Rationale:

I made the GameBoard class check valid tile placement, and city/farm features are implemented such that there are 8 places they can belong on a tile, 2 on each side, cloister features are 1 per tile, and road features can have at most 4 indexes on a tile. The FeatureWrapper class essentially functions as a pointer, so when a new Tile is placed down, the merge() function can just create a combined feature (of the placed tile and the other either up,down,left, or right) and make the FeatureWrappers on both tiles point to the same, new, combined feature.

In the feature classes, I implemented all the different scoring methods for each type of feature, along with their scoring methods when the game ends.

I also added a next() method into the GameSystem, so it would be easier for the GUI to complete all the operations it needs in its thread, such as placing a follower onto the newly placed tile, and pass on the correct information into the GameSystem for the next state of the game. Originally, I made GameSystem call next() by itself, after a tile is placed, but that made my GUI really buggy, since the GUI would still be waiting for the player to place a follower, while GameSystem already finished scoring the tiles.

I tried my best to adhere to the observer pattern and use good information hiding on my GameSystem; the public methods in GameSystem are relatively simple (placeTile, placeFollower, next, etc), and let my other classes (Tile, Feature) handle the more complex work. The Feature classes are definitely the most complicated classes in my implementation, but I think it deserves to be so, since the entire game is based on features. The features can merge when a new feature connects to it via a new tile, followers have to be removed correctly when a feature is scored, and all the features are scored in a different way

Inside the next() method is score, drawTile, changing the current player, and checking if the game is over. If the game is over (no more tiles), the method will call scoreIncomplete() to score the rest of the board correctly.