

One key element of my object model design is the segments. I designed it so each tile has 4 segments, one for each of the sides of the tile. So if the tile is a 4 way crossroads, <sup>→ some sides can have more than 1 segment</sup> it would contain 4 road segments. There are 4 different kinds of segments: field, city, road, and cloister. They all have shared functionality to check if a combination of segments is valid <sup>?</sup> and to get the followers associated with a particular segment. <sup>→ It could have many more.</sup> <sup>→ Scoring? is complete?</sup>

Because of this they implement the segment interface. The strategy pattern will be used here with the differing algorithm being the differing calculations corresponding to the different segments.

I chose what objects to include with a heavy focus on real world representation. Most of the objects are taken directly from the rulebook, including the turn object. This serves as the workhorse of moving the game forward. It contains the functionality for the user making choices for the most part. I chose for the turn object to contain this rather than <sup>→ the user should do that?</sup>

the player object because I wanted the player object to be more of an information chest for the player rather than the control board for the user. Giving the turn object this <sup>→ user ≠ player → see course staff for clarification</sup> functionality also splits the game nicely into all of its turns rather than having them blend together.

One situation I made sure to account for in my design was the one where two followers exist on separate segments and are then combined. This results in multiple followers existing in a feature, whether that feature is complete or not. Since the segment has the follower instead of the feature its simple to build up a count of followers in the growing feature as more tiles are added. We can just examine all the segments that make up the feature and that will give us an exact count of the followers.