Game

Equivalence Partition

Partition ID	Input Variable	Valid Partition	Invalid Partition
1	Object g	()	
2	Object g	(board)	
3	Object g	(board, state)	
4	Object g	(board, check, type)	
5	Object g		Invalid Parameter data types that are against the valid partitions shown above for g
6	board	new Piece(type, color) type must be the chess pieces' names color must be white or black	
7	board		If the arguments in the parameters for Piece are misplaced, than it will result in a syntax error.
8	function g.swapColor()	The function can swap the color of the chess piece	
9	Function g.swapColor()	The function can swap the color of the chess piece as many times as possible if needed.	
10	Function g.swapColor()		Since the initial color of the piece is black, if the assertEquals were to be white, than an error will occur.
11	g.toString()	As long as the requirements and rules are followed for the Game object, than converting it using the	

		.toString() method shouldn't have problems.	
12	g.toString()		If the rules are not followed for implementing the Game object correctly, errors will occur.

Test Cases

Test ID	Test Inputs	Expected Output	Partition ID Covered
1	Game g = new Game();	Returns an empty board	1
2	Game g = new Game(board);	Returns a board where the position of the chess piece is	2
3	Game g = new Game(board, state);	Returns a board where the position of the chess piece is	3
4	Game g = new Game(board, true, Piece.Color.WHITE);	Returns a board where the position of the chess piece is	4
5	Game g = new Game(true, board);	Returns Syntax Error	5
6	board[1][1] = new Piece(Piece.Type.KING, Piece.Type.WHITE);	Returns Address	6
7	board[1][1] = new Piece(Piece.Type.WHITE, Piece.Type.KING);	Returns Syntax Error	7
8	g.swapColor(); assertEquals(Piece.Color.BLACK,g.state.turn)	Changes the color of the chess piece	8
9	g.swapColor(); g.swapColor(); assertEquals(Piece.Color.WHITE,g.state.turn)	Changes the color of the chess piece	9
10	g.swapColor(); assertEquals(Piece.Color.WHITE,g.state.turn)	Returns org.opentest4j.AssertionFailedError:	10

11	Game g = new Game(); g.toString()	Returns Empty Board	11
12	Game g = new Game(true,board);	Returns Syntax Error	12