## $GameBoard(void) - [Test\_GameBoard\_Constructor]$

Input:N/A	Output: N/A
State: N/A	State:
checkIfFree(int) – [Test_checkIfFree_col0_	Free]
Input: 0	Output: True
State:	State:
	Board remains unchanged
0	
X	
X	

0	Х	0	0	Х	

checkIfFree(int) - [Test\_checkIfFree\_col0\_notFree]

Input:	0				
State	:				
X					
0					
X					
0					
X					
0					
X					
0		0			
X	0	Х	Х		

Output: False

State:

Board remains unchanged

checkIfFree(int) - [Test\_checkIfFree\_col6\_free]

Input:	6				Output: True
State:					State:
					Board remains unchanged
				0	

checkIfFree(int) - [Test\_checkIfFree\_col6\_notFree]

Inpu	t: 6						Output: False
State	e:						State: Board remain unchanged
						0	
						Х	
						0	
						Х	
					Х	X	
					0	0	
					0	Х	
				Х	Х	0	
Х	0	Х	Х	Х	0	0	
	•	•	•	•	•		

 $checkIfFree(int) - [Test\_checkIfFree\_empty\_col3\_free]$ 

Input: 2	2			Output: True
State:				State: Board remains unchanged

checkHorizontalWin(BoardPosition, char) – [Test\_checkHorizontalWin\_playerO\_NoWin\_pos8\_0]

Input:	: Boar	dPosi	tion(8	,0), "C	)"	Output: False
State	:					State:
						Board remains unchanged
0	Х					

 $check Horizontal Win (Board Position, \, char) - [Test\_check Horizontal Win\_player X\_Win\_pos8\_6]$ 

							Board remains unchanged
			1				
-			<u> </u>				
-			<u> </u>				
0	0	Х	0	0	0		
$\ _{X}$	0	X	X	X	X	X	
		1	<u> </u>				
check	Horizo	ontalV	Vin(Bo	ardPo	osition	n, char)	- [Test_checkHorizontalWin_playerO_Win_pos5_6]
Inpu	t: Boai	rdPos	ition(5	5,6), "0	<b>)</b> "		Output: True
Ctoto							State
State	<b>ð</b> .						State:
							Board remains unchanged
			<u> </u>				
		0	0	0	0	0	
	Х	Х	0	0	Х	0	
X	Х	0	Х	Х	Х	Х	

Output: True

State:

Input: BoardPosition(8,6), "X"

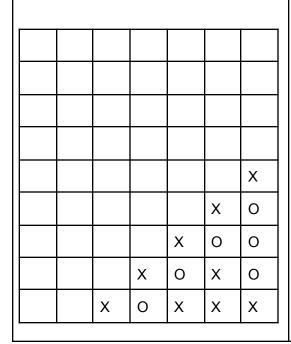
State:

	Х	Х	0	Х	0	0	0
-							•

 $check Horizontal Win (Board Position, \, char) - [Test\_check Horizontal Win\_player X\_NoWin\_pos4\_6]$ 

Input: BoardPosition(4,6), "X"

State:



Output: False

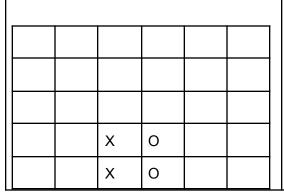
State:

Board remains unchanged

 $check VertWin (Board Position, \, char) - [test\_check VertWin\_Player O\_Win\_pos 3\_3]$ 

Input: BoardPosition(3,3), "O"

State:



Output: True

State:

Board remains unchanged

	Х	0		
	Χ	0		
	0	0	0	
	Х	Х	Х	Х

 $checkVertWin(BoardPosition,\ char)-[test\_checkVertWin\_PlayerX\_Win\_pos4\_2]$ 

Input:	Board	Posit	ion(4,	2), 'X	('	Output: True	
State	• •					State: Board remains unchanged	
		Х					
		Χ					
		Χ					
		Χ					
		Χ					

		0							
		0							
		0							
		0			0				

 $check VertWin (Board Position, \, char) - [test\_check VertWin\_Player O\_NoWin\_pos 3\_1] \\$ 

Χ

Χ

0

 $checkVertWin(BoardPosition, char) - [test\_checkVertWin\_PlayerX\_NoWin\_Interrupted\_pos5\_0]$ 

Input	:Board	dPosit	ion(5,	0), 'X	,		Output: False				
State	<b>:</b> :						State: Board remains unchanged				
[	1					T	1				
X							-				
X											
0							-				
X						<u> </u>	-				
X											
X											
0											
0											
check	DiagW	/in(Bo	ardPo	sition	, char	r) – [te	est_checkDiagWin_PlayerX_DiagWin_pos2_2				
Input	: Boar	dPosi	tion(2	, 2), 'X	ζ'		Output: True				
State			•	,			State: Board remains unchanged				
Х											
0	Х										
0	Х	Х									
0	Х	Х	Х								
ll o	10	0	10	X							

