# GameBoard(void) - [testConstructorCreatesEmptyBoard]

Input: None	Output: A new board is created
State: No board exists	State:
checkIfFree(int) – [testCheckIfFreeOnEmpty	Column]
Input: 2	Output: true

Input: 2	Output: true
State: Column 2 is empty	State: No change in the board

## check If Free (int) - [test Check If Free On Partially Filled Column]

Input: 3	: Column 3 has some tokens						е				
State: Column 3 h	nas son	ne tok	ens		State	: No c	hange	e in th	e boa	rd	
	Х							Χ			
	0							0			

		Χ				Χ		

check If Free (int) - [test Check If Free On Full Column]

Input: 4		Output: false
State: Column 4 is	completely full	State: No change in the board
	X	X
	0	0
	X	X
	0	
	X	X
	0	0
	X	X
	0	0
	X	X

 $check Horizontal Win (Board Position, \, char) - [test Check Horiz Win No Win] \\$ 

Input:	new B	oardPo	sition	(2, 3), '	X'		(	Outpu	t: false	)				
State:	ate: No horizontal sequence of tokens					าร	5	State:	No cha	ange				
	-	-	-	-	-	-		-	-	-	-	-	-	-
-	-	-	-	-	-	-		-	-	-	-	-	-	-
-	-	-	Χ	-	-	-		-	-	-	Χ	-	-	-
-	-	-	-	-	-	-		-	-	-	-	-	-	-
-	-	-	-	-	-	-		-	-	-	-	-	-	-
-	-	-	-	-	-	-		-	-	-	-	-	-	-

 $check Horizontal Win (Board Position, \, char) - [test Check Horiz Win With Win] \\$ 

Input	:: new	BoardF	Positio	n(5, 3),	, 'X'		Outp	ut: tru	е				
State	: 4 cor	nsecuti	ive 'X' t	okens	horizo	ntally	State	: No cl	nange				
on ro	w 5						-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	X	Х	Χ	Х	-	-	-
Χ	Х	Х	Х	-	-	-		•	•		•	•	•
•	•	•	•	•	*	-							

 $check Horizontal Win (Board Position, \, char) - [test Check Horiz Win At Left Boundary] \\$ 

Input:	nput: new BoardPosition(0, 0), 'O' State: 4 consecutive O tokens horizontally tarting from column 1.							Outpu	t: true					
State:								State:	No cha	ange				
startin								-	-	-	-	-	-	-
-									-	-	-	-	-	-
-	-	-	-	-	-	-		ı	-	-	-	-	-	-
_	-	-	-	-	-	-		•	-	-	-	•	ı	-
-	-	-	-	-	-	-		-	-	-	-	-	-	-
_	-	-	-	-	-	-		0	0	0	0	-	-	-
0	0 0 0 0													

check Horizontal Win (Board Position, char) - [test Check Horiz Win At Right Boundary]

Input	: new B	oardPo	osition	(0, 6), '	O'		Outpu	t: true					
State	: 4 cons	secutiv	e O tol	kens h	orizont	ally,	State:	No cha	ange.				
endir	ing at column 6.						-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	0	0	0	0
-	-	-	0	0	0	0							
		•		•									

 $check VertWin (Board Position, \, char) - [testCheck VertWin NoWin] \\$ 

checkVertWin(BoardPosition, char) - [testCheckVertWinWithWin]

Input	: new E	BoardF	Positio	n(5, 0),	, 'X'		Outp	ut: true					
		secut	ive X to	kens v	ertical <sup>(</sup>	y in	State	: No ch	ange	1			1
colun	nn u.						-	-	-	-	-	-	-
_	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	Х	-	-	-	-	-	-
X	-	-	-	-	-	-	Х	-	-	-	-	-	-
X	-	-	-	-	-	-	Х	-	-	-	-	-	-
Χ	-	-	-	-	-	-	Х	-	-	-	-	-	-
Х	-	-	-	-	-	-		•	•	•	•	•	•
		•		•	•	•							

checkVertWin(BoardPosition, char) - [testCheckVertWinAtTopBoundary]

Input	kens placed vertically in column 0, with the p at row 0.							Outpu	t: true					
State	-							State:	No ch	ange.				
toker	okens placed vertically in column 0, with the							0	-	-	-	-	-	-
top a	okens placed vertically in column 0, with the op at row 0.							0	-	-	-	-	-	-
О	-	-	-	-	-	-		0	-	-	-	-	-	-
О	-	-	-	-	-	-		0	-	-	-	-	-	-
О	-	-	-	-	-	-		Χ	-	-	-	-	-	-
0	-	-	-	-	-	-		Χ	-	-	-	-	-	-
Х	X													<u> </u>
Х	X													

check VertWin (Board Position, char) - [test Check VertWin At Bottom Boundary]

