GameBoard(void) – [Test_GameBoard_Constructor]

<table>
<tr>
<td>Input:N/A

State: N/A</td>
<td>Output: N/A

State:

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td>
</tr>
</table>

checkIfFree(int) – [Test_checkIfFree_col0_Free]

<table>
<tr>
<td>Input: 0

State:

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td>
<td>Output: True

State:

Board remains unchanged</td>
</tr>
</table>

O	X	O	O	X		

checkIfFree(int) – [Test_checkIfFree_col0_notFree]

Input: 0

State:

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

Output: False

State:

Board remains unchanged

checkIfFree(int) – [Test_checkIfFree_col6_free]

Input: 6

State:

						O

Output: True

State:

Board remains unchanged

						O
					X	O
	O	X	O	X	O	X

checkIfFree(int) – [Test_checkIfFree_col6_notFree]

Input: 6

State:

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

Output: False

State:
Board remain unchanged

checkIfFree(int) – [Test_checkIfFree_empty_col3_free]

Input: 2

State:

Output: True

State:
Board remains unchanged

checkHorizontalWin(BoardPosition, char) – [Test_checkHorizontalWin_playerO_NoWin_pos8_0]

Input: BoardPosition(8,0), “O” State:	Output: False State: Board remains unchanged

O	X					

checkHorizontalWin(BoardPosition, char) – [Test_checkHorizontalWin_playerX_Win_pos8_6]

Input: BoardPosition(8,6), “X”

State:

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

Output: True

State:

Board remains unchanged

checkHorizontalWin(BoardPosition, char) – [Test_checkHorizontalWin_playerO_Win_pos5_6]

Input: BoardPosition(5,6), “O”

State:

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

Output: True

State:

Board remains unchanged

X	X	O	X	O	O	O

checkHorizontalWin(BoardPosition, char) – [Test_checkHorizontalWin_playerX_NoWin_pos4_6]

Input: BoardPosition(4,6), “X”

State:

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

Output: False

State:

Board remains unchanged

checkVertWin(BoardPosition, char) – [test_checkVertWin_PlayerO_Win_pos3_3]

Input: BoardPosition(3,3), “O”

State:

		X	O		
		X	O		

Output: True

State:

Board remains unchanged

		X	O		
		X	O		
		O	O	O	
		X	X	X	X

checkVertWin(BoardPosition, char) – [test_checkVertWin_PlayerX_Win_pos4_2]

Input: BoardPosition(4, 2), ‘X’

State:

		X				
		X				
		X				
		X				
		X				

Output: True

State: Board remains unchanged

		O				
		O				
		O				
		O			O	

checkVertWin(BoardPosition, char) – [test_checkVertWin_PlayerO_NoWin_pos3_1]

Input: BoardPosition(3, 1), ‘O’

State:

	O					
	O					
	X					
	X					
	X					
	O					
	O		X		X	

Output: False

State: Board remains unchanged

checkVertWin(BoardPosition, char) – [test_checkVertWin_PlayerX_NoWin_Interrupted_pos5_0]

Input:BoardPosition(5, 0), ‘X’

State:

X						
X						
X						
O						
X						
X						
X						
O						
O						

Output: False

State: Board remains unchanged

checkDiagWin(BoardPosition, char) – [test_checkDiagWin_PlayerX_DiagWin_pos2_2]

Input: BoardPosition(2, 2), ‘X’

State:

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

Output: True

State: Board remains unchanged

X	X	O	X	O		
O	X	O	X	O		

checkDiagWin(BoardPosition, char) – [test_checkDiagWin_PlayerO_NoWin_pos3_4]

Input: BoardPosition(3, 4), ‘O’

State:

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

Output: False

State: Board remains unchanged

checkDiagWin(BoardPosition, char) – [test_checkDiagWin_PlayerX_InterruptedWin_pos4_0]

Input: BoardPosition(4, 0), ‘X’

State:

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

Output: False

State: Board remains unchanged

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

checkDiagWin(BoardPosition, char) – [test_checkDiagWin_PlayerO_NoDiagWin_pos1_1]

Input: BoardPosition(1,1) ‘O’

State:

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

Output: False

State: Board remains unchanged

checkDiagWin(BoardPosition, char) –
[test_checkDiagWin_PlayerO_IncompleteDiagonal_pos1_1]

<table>
<tr>
<td>Input: BoardPosition(3, 3), ‘X’

State:

