

Homework 4a - Design Feedback

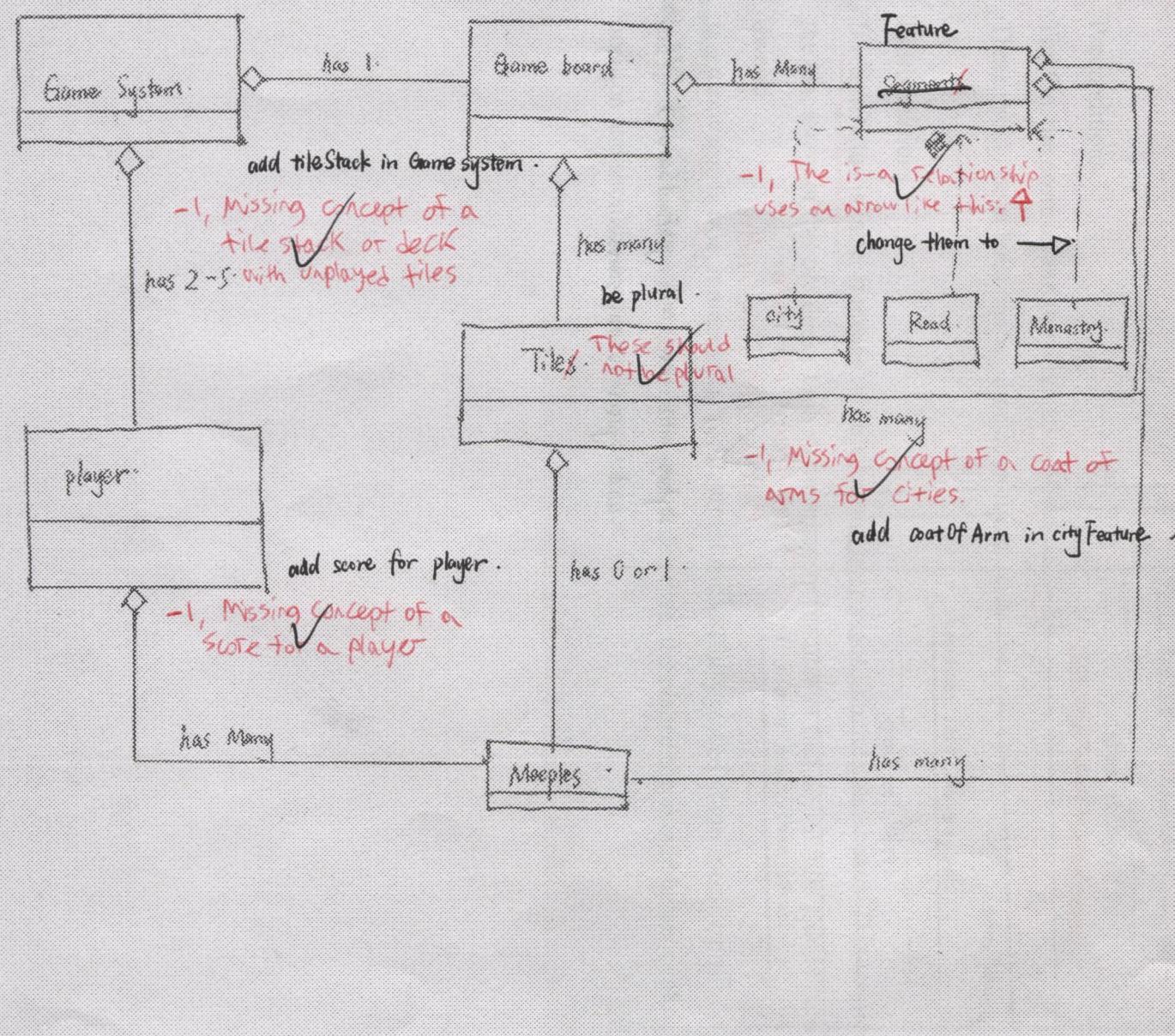
Andrew ID: luxingj

Section	Points	Score
Notation	20	18
Domain model	16	8
System sequence diagram	8	6
Behavioral contracts	4	3
Interaction diagram (move validation)	16	11
Interaction diagram (monastery scoring)	16	11
Object model	16	13
Justification	4	1
Consistency	20	10
Extra credit: single printable document	2	2
Total	120	83

You can regain up to 75% of the points lost in this milestone in Milestone C by addressing our comments and updating your design documents. If you have lost a large number of points, we recommend some interaction with the course staff.

