OthelloBoard(void) – [testConstructor_InitialSetup]

Input: new OthelloBoard() State: N/A	Output: N/A State: 0 1 2 3 4 5 6 7 +---------- 0 | 1 | 2 | 3 | X O 4 | O X 5 | 6 | 7 |

whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_3_3_PlayerOne]

Input: (3,3) State:	Output: X State:

State:

			X	O			
			O	X			

State:

			X	O			
			O	X			

whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_3_4_PlayerTwo]

Input: (3,4) State:	Output: O State:

State:

			X	O			
			O	X			

State:

			X	O			
			O	X			

whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_0_0_Empty]

Input: (0,0)

State:

			X	O			
			O	X			

Output: ‘ ‘

State:

			X	O			
			O	X			

whatsAtPos(BoardPosition pos) –[testWhatsAtPos_DefaultState_7_7_Empty]

Input: (7,7)

State:

			X	O			
			O	X			

Output: ‘ ‘

State:

			X	O			
			O	X			

whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_4_3_PlayerTwo]

Input: (4,3)

State:

			X	O			
			O	X			

Output: ‘O’

State:

			X	O			
			O	X			

placeToken(char p, BoardPosition pos) – [testPlaceToken_EmptyPosition_2_2_PlayerOne]

Input: p = ‘X’ AND pos = (2,2)

State:

			X	O			
			O	X			

Output: N/A

State:

		X					
			X	O			
			O	X			

placeToken(char p, BoardPosition pos) – [testPlaceToken_EmptyPosition_5_5_PlayerTwo]

Input: p = 'O', pos = (5, 5)

State:

			X	O			
			O	X			

Output: N/A (VOID)

State:

		X					
			X	O			
			O	X			
					O		

placeToken(char p, BoardPosition pos) – [testPlaceToken_OccupiedPosition_3_3_PlayerTwo]

Input: p = 'O', pos = (3, 3)

State:

			X	O			
			O	X			

Output: N/A

State:

			O	O			
			O	X			

placeToken(char p, BoardPosition pos) – [testPlaceToken_EdgePosition_0_7_PlayerOne]

Input: p = 'X', pos = (0, 7)
State:

			X	O			
			O	X			

Output:N/A
State:

							O
			X	O			
			O	X			

placeToken(char p, BoardPosition pos) – [testPlaceToken_CornerPosition_7_0_PlayerTwo]

Input: p = 'O', pos = (7, 0)

State:

			X	O			
			O	X			

Output: N/A

State:

			X	O			
			O	X			
O							

flipVertDirections(BoardPosition startingPos) – [testFlipVertDirections_NoFlip_1_3_PlayerOne]

Input: startingPos = (1, 3)

State:

		X					
		O					

Output: N/A

State:

		X					
		O					

flipVertDirections(BoardPosition startingPos) –
testFlipVertDirections_BottomToTop_5_3_PlayerTwo]

Input: startingPos = (5, 3)

State:

			O				
			X				
			X				
			O				

Output:

State:

			O				
			O				
			O				
			O				
			O				

flipVertDirections(BoardPosition startingPos) –
[testFlipVertDirections_TopToBottom_2_3_PlayerOne]

Input: startingPos = (2, 3)

State:

		X					
			O				
			O				
			X				

Output:

State:

		X					
			X				
			X				
			X				
			X				

flipHoriDirections(BoardPosition startingPos) –
[testFlipHoriDirections_LeftToRight_4_1_PlayerOne]

Input: startingPos = (4, 1)

State:

			X	O			
X	O	O	X				

Output: N/A

State:

			X	O			
X	X	X	X				

flipHoriDirections(BoardPosition startingPos) – [testFlipHoriDirections_RightToLeft_4_4_PlayerTwo]

Input: startingPos = (4, 4)

State:

			X	O			
	O	X	X		O		

Output: N/A

State:

			X	O			
	O	O	O	O	O		

flipHoriDirections(BoardPosition startingPos) – [testFlipHoriDirections_NoFlip_2_3_PlayerOne]

Input: startingPos = (2, 3)

State:

			X		O		
			X	O			
			O	X			

Output: N/A

State:

			X	X	O		
			X	O			
			O	X			

flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_DownRight_2_2_PlayerOne]

Input: startingPos = (2, 2)

State:

		X					
			X				
				O			
					O		
						X	

Output: N/A

State:

		X					
			X				
				X			
					X		
						X	

flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_UpLeft_5_5_PlayerTwo]

Input: startingPos = (5, 5)

State:

				O			
			X				
		X					
					O		

Output: N/A

State:

