

ERROR in American Checkers: When source coordinate is empty, the printed message is not “No piece at source coordinate” but “If any opponent's pieces can be captured it must be captured first!!!!”

Status: **UNSOLVED** – 23/09/2019

Error in:

Scenario Outline: Invalid Source Coordinate for Move

Given the game is played up to a certain point from file "<file_name>"

When the player picks an invalid "source" coordinate because "<invalidity_reason>"

And the player picks any destination coordinate

>>>>>>>> **Then** an error message is shown saying "<error_message>"

And the player is asked for another "source" coordinate

Examples:

file_name	invalidity_reason	error_message	
invalidSourceCoordinateForMoveEmpty2	source coordinate is empty	No piece at source coordinate	

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

```
[invalidSourceCoordinateForMoveEmpty2]
boardSetUp=validJumpMoveBoard1
playerMove=3,3>4,4
turn=black
```

```
[validJumpMoveBoard1]
5,3=black-pawn
6,6=black-king
6,4=white-pawn
5,5=white-king
```

Test Outputs:

Before the move is conducted:

```
Player turn: black
Player move: (3,3)->(4,4)
B: Pawn of 'black' player
A: King of 'black' player
W: Pawn of 'white' player
Z: King of 'white' player
  0  1  2  3  4  5  6  7
  ---
7 |  |  |  |  |  |  |  | 7
  ---
6 |  |  |  |  |  |  A  | 6
  ---
5 |  |  |  |  |  Z  |  | 5
  ---
4 |  |  |  |  |  W  |  | 4
  ---
3 |  |  |  |  B  |  |  | 3
  ---
2 |  |  |  |  |  |  |  | 2
  ---
1 |  |  |  |  |  |  |  | 1
  ---
0 |  |  |  |  |  |  |  | 0
  ---
  0  1  2  3  4  5  6  7
```

After the move is conducted:

```
Test: invalidSourceCoordinateForMoveEmpty2
Status: Test ended with an invalid player move.
Informers: [If any opponent's pieces can be captured then it must be captured first!!!!,
Player will be asked for another source coordinate (previous move was invalid)...]
Board:
  0  1  2  3  4  5  6  7
  ---
7 |  |  |  |  |  |  |  | 7
  ---
6 |  |  |  |  |  |  A  | 6
  ---
5 |  |  |  |  |  Z  |  | 5
  ---
4 |  |  |  |  |  W  |  | 4
  ---
3 |  |  |  |  B  |  |  | 3
  ---
2 |  |  |  |  |  |  |  | 2
  ---
1 |  |  |  |  |  |  |  | 1
  ---
0 |  |  |  |  |  |  |  | 0
  ---
  0  1  2  3  4  5  6  7
```

Expected behavior:

The move should not be conducted. The player should be shown an error message saying “No piece at source coordinate”. And the player should be asked for another source coordinate.

Actual behavior:

The move is not conducted. The player is shown an error message saying “If any opponent's pieces can be captured it must be captured first!!!!”. The player is asked for another move.

Error:

The current source coordinate errors hierarchy does not support this scenario.

HOW DOES CUCUMBER HELP US FIND THE PROBLEM?

The following method in AmericanCheckersScenarioTester fails in the 4th line:

```
1 @Override
2 public void anErrorMessageIsShownSayingP1(String p1) {
3     assertTrue(informers.size() > 0);
4     assertEquals(p1, informers.get(0));
5 }
```

Its output is:

```
org.junit.ComparisonFailure: expected:<[No piece at source coordinate]> but was:<[If
any opponent's pieces can be captured then it must be captured first!!!!]>
```

From that, we know the error message shown to user was not “No piece at source coordinate”.

WHERE IS THE PROBLEM?

In Referee class, in the following method, the check for move being one of the possible jump moves is made before the check for non-empty source coordinate is made:

```
protected boolean checkMove() {
    return isSatisfied(new RuleIfAnyPieceCanBeCapturedThenMoveMustBeThat(), this)
        && isSatisfied(new RuleThereMustBePieceAtSourceCoordinate(), this)
    ...
    //rest of the method
}
```