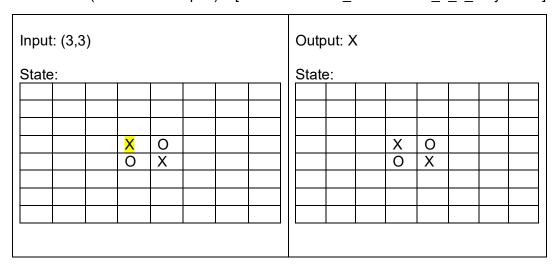
OthelloBoard(void) – [testConstructor_InitialSetup]

Input: new OthelloBoard()	Output: N/A
pati new earenegeara()	
State: N/A	State: 0 1 2 3 4 5 6 7
	+ 0 I
	1
	2 3 X O
	4 i O X
	5 6
	71

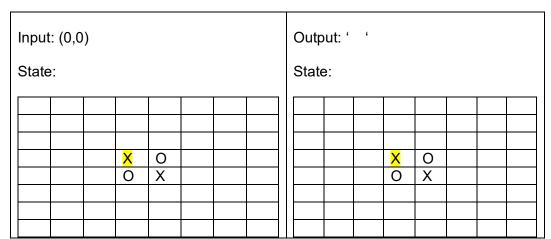
whatsAtPos(BoardPosition pos) - [testWhatsAtPos_DefaultState_3_3_PlayerOne]



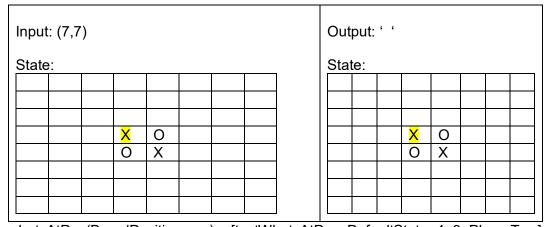
whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_3_4_PlayerTwo]

Input: (3,4)		Outp	ut: C)				
State:		State	e:					
X					X O	0 X		

whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_0_0_Empty]



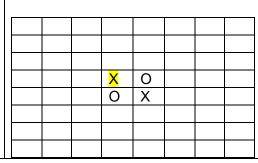
whatsAtPos(BoardPosition pos) –[testWhatsAtPos_DefaultState_7_7_Empty]



whatsAtPos(BoardPosition pos) – [testWhatsAtPos_DefaultState_4_3_PlayerTwo]

Output: 'O'

State:



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

Input: p = 'X' AND pos = (2,2)



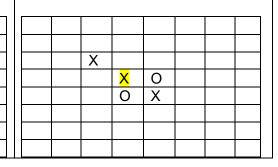
0

Χ

0

Output: N/A

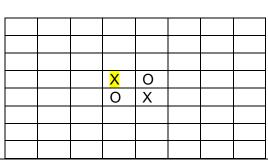
State:



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

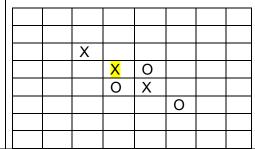
Input: p = 'O', pos = (5, 5)

State:



Output: N/A (VOID)

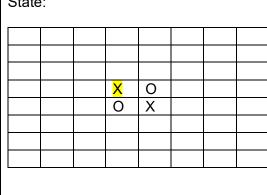
State:



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

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

State:



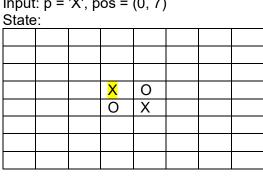
Output: N/A

State:

)			
)			
0	0		
0	X		

placeToken(char p, BoardPosition pos) – [testPlaceToken EdgePosition 0 7 PlayerOne]

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

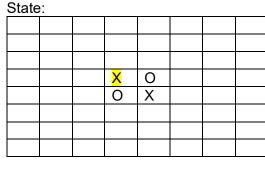


Output:N/A

State	э:				
					0
		X	0		
		0	Х		
	•			•	•

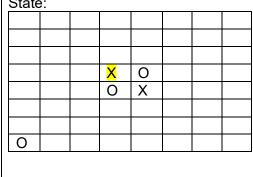
placeToken(char p, BoardPosition pos) – [testPlaceToken_CornerPosition_7_0_PlayerTwo]

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



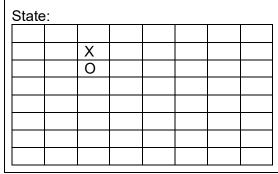
Output: N/A

State:



flipVertDirections(BoardPosition startingPos) – [testFlipVertDirections_NoFlip_1_3_PlayerOne]

Input: startingPos = (1, 3)



Output: N/A

State) :						
		Χ					
		0					
	•	•	•	•	•	•	•

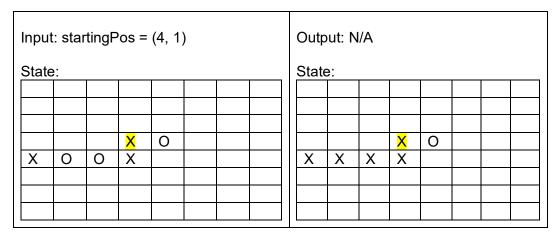
flipVertDirections(BoardPosition startingPos) testFlipVertDirections_BottomToTop_5_3_PlayerTwo]