<table>
<tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>O</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>X</td><td>O</td><td>O</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td>O</td><td>X</td><td>O</td><td>X</td><td></td><td></td><td></td></tr>
<tr><td>O</td><td>O</td><td>X</td><td>X</td><td>O</td><td>X</td><td>X</td></tr>
<tr><td>X</td><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td>O</td><td>O</td><td>X</td><td>O</td><td>O</td><td>X</td><td>X</td></tr>
<tr><td>X</td><td>O</td><td>O</td><td>X</td><td>X</td><td>X</td><td>O</td></tr>
</table>
</td>
<td>Output: False

State: Board remains unchanged</td>
</tr>
</table>

checkDiagWin(BoardPosition, char) –
[test_checkDiagWin_PlayerO_oppositeDiagonalWin_pos4_1]

Input: BoardPosition(4, 1), ‘O’

State:

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

Output: True

State: Board remains unchanged

checkDiagWin(BoardPosition, char) –
[test_checkDiagWin_PlayerX_NoWin_ScatteredTokens_pos3_3]

Input: BoardPosition(3, 3), ‘X’

State:

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

Output: False

State: Board remains unchanged

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

Output: True

State:
Board remains unchanged

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

checkTie(void) – [test_checkTie_halfFull_false]

Input: N/A

State:

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

Output: False

State:
Board remains unchanged

checkTie(void) – [test_checkTie_PlayerXWin_false]

Input: N/A

State:

Output: False

State:
Board remains the same

	X	X	X	X	X	

whatsAtPos(BoardPosition) – [Test_whatsAtPos_markerX_pos8_2]

Input: BoardPosition(8, 2)

State:

		X				

Output: ‘X’

State:

Board is unchanged

whatsAtPos(BoardPosition) – [test_whatsAtPos_empty_pos4_1]

Input: BoardPosition(4, 1)

State:

Output: ‘ ‘

State:
Board is unchanged

whatsAtPos(BoardPosition) – [Test_whatsAtPos_markerO_pos3_4]

Input: BoardPosition(3, 4)

State:

				O		
				X		
				O		
				X		
				O		
				X		

Output: ‘O’

State:
Board is unchanged

whatsAtPos(BoardPosition) – [Test_whatsAtPos_markerO_pos8_6]

Input: BoardPosition(8,6)

State:

						O

Output: ‘O’

State:
Board is unchanged

whatsAtPos(BoardPosition) – [Test_whatsAtPos_markerO_pos0_0]

Input: BoardPosition(0,0)

State:

X						
O						
X						
O						
X						
O						
X						

Output: ‘X’

State:
Board is unchanged

O						
X						

isPlayerAtPos(BoardPosition, char) – [Test_isPlayerAtPos_playerO_pos8_0]

Input: BoardPosition(8,0), ‘O’

State:

X				O		

Output:
False

State:
Board remains unchanged

isPlayerAtPos(BoardPosition, char) – [Test_isPlayerAtPos_playerX_pos1_2]]

Input: BoardPosition(1,2), ‘X’

State:

		X				
		O				
		X				
		O				

Output:
True

State:
Board remains unchanged

		X			O	
		O			X	O
		X			X	O
		O			X	O

isPlayerAtPos(BoardPosition, char) – [Test_isPlayerAtPos_playerX_pos0_0]

Input: BoardPosition(0,0), ‘X’

State:

O						
O						
X						
O						
X						
O						
X						
O						
X						

Output:
False

State:
Board remains unchanged

isPlayerAtPos(BoardPosition, char) – [Test_isPlayerAtPos_playerO_pos8_6]

Input: BoardPosition(8,6), ‘O’

State:

						X

Output:
False

State:
Board remains unchanged

isPlayerAtPos(BoardPosition, char) – [Test_isPlayerAtPos_playerX_pos2_3]

Input: BoardPosition(2,3), ‘X’

State:

		X				
		O				
		O				
		X				
		O	X			

Output:
False

State:
Board remains unchanged

dropToken(char, int) – [Test_dropToken_playerX_column0_regular]

Input: ‘X’, 0

State:

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

Output: None

State:

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

dropToken(char, int) – [Test_dropToken_playerO_column6_filled]

Input: ‘O’, 6

State:

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

Output: None

State:

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

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

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

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

State:

Output: None

State:

				O		
				O		
				O		X
X	X			O	X	X

				O		
				O		
				O		
				O		X
X	X			O	X	X

dropToken(char, int) – [Test_dropToken_playerX_column3_Hwin]

Input: ‘X’, 3

State:

						O
X					X	O
X	X	X		X	X	X

Output: None

State:

						O
X					X	O
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