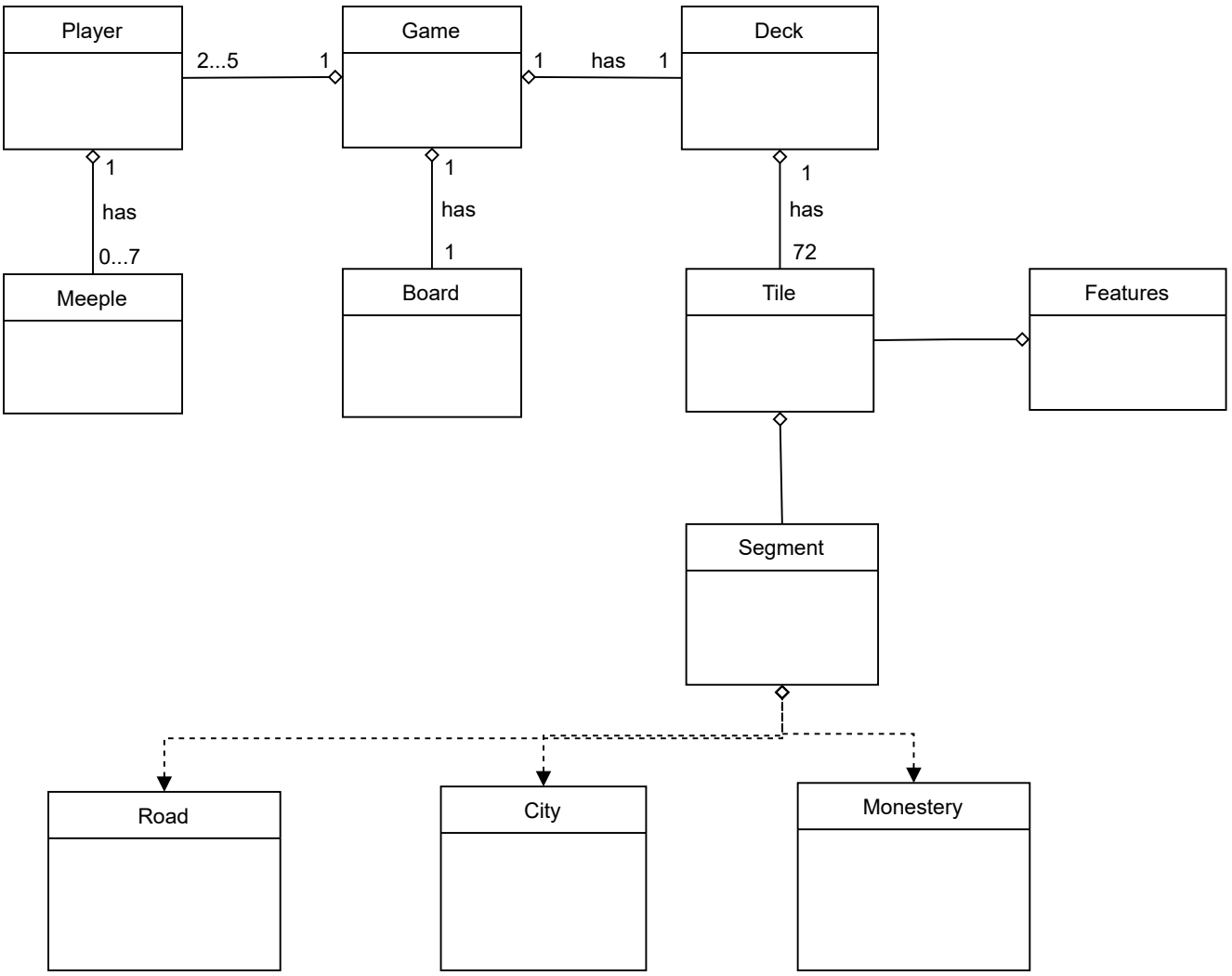