				O			
			O				
		O					
					O		

flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_NoFlip_2_2_PlayerOne]

Input: startingPos = (2, 2)

State:

8	0	1	2	3	4	5	6	7
		X						
			X					
				O				

Output:N/A

State:

		X					
			X				
				O			

flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_DownLeft_2_4_PlayerOne]

Input: startingPos = (2, 4)

State:

				X			
					X		
				O			
			O				
		X					

Output: N/A

State:

				X			
					X		
				O			
			O				
		X					

flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_UpRight_5_1_PlayerTwo]

Input: startingPos = (5, 1)

State:

					O		
				X			
			X				
		O					
	O						

Output: N/A

State:

					O		
				O			
			O				
		O					
	O						

isPositionValid(BoardPosition) – [testIsPositionValid_BoardHasEmptySpaces_ReturnsTrue]

Input: N/A , no parameters

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	O	X	X	X
X	X	X	O	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X		X

Output:TRUE

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	O	X	X	X
X	X	X	O	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X		X

isPositionValid(BoardPosition) – [testIsPositionValid_BoardFull_ReturnsFalse]

Input: N/A

State:

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
X	X	X	X	X	X	X	X
O	O	O	O	O	O	O	O
X	X	X	X	X	X	X	X
O	O	O	O	O	O	O	O

Output: FALSE

State:

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
X	X	X	X	X	X	X	X
O	O	O	O	O	O	O	O
X	X	X	X	X	X	X	X
O	O	O	O	O	O	O	O

checkPlayerWin(Character player) – [testCheckPlayerWin_OpponentHasNoTokens_PlayerOneWins]

Input: 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
X	X	X	X	X	X	X	X

Output: TRUE

State: same as before

checkPlayerWin(Character player) – [testCheckPlayerWin_OpponentHasTokens_PlayerOneLoses]

Input: player = 'X'

State:

X	X	X	X	X	X	X	X
O	O	O	O	O	O	O	O
X	X	X	X	X	X	X	X
X	X	O	O	O	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X

Output: FALSE

State:
same as before

isPlayerAtPos(BoardPosition pos, char player) –
[testIsPlayerAtPos_PlayerOneAt_3_3_ReturnsTrue]

Input: pos = (3, 3), player = 'X'

State: DEFAULT

			X	O			
			O	X			

Output: TRUE

State: same as before

			X	O			
			O	X			

isPlayerAtPos(BoardPosition pos, char player) –
[testIsPlayerAtPos_PlayerTwoAt_3_4_ReturnsTrue]

Input: pos = (3, 4), player = 'O'

State: DEFAULT

			X	O			
			O	X			

Output: TRUE

State: same as before

			X	O			
			O	X			

isPlayerAtPos(BoardPosition pos, char player) –
[testIsPlayerAtPos_EmptyAt_0_0_ReturnsFalse]

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

State: DEFAULT

			X	O			
			O	X			

Output: FALSE

State: same as before

			X	O			
			O	X			

isPlayerAtPos(BoardPosition pos, char player) –
[testIsPlayerAtPos_PlayerTwoAt_4_3_ReturnsTrue]

Input: pos = (4, 3), player = 'O'

State: DEFAULT

			X	O			
			O	X			

Output: TRUE

State: same as before

			X	O			
			O	X			

isPlayerAtPos(BoardPosition pos, char player) –
[testIsPlayerAtPos_PlayerOneAt_4_4_ReturnsTrue]

Input: pos = (4, 4), player = 'X'

State: DEFAULT

			X	O			
			O	X			

Output: TRUE

State: same as before

			X	O			
			O	X			

getScores(void) – [testGetScores_DefaultStartState_Returns2Each]

Input: N/A

State: DEFAULT BOARD

			X	O			
			O	X			

Output: { 'X' = 2, 'O' = 2 }

State: BOARD IS UNCHANGED

			X	O			
			O	X			

getScores(void) – [testGetScores_CustomBoardState_MixedCounts]

Input:
placeToken('X', (0,0)
placeToken('X', (1,1)
placeToken('X', (3,3)
placeToken('X', (4,4)
placeToken('O', (2,2)
placeToken('O', (4,3)
placeToken('O', (3,4)
State:

X							
	X						
		O					
			X	O			
			O	X			

Output: { 'X' = 4, 'O' = 3 }

State: same as before

X							
	X						
		O					
			X	O			
			O	X			

toString(void) – [testToString_DefaultStartState_DisplaysCorrectBoardAndScore]

Input: N/A9just call toString)

State: default board

			X	O			
			O	X			

Output: Score: X - 2 | O - 2

State:

			X	O			
			O	X			