hizalc

These are implementation level contracts.

Operation: insertWord(ArrayList<Tile> tiles)

Preconditions:
- tiles are in player's rack or already in play on the board
- tiles <= [size of board]

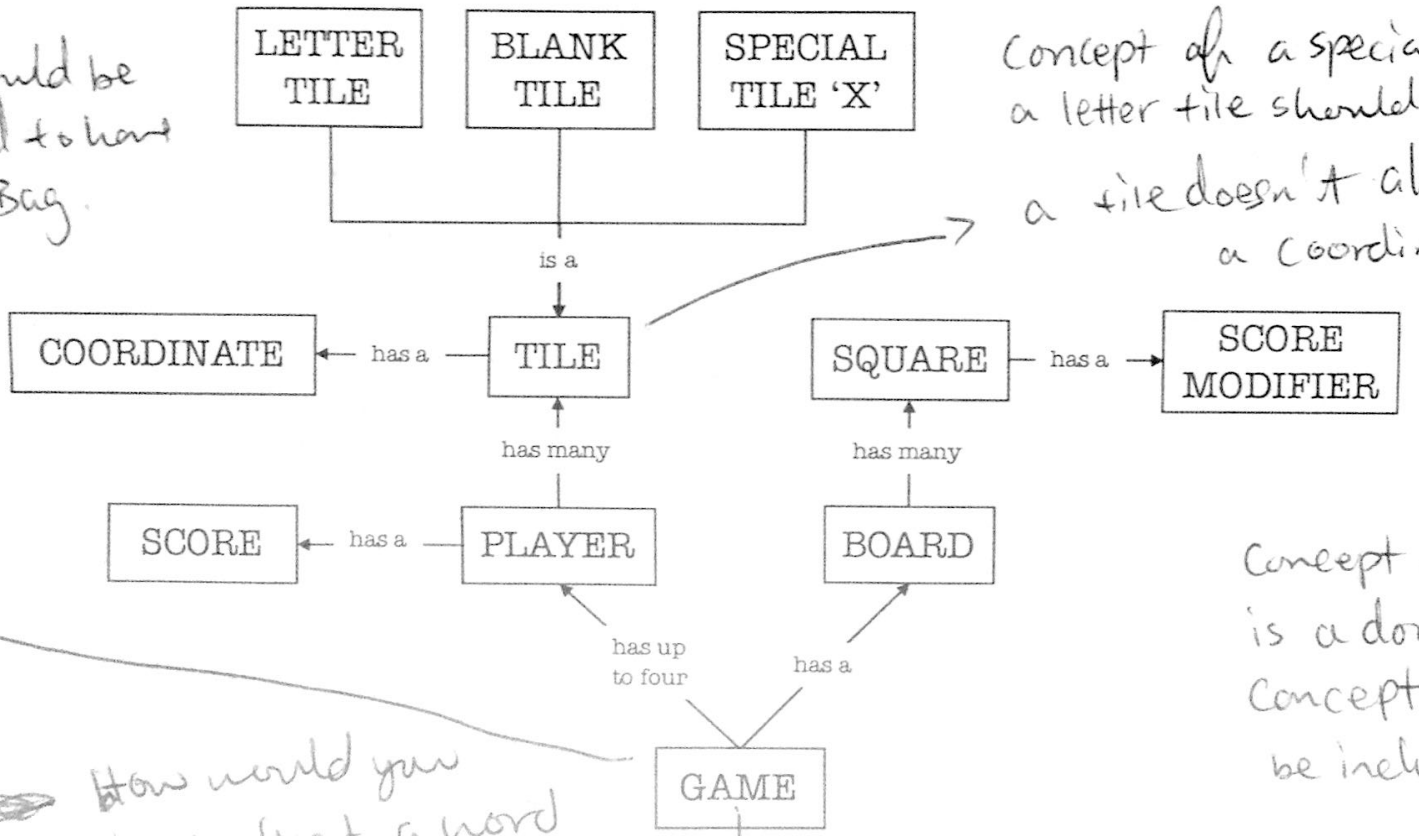
Should phrase to be like there should be at most 7 tiles ~~location~~ in a move.