Graded By: Ari Cohn (acohn@andrew.cmu.edu)

Domain Model



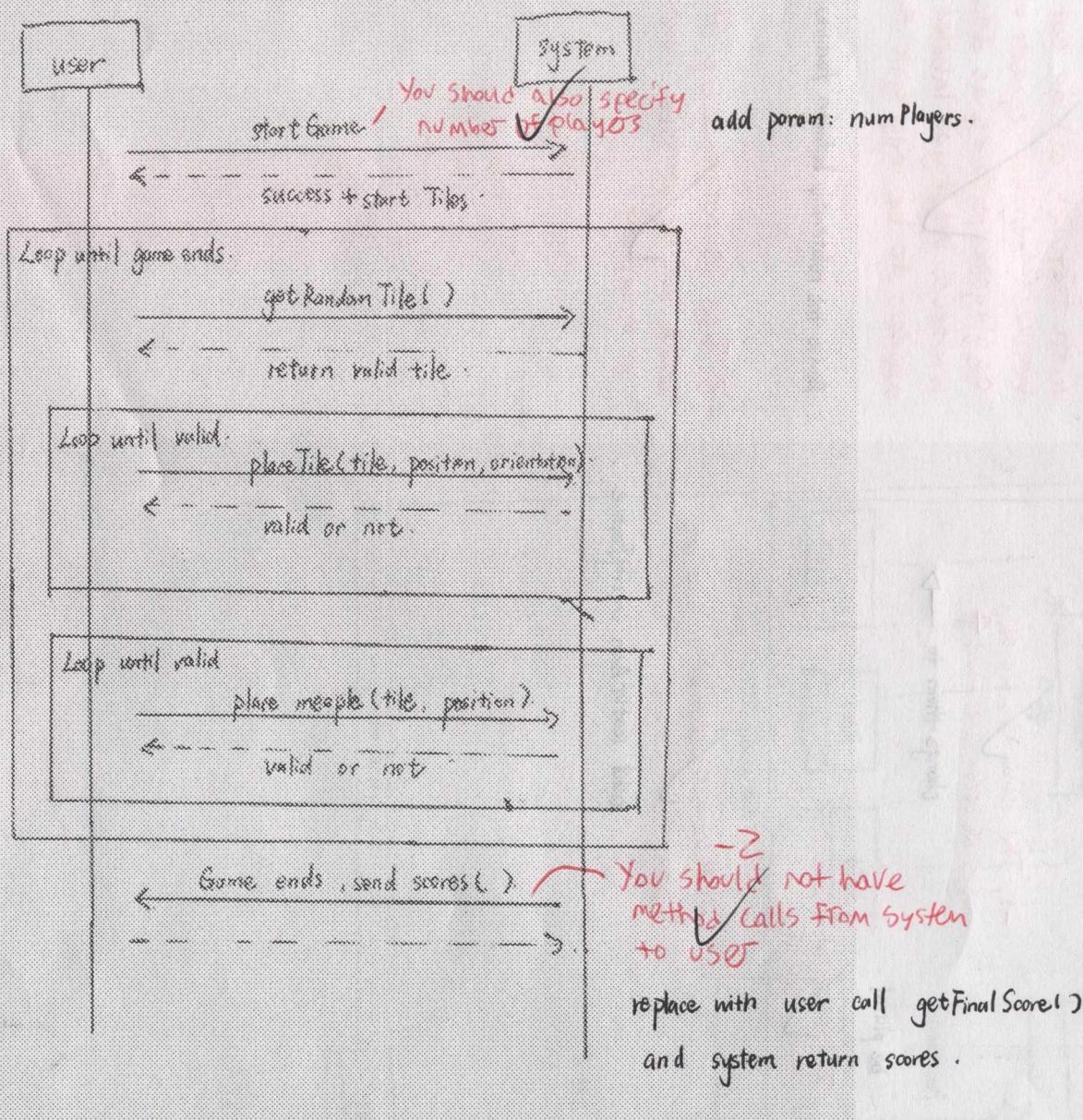
Rename ~~Segments~~ to Features

-2, You seem to be confusing segments with features and do not make a distinction between individual segments on a tile and aggregate features, as discussed in the writeup. What you labeled as Segment appears to be aggregate features.

Revise the relationship between features and tiles.

-3, Tile is not made up of segments which can be of type Road, City, Field, or Monastery.

System sequence diagram.



Behavior Contract

Playing a tile (without a meeple).

These are implementation details and do not belong here

