

Commit group/names	Commit number	Description
Test1: Default Board (3x3)	2cba1d7	<pre> @Test public void test1(){ String[][] result = {{ "_", "_", "_"}, {"_", "_", "_"}, {"_", "_", "_"}; assertTrue(Arrays.deepEquals(result, <u>tic.default Board()</u>)); } public static String[][] default_Board(){ String[][] board = new String[3][3]; for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) { board[i][j] = "_"; } } return board; } </pre>
Test2: AssertEquals (if a box is empty)	a916869	<pre> @Test void test2() { Tic tic = new Tic(row:3,col:3); assertEquals("_", tic.board[1][1]); } </pre>

Test3: PlaceO (places o in an specific case)	f0dd154	<pre> @Test public void test3(){ Tic board=new Tic(row:3,col:3); board.placeO(row:1,col:1); assertEquals(board.board[1][1],"o"); } public void placeO(int row,int col){ if(row>this.rows col>this.cols){ return; } this.board[row][col] ="o"; } </pre>
Test4: Equals (compares boards)	ec3aa92	<pre> @Test void test4() { Tic tic1 = new Tic(row:3, col:3); Tic tic2 = new Tic(row:3,col:3); assertTrue(Tic.Equals(tic1, tic2)); } public static boolean Equals(Tic tic1, Tic tic2){ if(tic1.rows != tic2.rows tic1.cols != tic2.cols){ return false; } for(int i=0; i<tic1.rows; i++){ for(int j=0; j<tic1.cols; j++){ if(tic1.board[i][j]!=tic2.board[i][j]){ return false; } } } return true; } </pre>
Test5: BoardSize (number vs board size)	89172c8	<pre> @Test public void test5() { assertEquals(4, Tic.boardSize(indexI:2,indexJ:2)); } public static int boardSize(int indexI,int indexJ){ return indexI*indexJ; } </pre>

Test6: FullRow (if a row contains same values)	5aa7ed9	<pre> @Test public void test6() { Tic tic = new Tic(row:3, col:3); for(int i=0; i<3; i++){ tic.board[0][i] = "o"; } assertTrue(tic.fullRow(row:2)); } public boolean fullRow(int row) { String value = this.board[row][0]; for(int i=0; i<this.cols; i++){ if(this.board[row][i]!=value){ return false; } } return true; } </pre>
Test7: EmptyBoa rd (is board is empty)	2a40e68	<pre> @Test public void test7() { String[][] result = {"_","_","_"},{"_","_","_"}; assertTrue(Arrays.deepEquals(result,tic.emptyboard(indexI:2,indexJ:3))); } public static String[][] emptyboard(int indexI, int indexJ) { String[][] board = new String[indexI][indexJ]; for (int i = 0; i < indexI; i++) { for (int j = 0; j < indexJ; j++) { board[i][j] = "_"; } } return board; } </pre>

Test8: fullCol (if column contains same values)	0fc7632	<pre> @Test public void test8() { Tic tic = new Tic(row:3, col:3); for(int i=0; i<3; i++){ tic.board[i][0] = "X"; } assertTrue(tic.fullCol(col:1)); } public boolean fullCol(int col) { String value = this.board[0][col]; for(int i=0; i<this.rows; i++){ if(this.board[i][col]!=value){ return false; } } return true; } </pre>
Test9: PlaceX (places x in an specific case)	428fbb9	<pre> @Test public void test9(){ Tic board=new Tic(row:3,col:3); board.placeX(row:1,col:1); assertEquals(board.board[1][1],"x"); } public void placeX(int row,int col){ if(row>this.rows col>this.cols){ return; } this.board[row][col] ="x"; } </pre>
Test10: FullCol+pl aceX (merging methods)	11563c5	<pre> @Test public void test10() { Tic tic = new Tic(row:3, col:3); for(int i=0; i<3; i++){ tic.placeX(i,col:0); } assertTrue(tic.fullCol(col:1)); } </pre>

		<pre> public boolean fullCol(int col) { String value = this.board[0][col]; for(int i=0; i<this.rows; i++){ if(this.board[i][col]!=value){ return false; } } return true; } </pre>
--	--	--

```

+-- JUnit Jupiter [OK]
| '-- TicTest [OK]
|   +-- test10() [X] Index 2 out of bounds for length 2
|   +-- test1() [X] expected: <true> but was: <false>
|   +-- test2() [X] expected: <> but was: <>
|   +-- test3() [X] expected: <o> but was: <x>
|   +-- test4() [X] expected: <true> but was: <false>
|   +-- test5() [X] expected: <2> but was: <4>
|   +-- test6() [X] Index -1 out of bounds for length 3
|   +-- test7() [X] expected: <true> but was: <false>
|   +-- test8() [X] Index 2 out of bounds for length 2
|   '-- test9() [X] expected: <x> but was: <o>
+-- JUnit Vintage [OK]

```

```

+-- JUnit Jupiter [OK]
| '-- TicTest [OK]
|   +-- test10() [OK]
|   +-- test1() [OK]
|   +-- test2() [OK]
|   +-- test3() [OK]
|   +-- test4() [OK]
|   +-- test5() [OK]
|   +-- test6() [OK]
|   +-- test7() [OK]
|   +-- test8() [OK]
|   '-- test9() [OK]
+-- JUnit Vintage [OK]

```