Postconditions:
- tiles are assigned to squares on board
- the row/col of assigned tiles makes a valid word
- any existing score modifiers were implemented (if in same location as any letter tile from inserted word)
- any existing specials were activated, in order of insertion to the square (in case of multiples)
- score was updated accordingly
- player's rack was updated (removed used tiles, added new ones)

This behavioral contract is ~~there~~ at the implementation level. The behavioral contract should be at the domain level. This is at the wrong abstraction level.

A tile should never be blank.
A square can be blank.

Notation should be have boxes always having another box in the field.



A Game should be ~~How~~ expected to have a tileBag.

Concept of a special tile and a letter tile should be different.

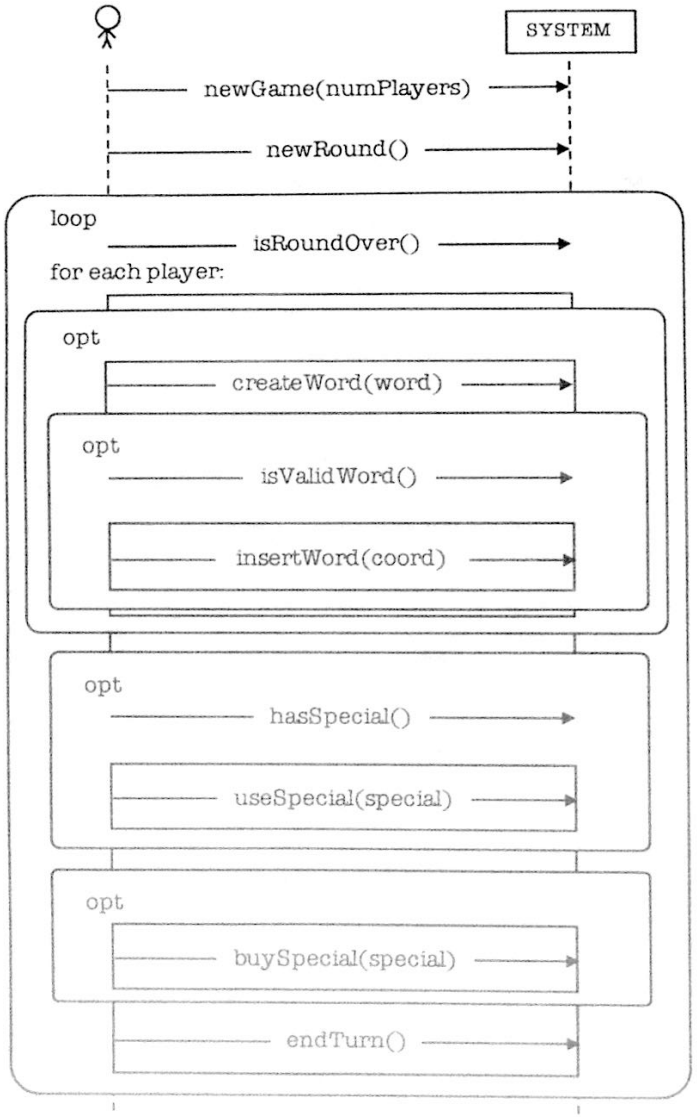
a tile doesn't always have a coordinate.

Concept of a word is a domain level Concept that should be included.

How would you check that a word is valid? concept of a dictionary

what kind of Game?