Input: new BoardPosition(0, 4), 'O' Output: true State: 4 consecutive O tokens and two X State: No change tokens placed vertically in column 0, starting Χ from row O. Χ Χ 0 Χ \_ 0 0 0 0 0 0 0

checkDiagWin(BoardPosition, char) - [testCheckDiagWinNoWin]

check Diag Win (Board Position, char) - [test Check Diag Win With Up Right Diagonal Win]

 $check Diag Win (Board Position, \, char) - [test Check Diag Win With Down Right Diagonal Win] \\$ 

Input: new BoardPosition(5, 0), 'O' Output: true State: 4 consecutive O tokens on a State: No change downward-right diagonal 0 0 \_ О 0 0 0 -\_ 0 0

checkDiagWin(BoardPosition, char) - [testCheckDiagWinWithUpLeftDiagonalWin]

checkDiagWin(BoardPosition, char) - [testCheckDiagWinAtTopRightBoundary]

check Diag Win (Board Position, char) - [test Check Diag Win At Bottom Left Boundary]

checkDiagWin(BoardPosition, char) - [testCheckDiagWinWithOverlapNoWin]

checkTie(void) - [testCheckTie\_FullBoard\_True]

Input:	N/A						Outp	ut: trı	ıe				
State:							State	e:					
Χ	0	Х	0	Х	0	О	Х	0	Х	0	Х	0	0
Χ	0	Χ	0	Χ	0	Χ	Х	0	Х	0	Χ	0	Χ
0	Χ	0	Χ	0	Х	Χ	0	Х	0	Χ	0	Χ	Χ
0	Χ	0	Χ	0	Χ	0	0	Χ	0	Χ	0	Χ	0
Χ	0	Χ	0	Χ	0	0	Х	0	Х	0	Χ	0	0
Χ	0	Χ	0	Χ	0	Χ	Х	0	Х	0	Χ	0	Χ
0	Χ	0	Х	0	Х	Χ	0	Х	0	Χ	0	Χ	Χ
0	Χ	0	Х	0	Х	О	0	Х	0	Х	0	Χ	0
Χ	0	Χ	0	Χ	0	Χ	Х	0	Χ	0	Χ	0	Χ

checkTie(void) - [testCheckTie\_AlmostFull\_False]

Input	: N/A						Outp	ut: fal	se				
State	:						State	:					
	0	Х	0	Х	0	0		0	Х	0	Х	0	0
Χ	0	Χ	0	Χ	0	Χ	Χ	0	Χ	0	Χ	0	Χ
0	Х	0	Χ	0	Χ	Χ	0	Χ	0	Χ	0	Χ	Χ
0	Χ	0	Χ	0	Χ	О	0	Χ	0	Χ	0	Χ	0
Χ	0	Χ	0	Χ	0	О	Χ	0	Χ	0	Χ	0	0
Χ	0	Χ	0	Χ	0	Χ	Χ	0	Χ	0	Χ	0	Χ
0	Χ	0	Χ	0	Χ	Χ	0	Χ	0	Χ	0	Χ	Χ
0	Χ	0	Χ	0	Χ	О	0	Χ	0	Χ	0	Χ	0
Χ	0	Χ	0	Χ	0	Χ	Χ	0	Χ	0	Χ	0	Χ

 $checkTie(void) - [testCheckTie\_FullWin\_False]$ 

Input	: N/A						Outp	ut: fal	se				
State	:						State	:					
Χ	0	Х	0	Х	0	0	Х	0	Х	0	Х	0	0
Χ	0	0	Χ	Х	0	Χ	Χ	0	0	Χ	Χ	0	Χ
0	0	Χ	Χ	0	Χ	Χ	0	0	Χ	Χ	0	Χ	Χ
0	Х	0	Χ	0	Χ	0	0	Х	0	Χ	0	Χ	0
Χ	0	Χ	0	Χ	0	0	Χ	0	Χ	0	Χ	0	0
Χ	0	Χ	0	Χ	0	X	Χ	0	Χ	0	Χ	0	Χ
0	Χ	0	Χ	0	Χ	X	0	Χ	0	Χ	0	Χ	Χ
0	Χ	0	Χ	0	Χ	0	0	Χ	0	Χ	0	Χ	0
Χ	0	Χ	0	Χ	0	Χ	Χ	0	Χ	0	Χ	0	Χ

