Discussion:

In my original object model, I put a lot of game-playing methods (like placeTile, drawTile) inside the Player class. After receiving feedback and starting to code the game, I moved the game-playing methods into GameSystem, and used the Player class as only a class to store information about each player (score, followers left), and let GameSystem handle the tile-getting, placement, calculating scores, and so on. I implemented my deck of tiles as a stack, instead of an array, so that it's easier to keep track of which tiles are gone from the deck already (just pop from the stack, instead of generating a random index to get from the array).

I also added placeFollower(index) into GameSystem, so that the action of placing a follower onto a tile is more well-defined (can specify which feature to put follower on).

The most difficult part of this whole project was probably ensuring that after scoring, the correct followers get removed from the scored tiles. I did this by storing the FeatureWrapper of what feature a follower is placed on, then when scoring, check if the follower's feature is indeed the actual feature that is being scored, instead of another feature on the same tile.