Χ

0

0

Χ

0

Χ

Ο

Χ

checkDiagWin(BoardPosition, char) - [test\_checkDiagWin\_PlayerO\_NoWin\_pos3\_4]

Input: BoardPosition(3, 4), 'O'

State:

			_	_		
		0				
	Х	0	0			
	0	Х	0	0		
	Х	0	0	0	Х	
	Х	0	Х	Х	X	X
	Х	0	0	Х	0	0
	0	Х	0	0	Х	0
	Х	0	0	Х	Х	0
-						

Output: False

State: Board remains unchanged

checkDiagWin(BoardPosition, char) – [test\_checkDiagWin\_PlayerX\_InterruptedWin\_pos4\_0]

Input: BoardPosition(4, 0), 'X'

State:

l						
				0	X	
			Х	X	X	
		0	0	0	Х	
	Х	Х	0	0	0	

Output: False

State: Board remains unchanged

Χ	0	0	Х	0	0	
Х	Х	0	0	Х	Х	
Х	X	0	0	0	0	
0	Х	Х	0	Х	Х	
0	0	Х	Х	0	0	

 $check Diag Win (Board Position, \, char) - [test\_check Diag Win\_Player O\_No Diag Win\_pos 1\_1] \\$ 

						T	
Input	: Boar	<sup>-</sup> dPosi	tion(1	,1) 'O'		Output: False	
State	:					State: Board remains unchanged	
0							
Х	0						
Х	Х	0					
Х	Х	Х					
0	0	Х					
Х	Х	Х	0	0			
0	Х	0	Х	Х			
0	Х	0	0	Х	Х	0	
0	0	Х	0	0	0	0	

checkDiagWin(BoardPosition, char) – [test\_checkDiagWin\_PlayerO\_IncompleteDiagonal\_pos1\_1]

Input		dPos	ition(3	, 3), ')	<b>K</b> '	Output: False State: Board remains unchanged	
State	•						State. Board remains unchanged
Х							
0							
0	0	Х					
Х	0	0	Х				
0	Х	0	Х				
0	0	Х	Х	0	Х	Х	
Х	0	0	Х	0	0	0	
0	0	Х	0	0	Х	Х	
Х	0	0	Х	Х	Х	0	

checkDiagWin(BoardPosition, char) [test\_checkDiagWin\_PlayerO\_oppositeDiagonalWin\_pos4\_1]

				<u> </u>			
			0	Х	0	Х	
		0	0	0	Х	0	
	0	Х	Х	Х	Х	0	
0	0	Х	0	Х	0	Х	
Х	Х	0	Х	0	Х	0	
Х	0	0	0	Х	0	0	
Х	0	0	0	Х	0	Х	
	heck[	DiagW	in_Pla		_NoW	Output: False State: Board remains unchanged	
							Coard Formanic and San gen
X			\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \				
0	0	X	X				
0	0	0	X				
0	Х	Х	Х				
Ιv	v			V			
х О	X	0	0	Х			

Output: True

State: Board remains unchanged

Input: BoardPosition(4, 1), 'O'

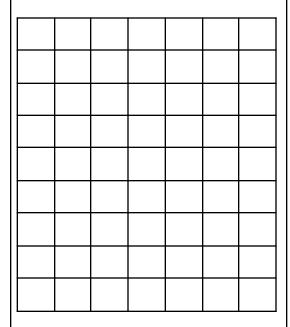
State:

0	0	X	0	Х	X	

checkTie(void) - [test\_checkTie\_empty\_false]

Input: N/A

State:



Output: False

State:

Board remains unchanged

 $checkTie(void) - [test\_checkTie\_full\_true]$ 

Input: N/A

State:

					-		_
X		0	X	0	Х	0	Х
X		0	Х	0	Х	0	Х
С	)	Х	0	Х	0	Х	0
X		0	Х	Х	Х	0	Х
С	)	0	0	0	Х	0	Х
С	)	Х	0	Х	Х	0	Х

Output: True

State:

Board remains unchanged

0 0 x 0 0 x 0
x o o x x x x

 $checkTie(void) - [test\_checkTie\_halfFull\_false]$ 

Input	: N/A						Output: False
State	:					State: Board remains unchanged	
						, and the second	
Х	0	0	0	Х	Х	0	
0	Х	Х	Х	0	Х	0	
0	Х	Х	0	Х	0	0	
Х	0	Х	0	0	Х		

 $checkTie(void) - [test\_checkTie\_PlayerXWin\_false]$ 

Input:	N/A			Output: False	
State:				State: Board remains the same	

		X	X	X	Х	Х			
L									

 $whatsAtPos(BoardPosition) - [Test\_whatsAtPos\_markerX\_pos8\_2]$ 

Input: Boar	<sup>-</sup> dPosi	tion(8	, 2)		Output: 'X'	
State:					State:	
					Board is unchanged	
	Х					
•						

whatsAtPos(BoardPosition) – [test\_whatsAtPos\_empty\_pos4\_1]

