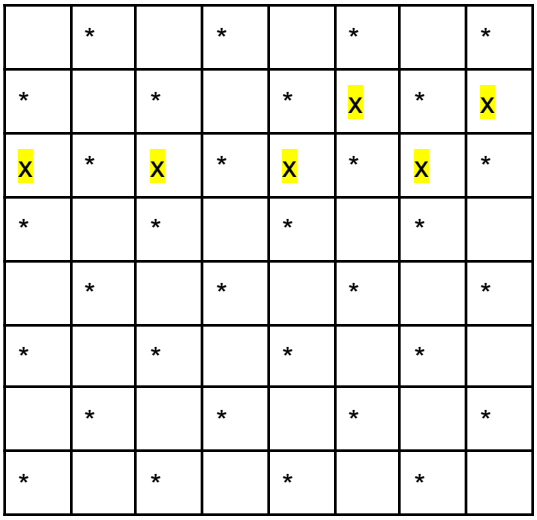
x	*	x	*	x	*	x	*
*	x	*	x	*	x	*	x
x	*	x	*	x	*	x	*
*		*		*		*	
	*		*		*		*
*	o	*	o	*	o	*	o
o	*	o	*	o	*	o	*
*	o	*	o	*	o	*	o

Output: N/A

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	*	o

getPieceCounts(void) – testGetPieceCounts_x_6_o_min

Input: N/A State: pieceCount = {('x', 6), ('o', 0)} 	Output: pieceCount = {('x', 6), ('o', 0)} State: pieceCount = [pieceCount hashmap is unchanged] [state of board is unchanged]

getPieceCounts(void) – testGetPieceCounts_x_max_o_3

Input: N/A State: pieceCount = {('x', 12), ('o', 8)}	Output: pieceCount = {('x', 12), ('o', 8)} State: pieceCount = [pieceCount hashmap is unchanged]

x * x * x * x * * x * x * x * x x * x * x * x * * * * * * * * * * * * * * * * o * o * o * o o * o * o * o * * * * * * * *	[state of board is unchanged]

getViableDirections(void) – testGetViableDirections_PLAYER_ONE

Input: None State: x = PLAYER_ONE viableDirections = {‘NE’,’NW’, ‘SE’, ‘SW’}	Output: viableDirections = {‘NE’,’NW’,‘SE’, ‘SW’} State:

addViableDirections(char player, DirectionEnum dir) – testAddViableDirections_PLAYER_ONE

Input: player = ‘x’ dir = NW State: x = PLAYER_ONE viableDirections = null	Output: viableDirections = NW State:

getRowNum(void) – getRowNumTest_8

Input: none State: 8x8 board x * x * x * x * * x * x * x * x x * x * x * x * * * * * * *

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

Output: 8 State: [state of the board is unchanged]

getColNum(void) – getColNumTest_8

Input: State: 8x8 board <table><tr><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></tr><tr><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></tr><tr><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></tr><tr><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>x</td><td>*</td></tr><tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td></td></tr></table>	Output: 8 State: [state of the board is unchanged]

checkPlayerWin(Character player) – testCheckPlayerWin_PLAYER_ONE_Win

Input:

State:

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

pieceCount = {('x', 12), ('o', 0)}

Output:

True

State:

checkPlayerWin(Character player) – testCheckPlayerWin_PLAYER_TWO_NoWin

Input:

State:

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

Output:

False

State:

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

pieceCount = {('x', 12), ('o', 0)}

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_MinRow_MinColumn_x

Input: posOfPlayer = (0, 0)

State: 8x8 board

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

Output: N/A

State: 8x8 board

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

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_5_1_o

Input: posOfPlayer = (5, 1)

State: 8x8 board

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

Output: N/A

State: 8x8 board

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

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_1_MaxColumn_x

Input: posOfPlayer = (1, 7)

