GameBoard(void) – [testConstructorCreatesEmptyBoard]

Input: None

State: No board exists

Output: A new board is created

State:

checkIfFree(int) – [testCheckIfFreeOnEmptyColumn]

Input: 2

State: Column 2 is empty

Output: true

State: No change in the board

checkIfFree(int) – [testCheckIfFreeOnPartiallyFilledColumn]

Input: 3

State: Column 3 has some tokens

			X			
			O			

Output: true

State: No change in the board

			X			
			O			

			X			

			X			

checkIfFree(int) – [testCheckIfFreeOnFullColumn]

Input: 4

State: Column 4 is completely full

				X		
				O		
				X		
				O		
				X		
				O		
				X		
				O		
				X		

Output: false

State: No change in the board

				X		
				O		
				X		
				O		
				X		
				O		
				X		
				O		
				X		

checkHorizontalWin(BoardPosition, char) – [testCheckHorizWinNoWin]

Input: new BoardPosition(2, 3), ‘X’

State: No horizontal sequence of tokens

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	X	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Output: false

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	X	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

checkHorizontalWin(BoardPosition, char) – [testCheckHorizWinWithWin]

Input: new BoardPosition(5, 3), ‘X’

State: 4 consecutive ‘X’ tokens horizontally on row 5

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
X	X	X	X	-	-	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
X	X	X	X	-	-	-

checkHorizontalWin(BoardPosition, char) – [testCheckHorizWinAtLeftBoundary]

Input: new BoardPosition(0, 0), ‘O’

State: 4 consecutive O tokens horizontally starting from column 1.

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
O	O	O	O	-	-	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
O	O	O	O	-	-	-

checkHorizontalWin(BoardPosition, char) – [testCheckHorizWinAtRightBoundary]

Input: new BoardPosition(0, 6), ‘O’

State: 4 consecutive O tokens horizontally, ending at column 6.

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	O	O	O	O

Output: true

State: No change.

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	O	O	O	O

checkVertWin(BoardPosition, char) – [testCheckVertWinNoWin]

Input: new BoardPosition(3, 4), ‘X’

State: No vertical sequence of 5 tokens

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	X	-	-
-	-	-	-	X	-	-
-	-	-	-	O	-	-
-	-	-	-	X	-	-

Output: false

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	X	-	-
-	-	-	-	X	-	-
-	-	-	-	O	-	-
-	-	-	-	X	-	-

checkVertWin(BoardPosition, char) – [testCheckVertWinWithWin]

Input: new BoardPosition(5, 0), ‘X’

State: 4 consecutive X tokens vertically in column 0.

-	-	-	-	-	-	-
-	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-

checkVertWin(BoardPosition, char) – [testCheckVertWinAtTopBoundary]

Input: new BoardPosition(5, 2), ‘O’

State: 4 consecutive O tokens and two X tokens placed vertically in column 0, with the top at row 0.

O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-

Output: true

State: No change.

O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
X	-	-	-	-	-	-
X	-	-	-	-	-	-

checkVertWin(BoardPosition, char) – [testCheckVertWinAtBottomBoundary]

Input: new BoardPosition(0, 4), ‘O’

State: 4 consecutive O tokens and two X tokens placed vertically in column 0, starting from row O.

X	-	-	-	-	-	-
X	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-

Output: true

State: No change

X	-	-	-	-	-	-
X	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-
O	-	-	-	-	-	-

checkDiagWin(BoardPosition, char) – [testCheckDiagWinNoWin]

Input: new BoardPosition(2, 2), ‘X’

State: 3 X tokens and one O token at a diagonal

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	X
-	-	-	-	-	O	-
-	-	-	-	X	-	-
-	-	-	X	-	-	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	X
-	-	-	-	-	O	-
-	-	-	-	X	-	-
-	-	-	X	-	-	-

checkDiagWin(BoardPosition, char) – [testCheckDiagWinWithUpRightDiagonalWin]

Input: new BoardPosition(0, 0), ‘X’

State: 4 consecutive X tokens on an upward-right diagonal

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	X
-	-	-	-	-	X	-
-	-	-	-	X	-	-
-	-	-	X	-	-	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	X
-	-	-	-	-	X	-
-	-	-	-	X	-	-
-	-	-	X	-	-	-

checkDiagWin(BoardPosition, char) – [testCheckDiagWinWithDownRightDiagonalWin]

Input: new BoardPosition(5, 0), ‘O’

State: 4 consecutive O tokens on a downward-right diagonal

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	O	-	-	-	-
-	-	-	O	-	-	-
-	-	-	-	O	-	-
-	-	-	-	-	O	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	O	-	-	-
-	-	-	-	O	-	-
-	-	-	-	-	O	-
-	-	-	-	-	-	O

checkDiagWin(BoardPosition, char) – [testCheckDiagWinWithUpLeftDiagonalWin]

Input: new BoardPosition(3, 3), ‘O’

State: 4 consecutive O tokens on an upward-left diagonal

-	O	-	-	-	-	-
-	-	O	-	-	-	-
-	-	-	O	-	-	-
-	-	-	-	O	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Output: true

State: No change

-	O	-	-	-	-	-
-	-	O	-	-	-	-
-	-	-	O	-	-	-
-	-	-	-	O	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

checkDiagWin(BoardPosition, char) – [testCheckDiagWinAtTopRightBoundary]