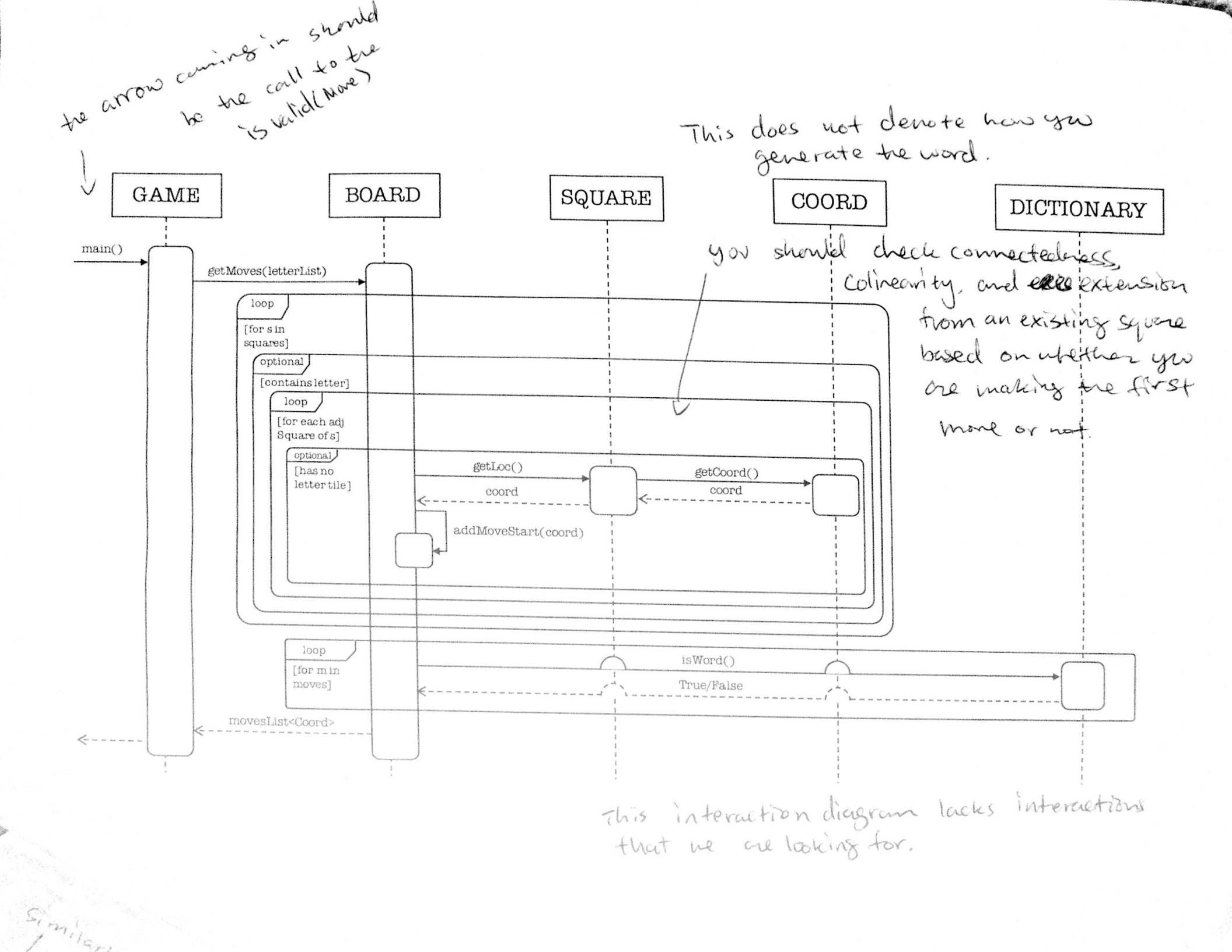
ScrabbleGame would be more appropriate.



- you should include loop guards.

