ERROR in Spanish Checkers: "Not the best sequence" error is not shown.
Status: **UNSOLVED** - 22/10/2019

Error in:

```
Scenario Outline: Invalid Destination Coordinate for Move
  Given the game is played up to a certain point from file "<file_name>"
  When the player picks a valid source coordinate that has a "<piece_type>"
    piece in it
  And the player picks an invalid "destination" coordinate because
    "<invalidity_reason>"
  >>>>>>>>> Then an error message is shown saying "<error_message>"
  And the player is asked for another "source" coordinate

  Examples:
    | file_name | piece_type | invalidity_reason | error_message |
    | invalidDestinationCoordinateForMoveNotBestSequence1 | queen |
       move is not part of the best sequence | Not the best move |
```

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

```
[invalidDestinationCoordinateForMoveNotBestSequence1]
boardSetUp=invalidDestinationCoordinateForMoveNotBestSequenceBoard1
bestSequence=6,0>4,2:4,2>2,4:2,4>0,2:0,2>2,0:2,0>6,4:6,4>3,7
playerMove=0,6>2,4
turn=black

[invalidDestinationCoordinateForMoveNotBestSequenceBoard1]
0,4=black-pawn
0,6=black-queen
1,1=white-pawn
1,3=white-queen
1,5=white-pawn
3,3=white-pawn
4,6=white-queen
5,1=white-pawn
5,3=white-pawn
6,0=black-queen
```

Expected Behavior:
The playerMove is not the expected move according to the calculated best sequence of moves (capturing the most opponent pieces). The move should NOT be conducted and the player should be asked for another move.

Actual Behavior:
The move IS CONDUCTED and the player is NOT asked for another move because the move was invalid, he is asked for another move because there are new possible jump moves.

Error:
The playerMove is not checked whether it conforms with the best sequence.

Test Outputs:

Before the move is conducted:

```
Testing: invalidDestinationCoordinateForMoveNotBestSequence1
Player turn: black
Player move: (0,6)->(2,4)
No promote count: 0
No capture count: 0
Best sequence: [(6,0)->(4,2), (4,2)->(2,4), (2,4)->(0,2), (0,2)-
>(2,0), (2,0)->(6,4), (6,4)->(3,7)]
Prior move sequence: []
Game begins ...
     0   1   2   3   4   5   6   7
   --- --- --- --- --- --- --- ---
7 |   |   |   |   |   |   |   |   | 7
   --- --- --- --- --- --- --- ---
6 | A |   |   |   | Z |   |   |   | 6
   --- --- --- --- --- --- --- ---
5 |   | W |   |   |   |   |   |   | 5
   --- --- --- --- --- --- --- ---
4 | B |   |   |   |   |   |   |   | 4
   --- --- --- --- --- --- --- ---
3 |   | Z |   | W |   | W |   |   | 3
   --- --- --- --- --- --- --- ---
2 |   |   |   |   |   |   |   |   | 2
   --- --- --- --- --- --- --- ---
1 |   | W |   |   |   | W |   |   | 1
   --- --- --- --- --- --- --- ---
0 |   |   |   |   |   |   | A |   | 0
   --- --- --- --- --- --- --- ---
     0   1   2   3   4   5   6   7
```

Test Outputs:

After the move is conducted:

```
TEST ABORTED!!!!!
Test: invalidDestinationCoordinateForMoveNotBestSequence1
Informers: [Player will be asked for another destination coordinate
(previous move was a jump move) (there are still possibilities for a
jump move).]
Final Informers: [Player will be asked for another destination coor-
dinate (previous move was a jump move) (there are still possibilities
for a jump move).]
Was player going to make another move?: true
End of the game?: false
Board:
      0   1   2   3   4   5   6   7
    --- --- --- --- --- --- --- ---
7 |   |   |   |   |   |   |   |   | 7
    --- --- --- --- --- --- --- ---
6 |   |   |   |   | Z |   |   |   | 6
    --- --- --- --- --- --- --- ---
5 |   |   |   |   |   |   |   |   | 5
    --- --- --- --- --- --- --- ---
4 | B |   | A |   |   |   |   |   | 4
    --- --- --- --- --- --- --- ---
3 |   | Z |   | W |   | W |   |   | 3
    --- --- --- --- --- --- --- ---
2 |   |   |   |   |   |   |   |   | 2
    --- --- --- --- --- --- --- ---
1 |   | W |   |   | W |   |   |   | 1
    --- --- --- --- --- --- --- ---
0 |   |   |   |   |   | A |   |   | 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 public void anErrorMessageIsShownSayingP1(String p1) {
2   assertTrue(referee.getInfo().getFinalInformers().size() > 0);
3   assertEquals(p1, referee.getInfo().getFinalInformers().get(0));
4 }
```

And we know that the error message is not shown to the user and the move is conducted.

WHERE IS THE PROBLEM?

There is no logic in the code that respects the "best sequence" rule of Spanish Checkers.

Note: The same error also causes
`invalidDestinationCoordinateForMoveNotBestSequence2` and
`invalidDestinationCoordinateForMoveNotBestSequence3` tests to fail.