Input: new BoardPosition(0, 6), ‘X’

State: 4 consecutive X tokens on an upward-right diagonal in the top-right corner

-	-	-	-	-	-	X
-	-	-	-	-	X	-
-	-	-	-	X	-	-
-	-	-	X	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

Output: true

State: No change

-	-	-	-	-	-	X
-	-	-	-	-	X	-
-	-	-	-	X	-	-
-	-	-	X	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-

checkDiagWin(BoardPosition, char) – [testCheckDiagWinAtBottomLeftBoundary]

Input: new BoardPosition(5, 0), ‘X’

State: 4 consecutive X tokens on a downward-left diagonal in the bottom-left corner

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	X	-	-	-
-	-	X	-	-	-	-
-	X	-	-	-	-	-
X	-	-	-	-	-	-

Output: true

State: No change

-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	X	-	-	-
-	-	X	-	-	-	-
-	X	-	-	-	-	-
X	-	-	-	-	-	-

checkDiagWin(BoardPosition, char) – [testCheckDiagWinWithOverlapNoWin]

Input: new BoardPosition(4, 4), ‘X’

State: Mixed diagonal tokens, no sequence of 4.

-	-	-	-	-	-	-
-	O	-	-	-	-	-
-	-	O	-	-	-	-
-	-	-	X	-	-	-
-	-	-	-	X	-	-
-	-	-	-	-	-	-

Output: false

State: No change.

-	-	-	-	-	-	-
-	O	-	-	-	-	-
-	-	O	-	-	-	-
-	-	-	X	-	-	-
-	-	-	-	X	-	-
-	-	-	-	-	-	-

checkTie(void) – [testCheckTie_FullBoard_True]

Input: N/A

State:

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

Output: true

State:

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

checkTie(void) – [testCheckTie_AlmostFull_False]

Input: N/A

State:

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

Output: false

State:

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

checkTie(void) – [testCheckTie_FullWin_False]

Input: N/A

State:

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

Output: false

State:

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

checkTie(void) – [testCheckTie_Empty_False]

Input: N/A

State:

Output: false

State:

whatsAtPos(BoardPosition) – [testWhatsAtPos_00_X]

Input: BoardPosition(8, 0)

State:

X						

Output: ‘X’

State:

X						

whatsAtPos(BoardPosition) – [testWhatsAtPos_43_O]

Input: BoardPostion(4, 3)

State:

			O			
			X			
			O			
			X			
			O			

Output: ‘O’

State:

			O			
			X			
			O			
			X			
			O			

whatsAtPos(BoardPosition) – [testWhatsAtPos_06_Space]

Input: BoardPosition(0, 6)

State:

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

Output: ‘ ‘

State:

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

X	O	X	O	X	O	X

X	O	X	O	X	O	X

whatsAtPos(BoardPosition) – [testWhatsAtPos_80_X]

Input: BoardPosition(0, 0)

State:

X						
O						
X						
O						
X						
O						
X						
O						
X						

Output: ‘X’

State:

whatsAtPos(BoardPosition) – [testWhatsAtPos_83_Space]

Input: : BoardPosition(8, 3)

State:

X						O

Output: ‘ ‘

State:

X						O

isPlayerAtPos(BoardPosition, char) – [testIsPlayerAtPos_Bottom_Right]

Input: BoardPosition(8, 6) ‘O’

State:

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

Output: true

State:

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

isPlayerAtPos(BoardPosition, char) – [testIsPlayerAtPos_Top_Left]

Input: BoardPosition(0, 0) ‘X’

State:

O						
O						
X						
X	O					
X	O	X	O	O	X	

Output: false

State:

O						
O						
X						
X	O					
X	O	X	O	O	X	

isPlayerAtPos(BoardPosition, char) – [testIsPlayerAtPos_Another_Player]

Input: BoardPosition(4, 4) ‘O’

State:

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

Output: true

State:

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

X	O	X	O	X	O	

X	O	X	O	X	O	

isPlayerAtPos(BoardPosition, char) – testIsPlayerAtPos_Left_Edge]

Input: BoardPosition(4, 0) ‘X’

State:

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

Output: True

State:

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

isPlayerAtPos(BoardPosition, char) – [testIsPlayerAtPos_Top_Middle]

Input: BoardPosition(0,3) ‘X’

State:

			X			
			O			
			X			
			O			
			X			
			O			
X	O	O	X	O	O	O
X	O	x	O	O	X	O
X	O	O	X	O	O	O

Output: true

State:

			X			
			O			
			X			
			O			
			X			
			O			
X	O	O	X	O	O	O
X	O	x	O	O	X	O
X	O	O	X	O	O	O

dropToken(char, int) – [testDropToken_Full_Column]

Input: ‘X’, 2

State:

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

Output:

State:

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

dropToken(char, int) – [testDropToken_Mininmum_Column]

Input: ‘X’, 0

State:

Output:

State:

X						

dropToken(char, int) – [testDropToken_DifferentColumn]

Input: ‘X’, x

State:

O						

Output:

State:

O						

X						

X		X				

dropToken(char, int) – [testDropToken_Far_Right_Column]

Input: ‘X’, 6

State:

Output: true

State:

						X

dropToken(char, int) – [testDropToken_Top_Middle_Column]

Input: ‘X’, 3

State:

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

Output:

State:

			O			
			O			
			X			
O			X			
X	O	X	O			
X	X	X	O			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)

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