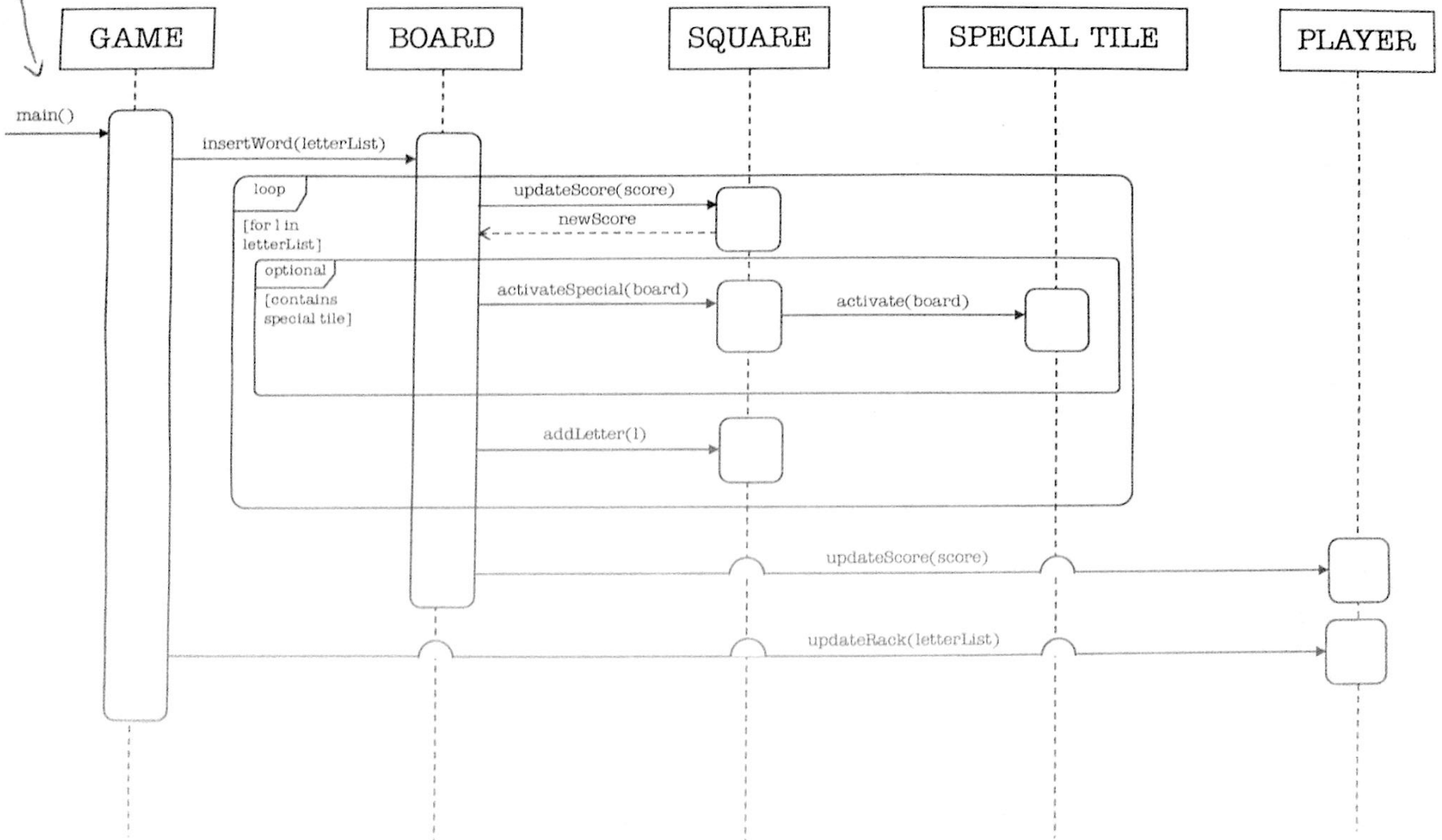
what about the possibility of just exchanging tiles and/or passing your turn?

It is unclear what you are doing because of the lack of loop guards. This should iterate over all rounds and return the final result.


the arrow coming in should be the call to the isValid(Move)
This does not denote how you generate the word.
GAME
BOARD
SQUARE
COORD
DICTIONARY
you should check connectedness, colinearity, and extension from an existing square based on whether you are making the first move or not.
main()
getMoves(letterList)
loop
[for s in squares]
optional
[contains letter]
loop
[for each adj Square of s]
optional
[has no letter tile]
getLoc()
coord
getCoord()
coord
addMoveStart(coord)
loop
[for m in moves]
isWord()
True/False
movesList<Coord>
This interaction diagram lacks interactions that we are looking for.

Similarly to the comment in the previous interaction diagram.

This interaction diagram doesn't show how the words are computed.



