Another situation I based my design around is that of validating a placement of a tile on the board. The validity of a tile placement is first checked by the board to ensure that the tile is abutting another tile and it is not being placed in an occupied spot. After that there is a check done to ensure that the abutting segments match appropriately. I chose to have the first check done by the board and the second check done by the segment object. It felt more appropriate this way because its more intuitive that a segment object makes sure the segments are correct and the board object makes sure the tile is put in a valid spot on the board.

↳ one segment shouldn't know about other segments

One thing that should be explained is how the board is being represented. It is a two-dimensional list of tiles. It starts with the starting tile right in the center. As tiles are placed we add the tile to the correct spot in this 2d list. We will make it a 73 by 73 list to begin with in order to contain any possible combination of tile placements.

The score is calculated by the player object as the score is stored as an attribute on the player object. I felt it would make the most sense if they were in the same object and in building the object interaction diagrams it worked out well this way. The Carcassonne object is mainly used just to start the game and be a sort of wrapper object for the rest of the game. It takes user input in the initialization of the game like the number of players desired.

- these arguments are just describing how your implementation will work, there is no reference to any design goals (extensibility, reuse, etc.)