Function: boolean placeTile(Tile tile, Position pos, int orien)

This function implements the placement of a tile with the given position and orientation. If return value is true, the game system will record this placement and pass the control to next player. If return value is false, user will have to change the position or orientation until the placement is valid.

Preconditions:

The tile, the position and the orientation should be valid. ~~placed~~ ^{ok}

There should be at least one legal position for this new tile.

Postconditions:

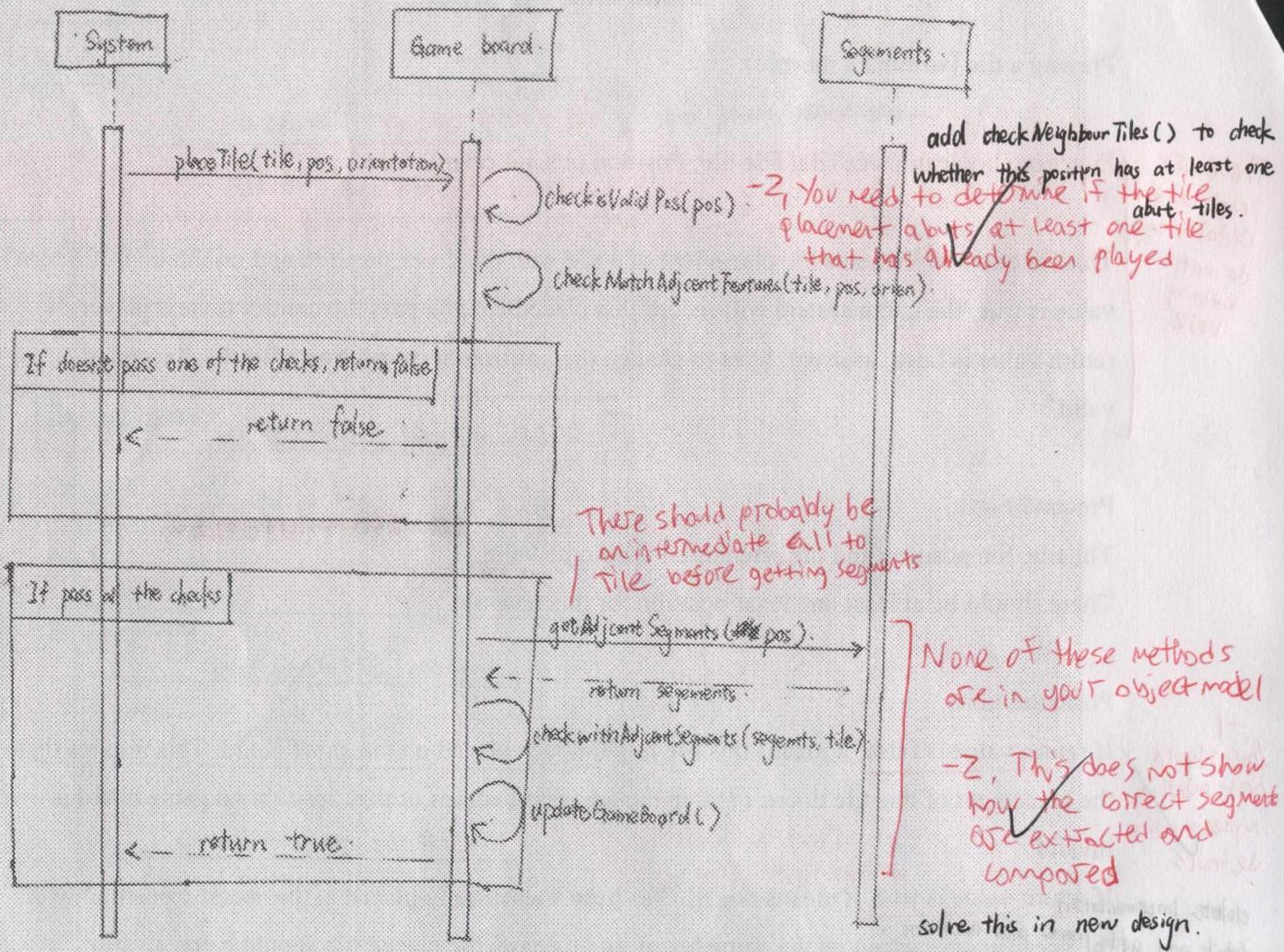
-1
You should not mention implementation details.