 $checkTie(void) - [testCheckTie\_Empty\_False]$ 

Input: N	/A			Outp	ut: fal	se			
State:				State	:				
									1
									1

# $whatsAtPos(BoardPosition) - [testWhatsAtPos\_00\_X]$

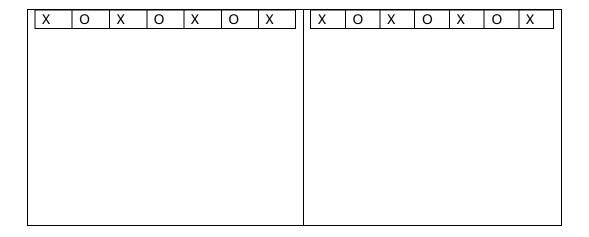
Input: BoardPosition(8,								
State:	nput: BoardPosition(8, 0)  State:							
X			Χ					

## whatsAtPos(BoardPosition) – [testWhatsAtPos\_43\_O]

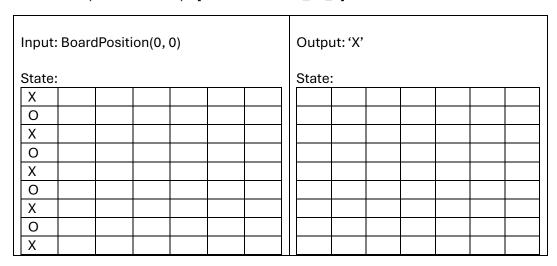
Input: Board	Input: BoardPostion(4, 3)  State:								
State:									
	0						0		
	X						X		
	0						0		
	X						X		
	0						0		

## whatsAtPos(BoardPosition) – [testWhatsAtPos\_06\_Space]

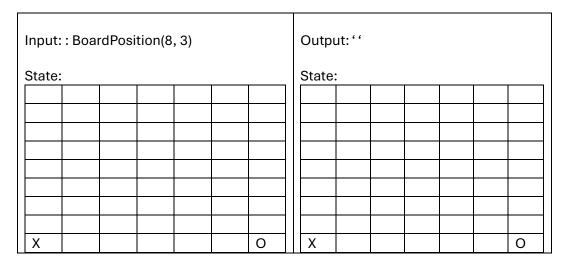
Input	: Boar	dPosit	tion(0,	6)				Outp						
State	State: X O X O X O								:					
Χ	0	Х	0	Х	0			Χ	0	Х	0	Х	0	
Χ	0	Х	0	Х	0	Х		Χ	0	Х	0	Х	0	Χ
0	Х	0	Х	0	Х	Х		0	Х	0	Χ	0	Х	Χ
0	Х	0	Х	0	Х	0		0	Х	0	Χ	0	Х	0
Χ	0	Х	0	Х	0	0		Χ	0	Х	0	Х	0	0
Χ	0	Х	0	Х	0	Х		Χ	0	Х	0	Х	0	Χ
0	Х	0	Х	0	Х	Х		0	Х	0	Х	0	Х	Χ
0	Х	0	Х	0	Х	0		0	Х	0	Х	0	Х	0



 $whatsAtPos(BoardPosition) - [testWhatsAtPos\_80\_X]$ 



whatsAtPos(BoardPosition) – [testWhatsAtPos\_83\_Space]



## isPlayerAtPos(BoardPosition, char) – [testIsPlayerAtPos\_Bottom\_Right]

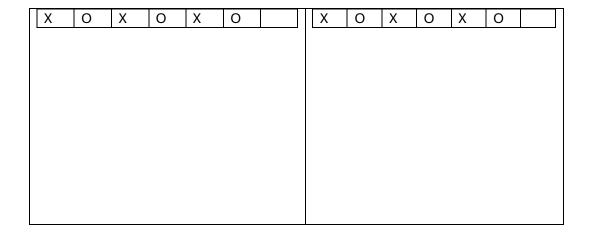
Input:	nput: BoardPosition(8, 6) 'O'						Outp	ut: tru	ie				
State	tate:						State	:					
					0							0	
					0							0	
Χ					Χ		Χ					Χ	
Χ	0	Χ			0		Χ	0	Χ			0	
Χ	0	Χ	0	0	Χ	0	Χ	0	Χ	0	0	Χ	0

 $is Player At Pos (Board Position, char) - [test Is Player At Pos\_Top\_Left] \\$ 