State: 8x8 board

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

Output: N/A

State: 8x8 board

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	*	

crownPiece(BoardPosition posOfPlayer) – testCrownPiece_MaxRow_1_o

Input: posOfPlayer = (7, 1)

State: 8x8 board

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

Output: N/A

State: 8x8 board

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

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece_maxRow_maxCol_X

Input:

startingPos = (6, 6)

dir = SE

State:

Call placePiece(startingPos, 'x')

Call placePiece(startingPos+SE, ' ')

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

Output:

movePiece = (7,7)

State:

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

movePiece(BoardPosition startingPos, DirectionEnum dir) – testMovePiece_minRow_minCol_O

Input:

startingPos = (1, 1)

dir = NW

State:

Call placePiece(startingPos, 'O')

Call placePiece(startingPos+NW, ' ')

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

Output:

movePiece = (0,0)

State:

O	*	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_2_2_kingBackwards_O

Input:

startingPos = (2,2)

dir = SE

State:

Call placePiece(startingPos, 'O')

Call placePiece(startingPos+SE, ' ')

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

Output:

movePiece = (3, 3)

State:

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

jumpPiece(BoardPosition startingPos, DirectionEnum dir) –
testJumpPiece_maxRow_maxCol_X

Input:

startingPos = (5, 5)

dir = SE

State:

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

Output:

jumpPiece = (7,7)

State:

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

jumpPiece(BoardPosition startingPos, DirectionEnum dir) – testJumpPiece_minRow_minCol_O

Input:

startingPos = (2, 2)

dir = NW

State:

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

Output:

jumpPiece(0,0)

State:

O	*	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_3_3_kingBackwardsO

Input:

startingPos = (1, 1)

dir = SE

State:

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

Output:

jumpPiece(3,3)

State:

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

playerLostPieces(int numPieces, char player, HashMap pieceCounts) –
testPlayerLostPieces_1_x_1

Input:

numPieces = 1

player = ‘x’

pieceCounts = {(‘x’, 1)}

State:

Output: N/A

pieceCounts.get(‘x’) = 0

State:

pieceCounts = {(x, 1), (o, 3)}	pieceCounts = {(x, 0), (o, 3)}

scanSurroundingPositions(BoardPosition startingPos) – testScanSurroundingPositions_2_4_x

Input:

startingPos = (2, 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:

scanSurroundingPositions = {(NE: ‘x’), (NW: ‘x’), (SE: ‘ ‘), (SW: ‘ ‘)}

State: [the state of the board is unchanged]

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_minRow_5_x

Input:

startingPos = (0, 5)

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:

scanSurroundingPositions = {(SE: ‘x‘), (SW: ‘x‘)}

State: [the state of the board is unchanged]

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_6_minColumn_x

Input:

startingPos = (6, 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:

scanSurroundingPositions = {(NE: ‘o‘), (SE: ‘o‘)}

State: [the state of the board is unchanged]

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_maxRow_3_o

Input:

startingPos = (7, 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:

scanSurroundingPositions = {(NE: ‘o‘), (NW: ‘o‘)}

State: [the state of the board is unchanged]

scanSurroundingPositions(BoardPosition startingPos) –
testScanSurroundingPositions_1_maxColumn_o

Input:

startingPos = (1, 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:

scanSurroundingPositions = {(NW: ‘x‘), (SW: ‘x‘)}

State: [the state of the board is unchanged]

getDirection(DirectionEnum dir) – testGetDirection_NE

Input: dir = NE State:	Output: new BoardPosition (-1, 1) State:

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)

James Schvaneveldt	CheckerBoard, placePiece, scanSurroundingPositions
Isaac Beres	☑ whatsAtPos ☐ movePiece (sorta) ☑ getRowNum ☑ getColNum
Nicolas Lozano	jumpPiece, checkPlayerWin, getViableDirections, addViableDirections
Liam Cassidy	crownPiece, playerLostPieces, getPieceCounts, getDirection