1: first legal.  Backup algorithm, in case nothing else works.

scan board

try your tile in each place

   try your tile in each orientation

   if it fits, use that

if no legal moves, say that

documents that should help:

A: <http://web.cs.du.edu/~sturtevant/papers/multiplayergamesthesis.pdf>

2: maximin / minimax: without knowing the actions of other players, evaluate all possible moves that you can make, and calculate how much that will give you

//At beginning:

reserve 400-length arrays to hold data of possible moves (placement, orientation, meeple, score to be gained, etc)

int possibleMoves; //to count the number of possible, legal movements for  a turn.

//Every turn, when we choose:

Tile is the tile we currently have

for each possible space

for each orientation

test if legal

Legal: calculate points it could get us, save data, possibleMoves++

illegal: discard

Search for highest point move (in a tie, doesn’t matter), use that, update data

or, if (possibleMoves == 0), send appropriate message

Clear move array, possibleMoves = 0;,

repeat

NOTE: some other algorithms, such as paranoid and max n, reduce to this at 2 players

My recommendation: get first legal working as a backup.  Then, use maximin without forethought (look only at the current turn).  Calculating all possible games gets too costly too quickly.