The game of Parcheesi involves 4 players, a board (contains 4 starting points with 4 pawns each, 80 squares, starting point, and home), and 2 dice. In our implementation, the class Player has 4 components: pawnPosn[4][3] (square number, safety status, and blockade status of each pawn), pawnFinish[4] (finish status of each pawn in Boolean form), diceRoll[2] (randomized dice roll values), doubleRoll (keeps track of consecutive double rolls by a single player to reinforce different rules).

In another class Play, players can roll the dice, with consequences depending on how they decide to move their pawns. addRollDie assumes that if the conditions for moving a pawn out of the starting point, then a pawn will automatically be started. Else, the dice rolls will be added to other random pawns already on the board. doubleDieRoll keeps track of how many consecutive double rolls a player has achieved, since there are different consequences depending on the number of double rolls and the status of each pawn. blockades compares each pawn for the same player and determines whether a blockade has been formed. safety works similarly, except only the pawn position values are compared to a predetermined “safe” squares. getToHome allows a player to move a pawn to home if and only if the dice roll and the current position adds up to 82 (predetermined position number for home).

More implementation has to be done for various functionalities, including bop, preventing players from passing a blockade, and determining which pawns get to move depending on what the outputs of rollDie are. Once there are multiple players, the server should be able to access player information and determine what the consequences are of each dice roll. The only possible actions for a player is to determine which pawn moves how many squares, and whether to form blockades or move to a safe position when possible.