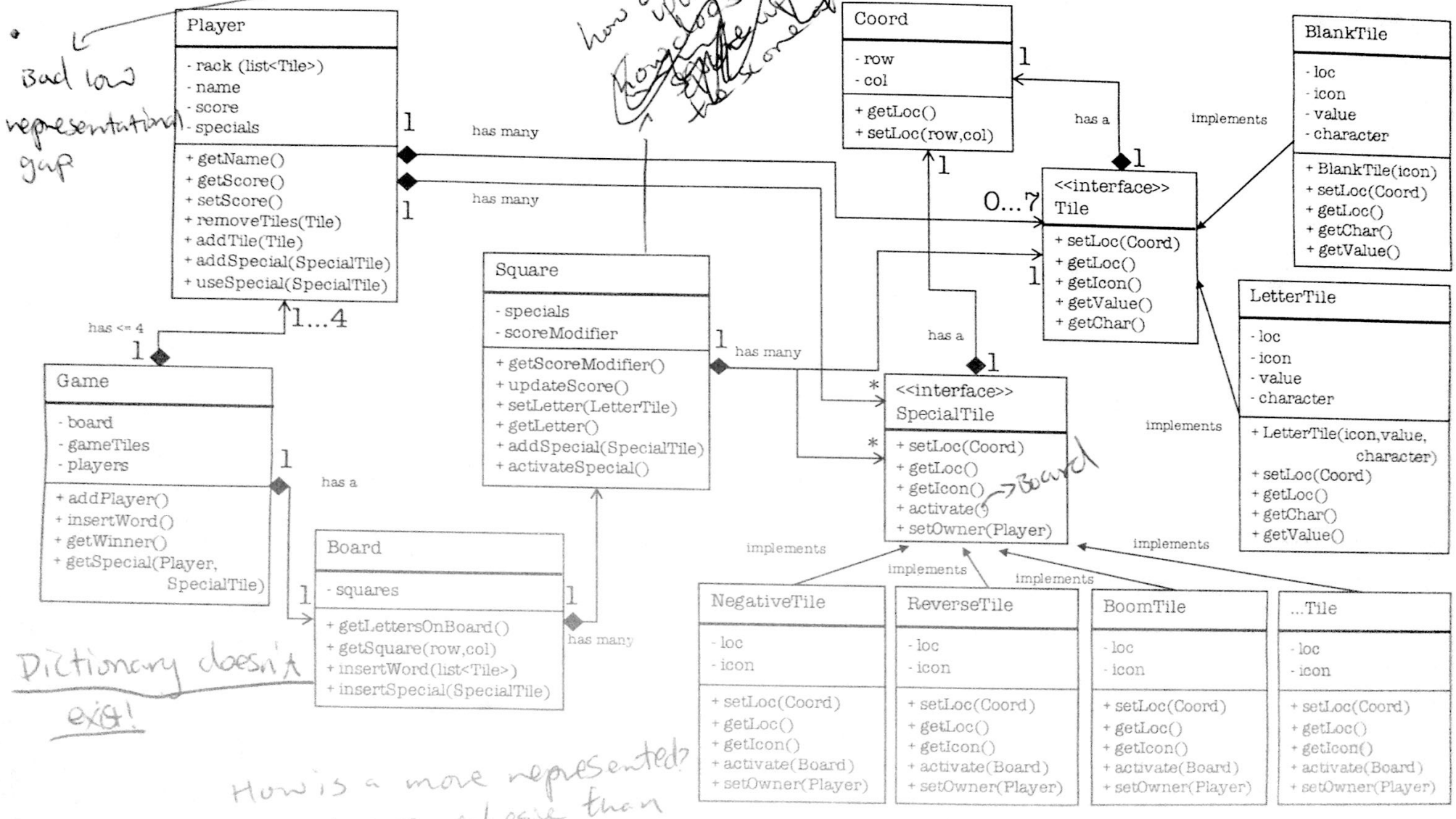
There are placements of special tile that should be included in a move.

The division of responsibility is not as cohesive as possible and unclear.

Adding a concept of a word would be better for cohesion.

how does square update the score if you don't pass in the player

Bad low representational gap



Dictionary doesn't exist!

How is a move represented? This is less cohesive than just using a move object.

how can you access the player order?