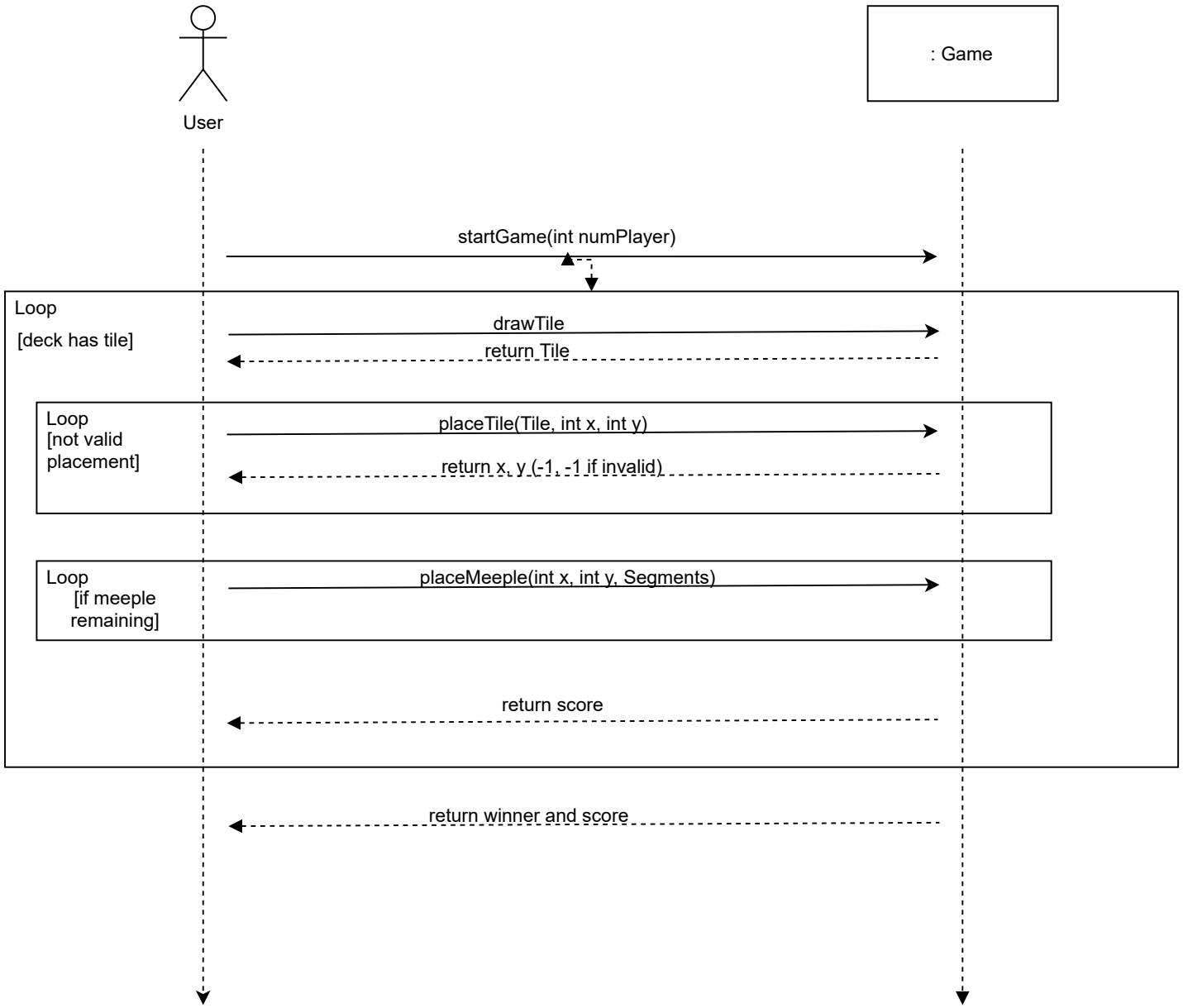
Domain Model