Input: starti	ngPos =	(5, 3)		Outpu	ut:			
State:			 	State	:			
	0					0		
	X					0		
	Х					0		
	0					0		
						0		

flipVertDirections(BoardPosition startingPos) – [testFlipVertDirections_TopToBottom_2_3_PlayerOne]

Input: s	tartingF	Pos =	(2, 3)			Outp	out:							
State:	State:							State:							
	X								Χ						
										Х					
		0								X					
		0								Х					
		Χ								Х					

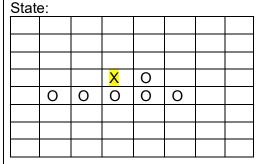
flipHoriDirections(BoardPosition startingPos) – [testFlipHoriDirections_LeftToRight_4_1_PlayerOne]



flipHoriDirections(BoardPosition startingPos) -[testFlipHoriDirections_RightToLeft_4_4_PlayerTwo]

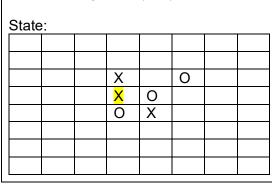
Input: startingPos = (4, 4) State: 0 0 Χ Χ 0

Output: N/A



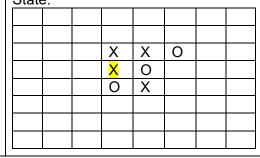
flipHoriDirections(BoardPosition startingPos) – [testFlipHoriDirections NoFlip 2 3 PlayerOne]

Input: startingPos = (2, 3)



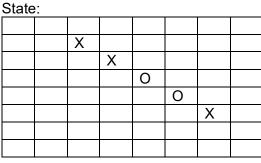
Output: N/A

State:



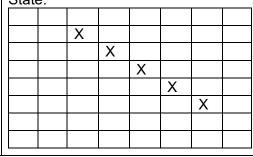
flipDiagDirections(BoardPosition startingPos) -[testFlipDiagDirections DownRight 2 2 PlayerOne]

Input: startingPos = (2, 2)



Output: N/A

State:

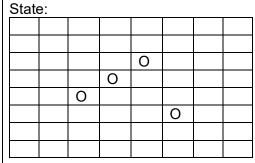


flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_UpLeft_5_5_PlayerTwo]

Input: startingPos = (5, 5)

State	e:					
				0		
			Χ			
		Χ				
					0	

Output: N/A



flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_NoFlip_2_2_PlayerOne]

Input: startingPos = (2, 2)

State:

0

1

Χ

2 3

Χ

0

5 6 Output:N/A

State:

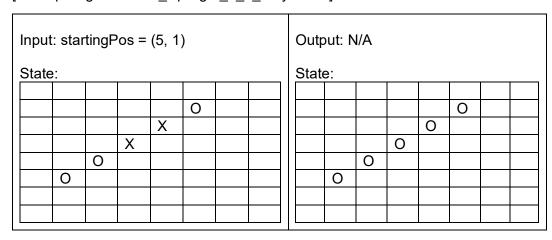
7

· ·				
Χ				
	Χ			
		0		
		X	 	

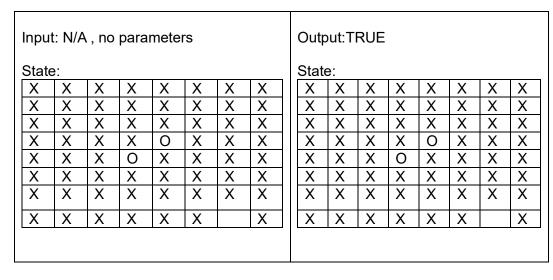
flipDiagDirections(BoardPosition startingPos) -[testFlipDiagDirections DownLeft 2 4 PlayerOne]

Input:	startingF	os =	(2, 4	.)			Outp	out: N	N/A					
State:	State:						State:							
			Χ								Х			
				X								Χ		
			0								0			
		0								0				
	X								Χ					

flipDiagDirections(BoardPosition startingPos) – [testFlipDiagDirections_UpRight_5_1_PlayerTwo]



 $is Position Valid (Board Position) - [test Is Position Valid_Board Has Empty Spaces_Returns True] \\$



isPositionValid(BoardPosition) – [testIsPositionValid_BoardFull_ReturnsFalse]

Input	Input: N/A									
State	e:									
X	Χ	Χ	Χ	Χ	Χ	Χ	Χ			
X	Χ	Χ	Χ	Χ	Χ	Χ	Χ			
0	0	0	0	0	0	0	0			
0	0	0	0	0	0	0	0			
X	Χ	Χ	Χ	Х	Χ	Χ	Χ			
0	0	0	0	0	0	0	0			
X	Χ	Χ	Χ	Х	Χ	Χ	Χ			
0	0	0	0	0	0	0	0			

