TEST CASE CHART FOR BOARD CLASS

Description of Test	**Test Case**	**Expected Result**
Testing that stones at a given pitPosition are correct after the board has been initialized.	pitPosition = 1	4
Testing that an exception is thrown when an invalid pitPosition is entered to fetch a given number of stones from	pitPosition = 0;	PitNotFoundException thrown
Testing that an exception is thrown when an invalid pitPosition is entered to fetch number of stones from	pitPosition = 13	PitNotFoundException thrown
Testing that a side of the board is empty (without the board being initialised)	pitPosition = 1;	True
Testing that a side of the board is NOT empty (after the board is initialised)	pitPosition = 1;	False
Testing that a side of the board is NOT empty (after the board is initialised)	pitPosition = 7;	False
Testing that the board resets correctly	Assert that after the board has been reset, there are 4 stones at each pit and 0 stones at each store	4 (at each pit), 0(at each store)
Testing that players are registered correctly	Assert that the owners of the store are equal to the right players Assert that the stores of each player is not null	PlayerOne (for board.getStores.get(0).getOwner()) PlayerTwo (for board.getStores.get(1).getOwner()))

Testing that stones were captured correctly after board is initialised	stoppingPoint = 2; check that stones captured are >=0	True, stones were captured
Testing that no stones were distributed when the board is empty	startingPoint = 1	0 (for number of stones distributed) False (because no stones were distributed)
Testing that stones were distributed correctly after the board is initialized	startingPoint = 1	True, stones were distributed The stones at the next 4 pits should be equals 5
Testing that stones were distributed correctly after the board is initialized	startingPoint = 4	True, stones were distributed 5 (no of stones at next pit) 1 (stones at the player's store) 0 (the stones at the starting point should be 0)
Testing that stones were distributed correctly after the board is initialized	startingPoint = 8	True, stones were distributed 0 (the stones at the starting point should be 0) 5 (no of stones at next pit)
Testing that when an invalid pit is entered, the distributeStones method throws PitNotFoundException	startingPoint = 0 startingpoint = 13	PitNotFoundException thrown