System Sequence Diagram



Behavioral Contract

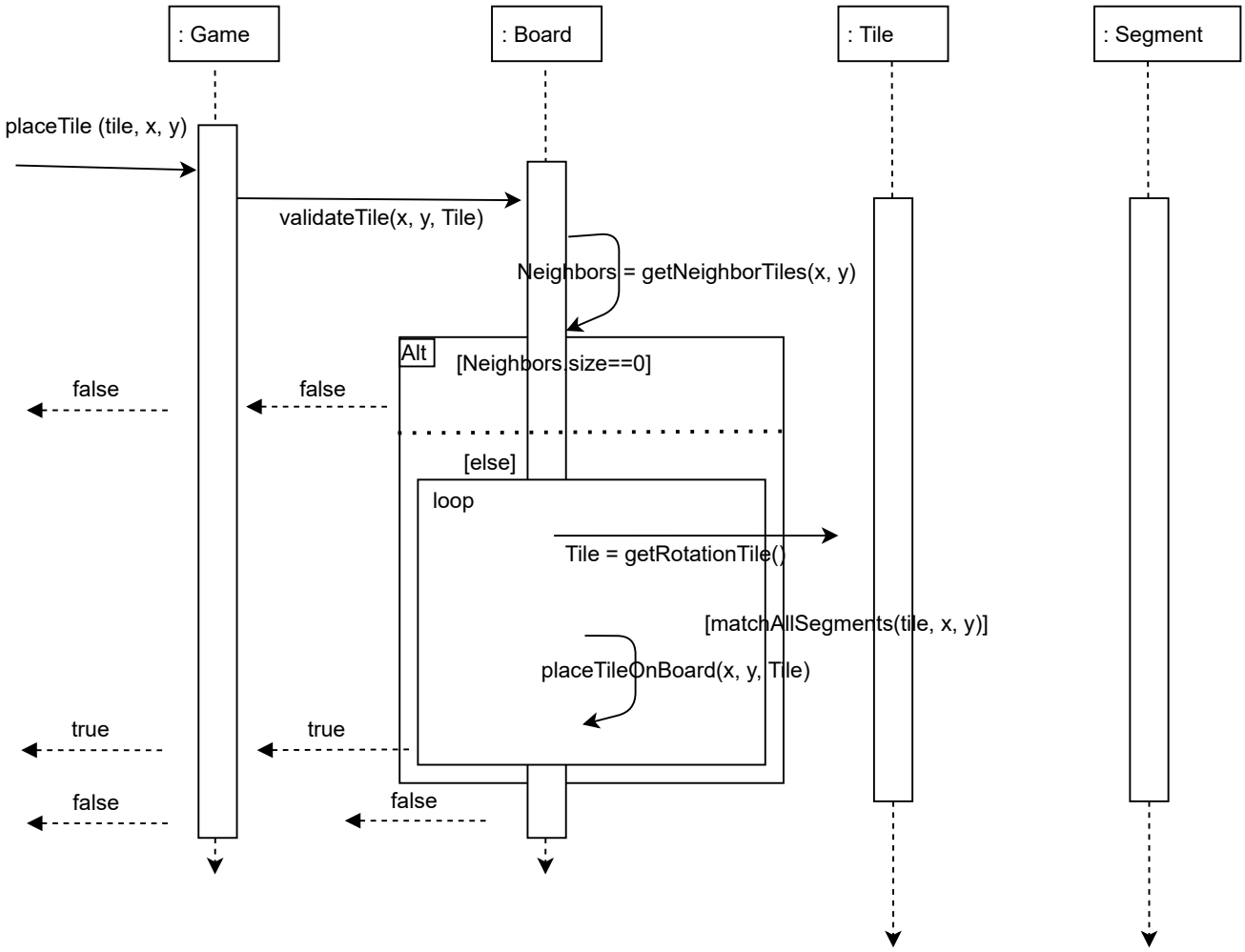
Operation: placeTile(x, y, Tile)

Preconditions:
- x > 0 and y > 0

Postconditions:
- the 4 directions of tile segments match its 4 adjacent tile segments.