If return value is false, it means this tile has not been placed on the game board. This implies that the placement of this tile doesn't match the adjacent features or that position on game board is invalid.

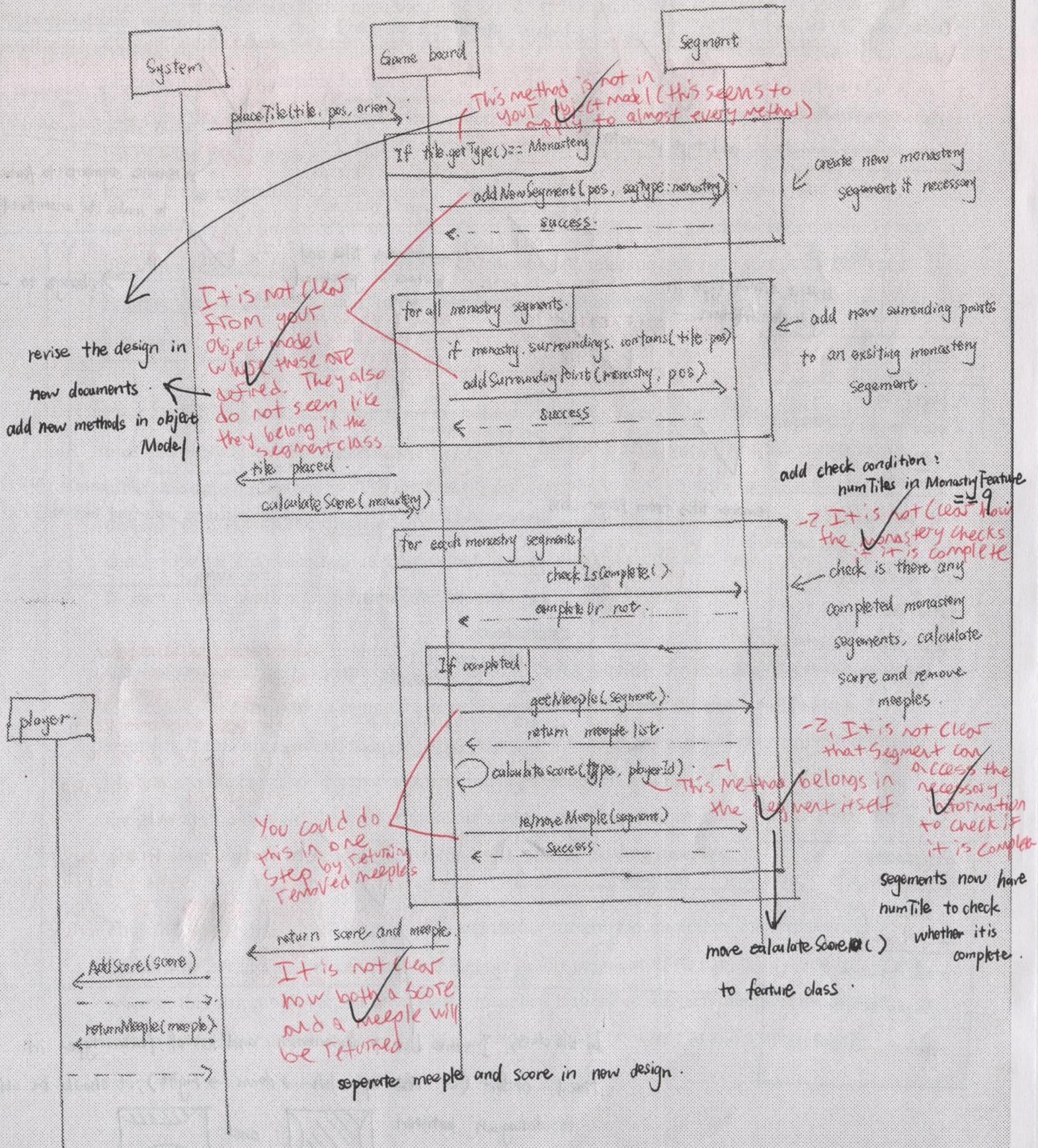
~~delete implementation~~ If return value is true, it means this tile has been successfully placed at the specific position with ~~details and add more post conditions~~.

