Discussion:

Basically, my main design idea does not have big changes over three milestones, but I changed the format to present the logic and design pattern and the class combinations. In the milestone A, I planned to use interface and abstract class to denote different features and segments, but I changed it into Enum format in the real implementation. Originally, I was not quite sure how to distinguish the segment and feature, then I just defined segments are the four edges and center's characteristic type, and the feature is a bunch of the connected tiles with same segment adjacencies. Also, I planned to put check feature completion and score methods in the feature class, but in milestone b, I found I could just integrate them into the Segment class, because each feature and each segment type has one to one relationship, in this way, I could use fewer classes. Also, to reduce unnecessary classes, I give up separate Player class and Meeple class, combined them in to the one class—— Carcassonne class, because they can be simplified and can be accessed by index if using Array or List. As you can see, the main changes I made are to simplify the code and keep the reasonability as possible as I can.

Besides, there some logic changes while designing functionalities of checking valid placement and valid meeple placement, feature updating, checking completion and scoring. The most impressive idea I believe is that I denote segments of a tile by a list, by using index to map the segments on four edges and center. Then for each edge and center, I will generate a new feature (if it is not merged with others), and with a unique feature code, also is the index of the feature list. So, it is very easy to know four edges and the center belong to which type and exactly which feature. To check the completion, it is very easy, just to check the feature codes of edges of tiles on current (merged) feature that abutting with vacancies, if all of them are not the same as current feature, then it proves current feature is completed.