Object-level interaction diagrams 1

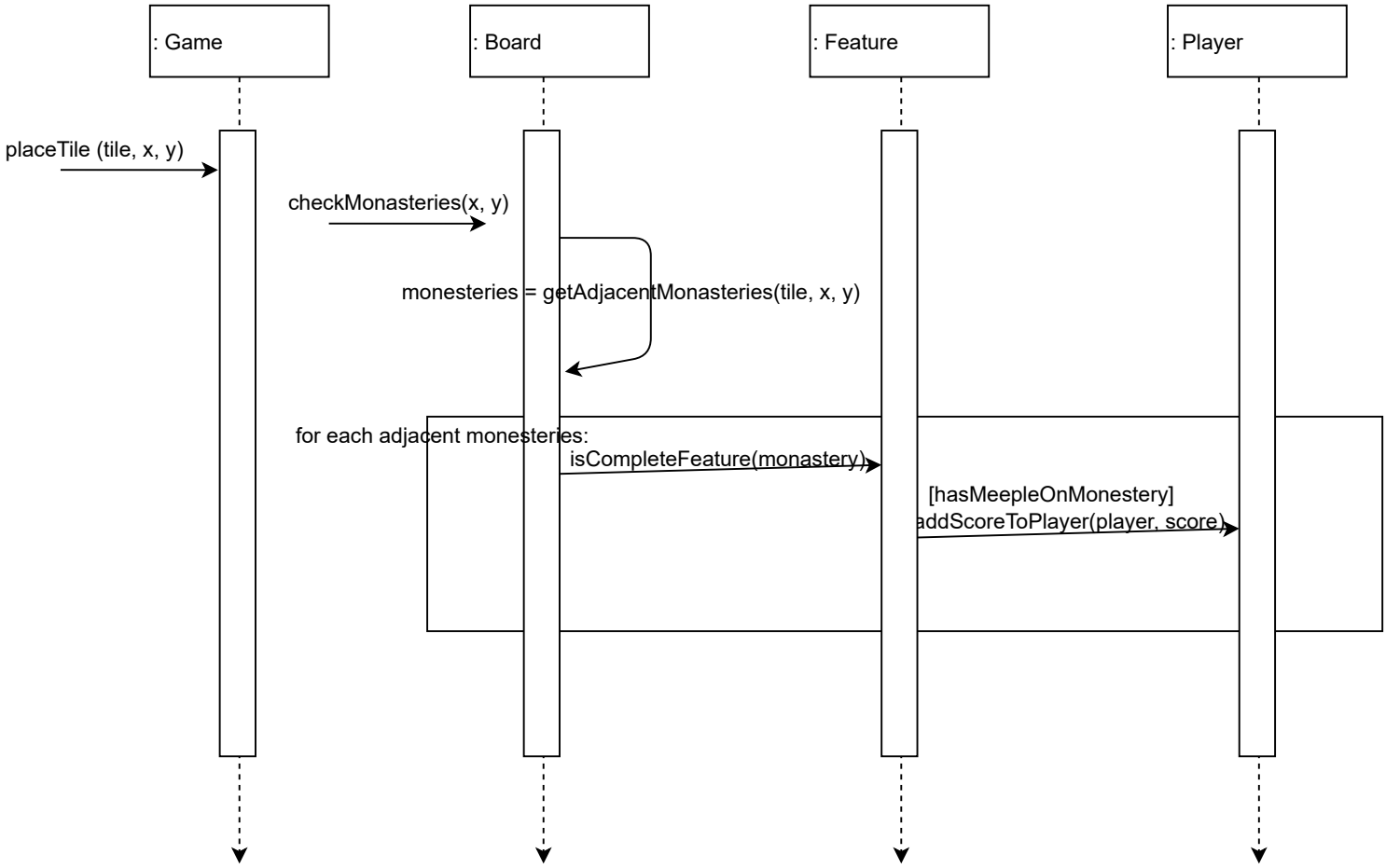
Describe how the placement of a tile (without a meeple) is validated by the game. You do not need to include any scoring-related details in this scenario.