Input:	Boar	dPosit	tion(4,	1)		Output: ' '	
State	: :					State: Board is unchanged	
whats/	AtPos(	(Board	dPosit	ion) –	[Test	_whats	sAtPos_markerO_pos3_4]
Input:	Boar	dPosit	tion(3,	4)			Output: 'O'
State							State: Board is unchanged
				0			
				Х			
				0			
				Х			
				0			
				Χ			

vhats	AtPos	(Board	dPosit	ion) –	[Test	_whats	sAtPos_markerO_pos8_6]
Input State		dPosi	tion(8,	.6)		Output: 'O' State: Board is unchanged	
						0	Board to amortalingou
/hats/	AtPos	(Board	dPosit	ion) –	[Test	_whats	sAtPos_markerO_pos0_0]
Input	: Boar	dPosi	tion(0	,0)			Output: 'X'
State	:					<u> </u>	State: Board is unchanged
Х							
0							
Х							
0							
Х							
0							
Х							

0									
Х									

isPlayerAtPos(BoardPosition, char) – [Test\_isPlayerAtPos\_playerO\_pos8\_0]

Input: BoardPosition(8,0), 'O'
State:

State:

Board remains unchanged

X
Output:
False
State:
Board remains unchanged

isPlayerAtPos(BoardPosition, char) – [Test\_isPlayerAtPos\_playerX\_pos1\_2]]

Input:	dPosi	tion(1,	2), 'X'	,	Output: True State:	
						Board remains unchanged
	Х					
	0					
	X					
	0					

		Х			0				
		0			Х	0			
		X			Х	0			
		0			Х	0			

isPlayerAtPos(BoardPosition, char) – [Test\_isPlayerAtPos\_playerX\_pos0\_0]

Input: State:	dPosit	tion(0,	0), 'X	,	Output: False State:	
0						Board remains unchanged
0						
Х						
0						
Х						
0						
Х						
0						
Х						

 $is Player At Pos (Board Position, char) - [Test\_is Player At Pos\_player O\_pos 8\_6] \\$ 

Input:	BoardPo	sition(8	,6), 'O	)'	Output: False	
State:						State:
						Board remains unchanged
					X	
Input:	BoardPo	sition(2	,3), 'X	,		Output: False
						State: Board remains unchanged
1						
	Х					
	0					
	0					
	Х					
	0	X				

Input: 'X', 0									Output: None							
State:								State	<b>)</b> :							
			<u> </u> 													
				Х		Х						Х		Х		
	0			0		0			0			0		0		
	Х		Х	0	Х	0			Х		Х	0	х	0		
	Х	Х	0	0	О	Х		Х	Х	Х	0	0	0	Х		
ropToken(char, int) – [Test_dropToken_pla nput: 'O', 6 State:									_colui		filled]					
	Х	0	0	Х	0			0	Х	0	0	Х	0	0		
0						1		l	1		1					
О Х	Х	0	Х	Х	Х	Х		X	X	0	Х	Х	Х	Х		
	+	о х	X O	х О	X	Х О		0	х О	O X	х О	х О	X	х О		

Χ

0

О

Ο

Χ

0

X

0

0

0

Χ

0

Χ

0

Χ

Χ

0

Χ

Χ

0

Χ

0

Χ

О

Χ

0

0

Χ

0

Х

Х

0

О

0

Χ

0

Χ

0

Χ

Χ

0

Χ

Х	Х	0	0	0	0	Х	Х	Х	0	0	0	0	Х
0	0	Х	0	0	Х	0	0	0	Х	0	0	Х	0
							-						

 $dropToken(char,\,int) - [Test\_dropToken\_playerX\_column3\_empty]$ 

Input: 'X', 3 State:	Output: None State:							
	X							

dropToken(char, int) – [Test\_dropToken\_playerO\_column4\_Vwin]

Input: 'O', 4									Output: None							
State:								State:								

									0		
			0						0		
			0						0		
			0		Х				0		Х
Х	Х		0	Х	Х	Х	Х		0	Х	Х

dropToken(char, int) – [Test\_dropToken\_playerX\_column3\_Hwin]

nput	:: 'X', :	3			Outp	Output: None							
State	<b>)</b> :			State	<b>e</b> :								
							_						
							_						
					0							0	
Χ				Х	0	X					Х	0	
X	Х	X	X	X	X	x	X	X	X	X	X	X	

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)

[member 1] Cooper Taylor	Wrote all GameBoard, checkIfFree, checkHorizontalWin, and test_checkVertWin_PlayerO_Win_pos3_3
[member 2] Michael Ellis	Wrote three checkVertWin test cases and wrote all seven checkDiagWin test cases.
[member 3] Ryan Chen	Wrote all test cases for checkTie and whatsAtPos
[member 4] - Adam Niemczura	Wrote all the test cases for dropToken and isPlayerAtPos