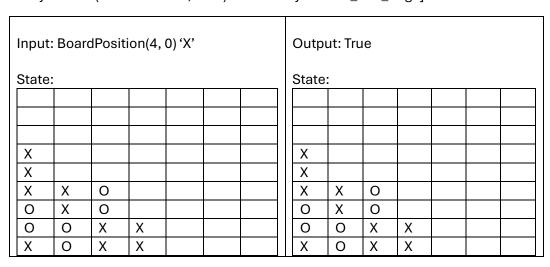
Input	: Board	dPosit	ion(0,	0) 'X'				Outp	ut: fal	se				
State	Input: BoardPosition(0, 0) 'X'  State:								:					
0								0						
0								0						
Х								Χ						
Χ	0							Χ	0					
Χ	0	Χ	0	0	Χ			Χ	0	Χ	0	0	Χ	

 $is Player At Pos (Board Position, char) - [test Is Player At Pos\_Another\_Player] \\$ 

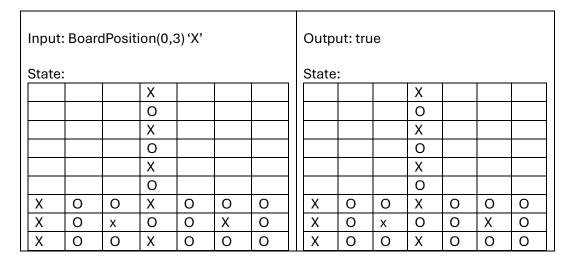
Input	Input: BoardPosition(4, 4) 'O' State:							Outp	ut: tru	ıe			
State	State:												
			0								0		
0	Χ	0	0	0				0	Χ	0	0	0	
Х	0	Χ	Χ	0				Χ	0	Χ	Χ	0	
Χ	0	0	Χ	Х				Χ	0	0	Χ	Χ	
0	Χ	0	Χ	0				0	Χ	0	Χ	0	



isPlayerAtPos(BoardPosition, char) – testIsPlayerAtPos\_Left\_Edge]



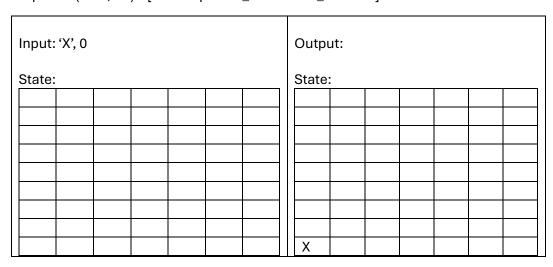
isPlayerAtPos(BoardPosition, char) – [testIsPlayerAtPos\_Top\_Middle]



dropToken(char, int) - [testDropToken\_Full\_Column]

Input	: 'X', 2				Outpi	ut:			
State	:				State	:			
X		0			Χ		0		
0		Χ			0		Χ		
Χ		0			Χ		0		
0		Χ			0		Χ		
Χ		0			Χ		0		
0		Χ			0		Χ		
Χ		0			Χ		0		
0		Χ			0		Χ		
Χ		0			Χ		0		

dropToken(char, int) - [testDropToken\_Mininmum\_Column]

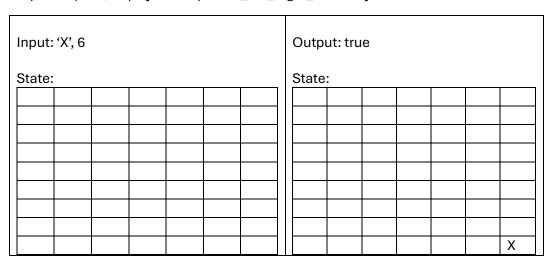


dropToken(char, int) - [testDropToken\_DifferentColumn]

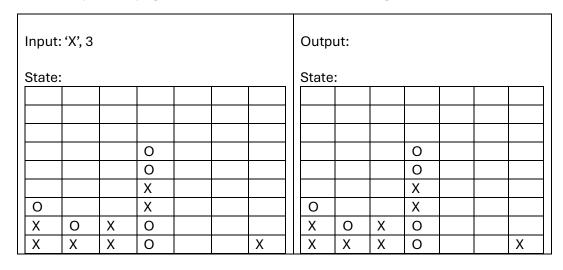
Input:	: 'X', x				Outpu	ut:			
State:	:				State	:			
0					0				

Х				Х		Χ			

dropToken(char, int) - [testDropToken\_Far\_Right\_Column]



dropToken(char, int) - [testDropToken\_Top\_Middle\_Column]



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)

[Jake Barz - uppishdonkey]	Constructor, checkIfFree, checkHorizWin, checkVertWin, checkDiagWin.
George Jubenvill - gjubenv	checkTie, whatsAtPos
[Haagen Williams- haagenwilliams]	dropToken, isPlayerAtPos
[member 4]	