Outp	Output: FALSE											
State	State:											
X X X X X X X X												
Χ	Χ	Χ	Χ	Χ	Χ	Χ	Χ					
0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	0					
Χ	Χ	Χ	Χ	Χ	Χ	Χ	Χ					
0	0	0	0	0	0	0	0					
Χ	Χ	Χ	Χ	Χ	Χ	Χ	Х					
0	0	0	0	0	0	0	0					

checkPlayerWin(Character player) -[testCheckPlayerWin_OpponentHasNoTokens_PlayerOneWins]

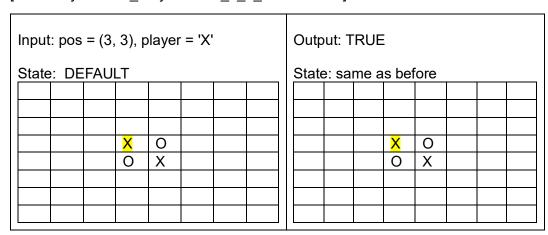
Inpu	t: pla	yer =	'X'					Output: TF
State	e:							State: sam
X	Х	Χ	Χ	X	Χ	Χ	Χ	
X	Х	Х	Х	X	Х	Х	Х	
X	Х	Х	Х	Х	Х	Х	Х	
X	Х	Х	X	Х	Х	Х	Х	
X	Х	Х	Х	Х	Х	Х	Х	
X	Х	Х	Х	Х	Х	Х	Х	
X	Х	Х	Х	Х	Х	Х	Х	
Χ	Х	Χ	Χ	Х	Х	Х	Х	

RUE

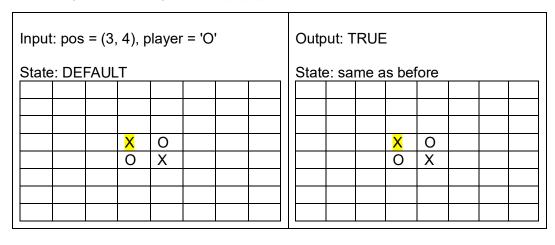
ne as before

Inpu	t: pla	yer =	Output: FALSE					
State	e:							State:
Χ	Х	Χ	Х	Х	Χ	Χ	Χ	same as before
0	0	0	0	0	0	0	0]
Χ	Х	Х	Х	Х	Х	Х	Х	
Χ	Х	0	0	0	Х	Х	Х	
Χ	Х	Х	Х	Х	Х	Х	Х	
Χ	Х	Х	Х	Х	Х	Х	Χ	
Χ	Х	Х	Х	Х	Х	Х	Х]
Χ	Х	Х	Х	Х	Х	Х	Х]

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



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



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

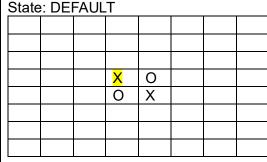
Input: pos = (0, 0), player = 'X'State: DEFAULT 0 0 Χ

Output: FALSE State: same as before X 0 0 Х

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

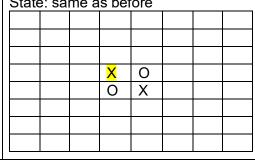
Input: pos = (4, 3), player = 'O'

State: DEFAULT



Output: TRUE

State: same as before



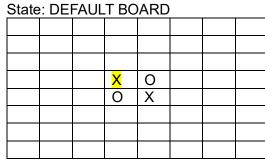
isPlayerAtPos(BoardPosition pos, char player) -[testIsPlayerAtPos_PlayerOneAt_4_4_ReturnsTrue]

Input: pos = (4, 4), player = 'X' State: DEFAULT 0 0 Χ

Output: TRUE State: same as before X 0 0 Χ

 $getScores(void) - [testGetScores_DefaultStartState_Returns2Each]$

Input: N/A



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

State: BOARD IS UNCHANGED

0

0

Χ

getScores(void) - [testGetScores_CustomBoardState_MixedCounts]

Input:				
•				
placeTok	:en('X	', (0,0	1)	
placeTok	en('X	', (1,1)	
placeTok	en('X'	(3,3))	
placeTok	en('X'	, (4,4))	
placeToke	en('O'	, (2,2)	
placeToke	en('O'	, (4,3)	
placeToke	en(ˈOˈ	, (3,4)	
State:	•	-	-	
Y				

place			, (3,4			
Χ						
	Χ					
		0				
			Χ	0		
			0	Χ		

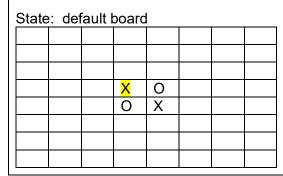
Output: { 'X' = 4, 'O' = 3 }

State: same as before

Χ						
	Χ					
		0				
			Χ	0		
			0	Χ		
		1	1	1		

 $toString(void) - [testToString_DefaultStartState_DisplaysCorrectBoardAndScore]$

Input: N/A9just call toString)



Output: Score: X - 2 | O - 2

State:

