CheckerBoard(void) – [Test_Constructor_DefaultBoard_8x8]

Input: (8x8)

State: N/A

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

whatsAtPos(BoardPosition pos) – [Test_whatsAtPos_EmptySquare_row3_column1]

Input:
pos = (3, 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: ‘ ‘

State:

[State of the board is unchanged]

whatsAtPos(BoardPosition pos) – [Test_whatsAtPos_ BlackPiece_row0_column0]

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:

[State of the board is unchanged]

whatsAtPos(BoardPosition pos) – [Test_whatsAtPos_WhitePiece_row5_column1]

Input:
pos = (5,1)
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: ‘o’

State:
[State of the board is unchanged]

whatsAtPos(BoardPosition pos) – [Test_whatsAtPos_Black_Tile_row0_column1]

Input:
pos = (0,1)
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: '*'

State:
[State of the board is unchanged]

whatsAtPos(BoardPosition pos) – [Test_whatsAtPos_OutOfBounds_row8_column8]

Input:
pos = (8,8)
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

Output:
Throws an exception

State:
[State of the board is unchanged]

placePiece(BoardPosition pos, char player) – [Test_placePiece_ValidPlacement_row3_column1]

Input:
pos = (3,1)
player = 'x'
State:

	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	

Output: N/A

State:

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

placePiece(BoardPosition pos, char player) –
[Test_placePiece_InvalidPlacement_Black_Tile_row0_column1]

Input:
pos = (0, 1)
player = ‘x’
State:

	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	

Output:
Exception Thrown

State:
[State of the board is unchanged]

placePiece(BoardPosition pos, char player) – [Test_placePiece_OccupiedSquare_row3_column3]

Input:
pos = (3,3)
player = 'o'
State:

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

Output:
Exception thrown

State:
[State of the board is unchanged]

placePiece(BoardPosition pos, char player) – [Test_placePiece_InvalidCharacter_row4_column4]

Input:
pos = (4,4)
player = 'z'
State:

	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	

Output:
Exception Thrown

State:
[State of the board is unchanged]

placePiece(BoardPosition pos, char player) – [Test_placePiece_CrownedPiece_row2_column0]

Input:
pos = (2,0)
player = 'X'
State:

	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	

Output: N/A

State:

	*		*		*		*
*		*		*		*	
X	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	
	*		*		*		*
*		*		*		*	

getPieceCounts(void) – [Test_getPieceCounts_AfterPlacement]

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: {12, 12}
(12 black pieces, 12 white pieces)

State:
[State of the board is unchanged]

getViableDirections(void) – [Test_getViableDirections_InitialSetup]

Input: N/A

State: N/A

Output:
'x' = [DirectionEnum.SE, DirectionEnum.SW]
'X' = [DirectionEnum.SE, DirectionEnum.SW, DirectionEnum.NE, DirectionEnum.NW]
'o' = [DirectionEnum.NE, DirectionEnum.NW]
'O' = [DirectionEnum.SE, DirectionEnum.SW, DirectionEnum.NE, DirectionEnum.NW]

State:
[State of the board is unchanged]

checkPlayerWin(Character player) – [Test_checkPlayerWin_PlayerWins]

Input: player = 'x' State:	Output: True State: [State of the board is unchanged]

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

checkPlayerWin(Character player) – [Test_checkPlayerWin_PlayerNotWin]

Input: player = 'x' State:	Output: False State: [State of the board is unchanged]	Input: player = 'o' State:

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

crownPiece(BoardPosition posOfPlayer) – [Test_crownPiece_PLAYER_ONE_row2_column6]

Input:
posOfPlayer = (2, 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: 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

crownPiece(BoardPosition posOfPlayer) – [Test_crownPiece_PLAYER_TWO_row6_column0]

Input:
posOfPlayer = (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: 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

crownPiece(BoardPosition posOfPlayer) – [Test_crownPiece_NonPlayer_row0_column1]

Input:
posOfPlayer = (0, 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: Exception Thrown

State:
[State of the board is unchanged]

movePiece(BoardPosition startingPos, DirectionEnum dir) –
[Test_movePiece_ValidMove_SE_row2_column2]

Input:
startingPos = (2, 2)
dir = 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:
BoardPosition(3, 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

movePiece(BoardPosition startingPos, DirectionEnum dir) –
[Test_movePiece_ValidMove_NW_row5_column1]

Input:
startingPos = (5,1)
dir = NW

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:
BoardPosition(4,0)

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

movePiece(BoardPosition startingPos, DirectionEnum dir) –
[Test_movePiece_Crowned_SE_row0_column2]

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

State:

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

Output:
BoardPosition(1,3)

State:

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

jumpPiece(BoardPosition startingPos, DirectionEnum dir) –
[Test_jumpPiece_ValidJump_NW_row4_column2]

Input:
startingPos = (4,2)
dir = NW

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:
BoardPosition(2,0)

State:

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

jumpPiece(BoardPosition startingPos, DirectionEnum dir) –
[Test_jumpPiece_ValidJump_SE_row2_column4]

Input:
startingPos = (2,4)
dir = 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:
BoardPosition(4,6)

State:

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

jumpPiece(BoardPosition startingPos, DirectionEnum dir) –
[Test_jumpPiece_Crowned_SE_row0_column2]

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

State:

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

Output:
BoardPosition(2,4)

State:

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

scanSurroundingPositions(BoardPosition startingPos) –
[Test_scanSurroundingPositions_NonBoundary_row2_column2]

Input:
startingPos = (2,2)

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.NW: 'x'
DirectionEnum.NE: 'x'
DirectionEnum.SW: ' '
DirectionEnum.SE: ' '

State:
[State of the board is unchanged]

scanSurroundingPositions(BoardPosition startingPos) –
[Test_scanSurrondingPositions_row6_column6]

Input:
startingPos = (6,6)

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:
DirectionEnum.NW: 'o'
DirectionEnum.NE: 'o'
DirectionEnum.SW: 'o'
DirectionEnum.SE: ' o'

State:
[State of the board is unchanged]

scanSurroundingPositions(BoardPosition startingPos) –

[Test_scanSurrondingPositions_Crowned_row1_column3]

Input: startingPos = (1,3) State: <table><tr><td>x</td><td>*</td><td>O</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></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></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></tr><tr><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td><td>*</td><td>o</td></tr></table>	Output: DirectionEnum.NW = ‘O’ DirectionEnum.NE = ‘x’ DirectionEnum.SW = ‘x‘ DirectionEnum.SE = ‘ ‘ State: [State of the board is unchanged]

getDirection(DirectionEnum dir) – [Test_getDirection_ValidDirection_SE]

Input: dir = SE State: N/A	Output: new BoardPosition(1,1) State: [State of the board is unchanged]

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)

Nathan McMillan	movePiece(BoardPosition, DirectionEnum) jumpPiece(BoardPosition, DirectionEnum) scanSurroundingPositions(BoardPosition) getDirection(DirectionEnum)
Fahed-Dan Geravi	CheckerBoard(int) – 1 test whatsAtPos(BoardPosition) – 5 tests getPieceCounts(void) - 1 test checkPlayerWin(Character) – 2 test
Shaun Whitt	placePiece(BoardPosition, char) getViableDirections(void) crownPiece(BoardPosition)
[member 4]	