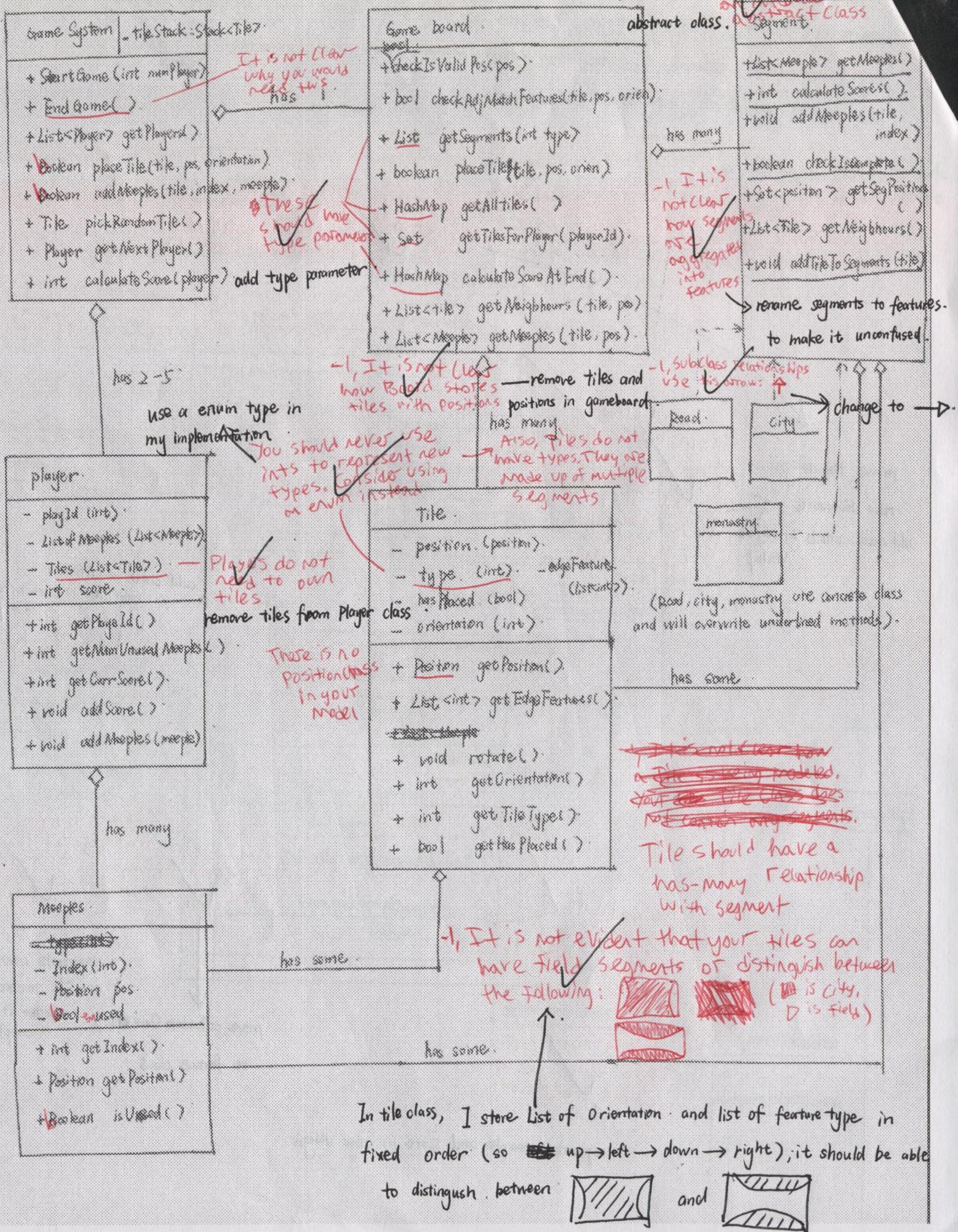
Object-level interaction diagrams.