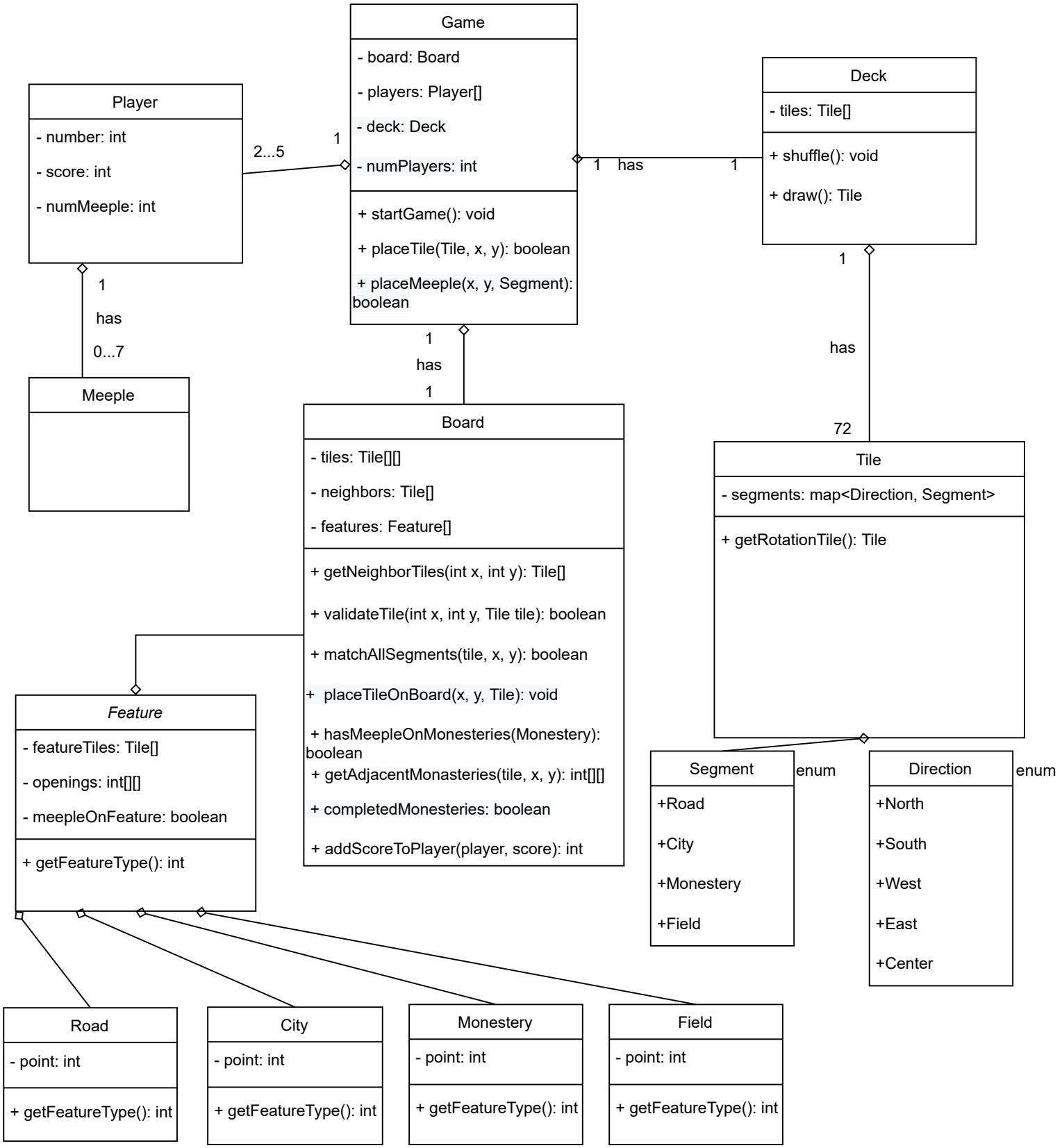
Side note: getRotationTile is a method simulated physical tabletop version of Carcassonne, where tile can be rotated to fill into board. If not rotated well, the tile cannot be placed onto the board.

Object-level interaction diagrams 2

Suppose a valid tile placement (without a meeple) completes one or more previously played 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.



Object Model



Rationale

A player interacts with the game by place tiles and place meeples. The game will add score to the player, and at the end of the game the player with highest score will win.

At the start of the game, a user calls startGame() method in Game object to initialize game, and the user should specify number of players. Players will take turns to play.

A player can place a tile with a meeple, which will deduct number of meeples in a player's object, and add meeple to a feature.

The current placements of tiles represented as a Tile object with a map<Direction, Segment>, which contains Enums of 4 directions (and center) and 4 different feature representations.

Tile rotations represented by calling rotate() method in Tile objects which will rotate the tile.

When a new tile is placed, the game determine the newly completed features to be scored by calculating featureTile.size() * point in feature object.

The game determine if a meeple placement conflicts with a previously played meeple by checking meepleOnFeature is true or not on the other part of feature.

I have a polymorphism about different features. So monasteries' point is different from cities' point.