ERROR in Spanish Checkers: Queen's multiple square march move is not allowed.

Status: **UNSOLVED** - 20/10/2019

Error in:

The game set-up in SpanishCheckers.ini file is specified as follows:

```
[validRegularMove7]
boardSetUp=validRegularMoveBoard7
playerMove=4,4>7,1
turn=black

[validRegularMoveBoard7]
6,4=white-pawn
5,1=white-pawn
4,4=black-queen
1,3=black-pawn
4,6=white-pawn
```

Expected Behavior:

The queen should be able to move to the desired destination coordinate. Queens can travel multiple squares in one move without jumping any piece.

Actual Behavior:

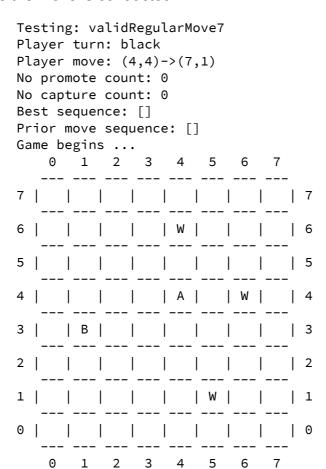
The queen is NOT moved to the desired destination coordinate. The player is asked for another move.

Error:

The move is not evaluated as a valid move. The game thinks that the queen can't travel multiple squares.

Test Outputs:

Before the move is conducted:



Test Outputs:

After the move is conducted:

Test: validRegularMove7 Informers: [There must be one piece on jump path 0, Player will be asked for another source coordinate because the previous move was invalid.] Final Informers: [There must be one piece on jump path 0, Player will

be asked for another source coordinate because the previous move was invalid.]

Was player going to make another move?: true End of the game?: false Board:

	0	1	2	3	_	4	5		7		
7	I	 	 	 	١					_	7
6	I		I	l 	١	W					6
5			I	l 	١						5
4			I	l 	١	Α		W			4
3	I	B	I	l	١		l				3
2	I		I	l	١						2
1	I		I	 	١		W			_	1
0	I		I	 	١					_	0
	0	1	2	3	_	4	5	6	 7	,	

HOW DOES CUCUMBER HELP US FIND THE PROBLEM?

The following method in AmericanCheckersScenarioTester (superclass of SpanishCheckersScenarioTester) fails (lines are renumbered):

```
1 @Override
2 public void thePieceAtTheSourceCoordinateIsMovedToTheDestinationCoordinate() {
3    //Check if destination coordinate now holds the moved piece.
4    assertEquals(pieceOfPlayerMove,
5         getPieceAtCoordinate(destinationCoordinateOfPlayerMove));
6    //Check if the source coordinate is now empty.
7    assertTrue(getPieceAtCoordinate(sourceCoordinateOfPlayerMove) == null);
8    //Check if the piece's current coordinate is the same as player move's
9    //destination coordinate.
10   assertEquals(destinationCoordinateOfPlayerMove,
11   pieceOfPlayerMove.getCurrentCoordinate());
12 }
```

And we know that the queen is not moved to the destination coordinate. We can debug the code to find which decision leads to that.

WHERE IS THE PROBLEM?

In the checkMove() method of SpanishReferee class, the RuleIfJumpMoveThen-JumpedPieceMustBeOpponentPiece rule class is evaluated. In its evaluate(...) method, the following check happens:

```
33 if(howManyPieceAreOnPath==0) {
34   if(moveConstraints.isMultipleSquareMarchMoveAllowed(moveDirection))
35   return true;
36 ...
```

It has to return true on the 35th line for this move to be valid, but QueenMoveConstrains class returns false from isMultipleSquareMarchMoveAllowed(moveDirection) for every possible direction.

Note: The same error also causes validRegularMove8 test to fail.