- Describe how the placement of a tile (without a meeple) is validated by the game.



Suppose a valid tile placement (without a meeple) completes one or more previously monasteries (possibly containing meeples). Describe how the game detects newly completed, previously played monasteries, determines whether they contain meeples, scores the monasteries as needed, and returns any scored meeples to their players.





Design Rational

Start with the domain model part, the game system is responsible for managing the game logic throughout the whole process. There will be 2 to 5 players in this game and one game board object which is responsible for managing all the tiles and segments on the game board. There is a Tile class that represents 72 tiles in this game, where each tile has its own type, orientation and a list of features. There is a segment abstract class which represents different segments in the game board. It can be extended by a city segment, a road segment or a monastery segment and each segment can be composed of multiple tiles. The segment class is responsible for combining new tile into a segment, checking if there is any completed segment and calculate scores for players. Finally, there is a meeple class, one player can have multiple meeples and each tile can have 0 or 1 meeple.

First, the game system starts the game with the number of players. The system will generate start tiles and place them on the game board and decide the order of players. Then the game system picks one random tile and pass to the first player. The player now chooses to place this tile in his desired position with the specified orientation on the board. The game board will be able to check whether this placement is valid or not. If not valid, the player will have to retry until receive a valid confirmation from game system.

After placing the tile, the game board will have to check whether the features of this tile, match the features of surrounding segments. In other words, whether this tile can be added to a segment. If this tile is added to an existing segment, this segment will have to be updated. If this tile leads to the creation of a new segment, this new segment will have to be added. After that, the game board will check whether there are any newly completed segments. If there are, the game board needs to record this. Then comes the step of placing meeples.

Then the system will check whether this segment is currently occupied by other players and decide whether this player can place his meeple in this segment. If he can, the system first checks whether this current player has any unused meeples. If there are unused meeples, the system responds to player and this player needs to decide whether he wants to add a meeple in this

placed tile and which position he wants to add. If there are any recorded completed segment, the game board have to calculate score for this segment and decide which player wins this score. It will also remove the meeples in this completed segments. The the game board have to return the added score, the corresponding player and removed meeples to the system. The system will then update scores and meeples for each player.

Finally, the first round finishes and the control is passed to the next player in a fixed order. The game system starts a new round and repeat the above steps until all the tiles are used. When the game ends, the game board is responsible for calculating scores for all the remaining uncompleted segments based on the rule. Then the final scores are passed to game system so it can now decide who is the winner.

- 3, Your justification does not refer to design goals or principles at all

add design goals and principles in new justification.