狼人杀Room Structure

* The first user that connects to the room is the host. If he quits the second user that connects is the host, and so on so forth.
  + Create a new table “Room-Data” in database that contains information about the host, users in a room (ordered by joining the room earliest to latest). The host will be used to determine who should be able to start the game. The order of users in a room will be used to determine who should become host if the host leaves.
* Only the host is able to start the game when the room reaches its maximum number of players limit.
  + Provide a message to the client whether they are the host or guest

Server-Side

* When a user’s previous-game property is undefined, and they are attempting to join a new room, they either will create a new room or join an existing room. If they create a new room, they will write to the Room-Data table with key *Host* and value their uuid. In this case, they will also write to key *Users* (which is an array of strings) with value their uuid. Likewise, if they join an existing room, they will update the existing room’s *Host* and *Users* values.
* When the host presses start game, send a message “StartGame” via the /play websocket endpoint. Upon receiving the message, check to see if the number of players in the room is satisfied by checking the Room-Data table (optional: check to see if each player is ready, implement a ready state). If number of players is satisfied, get all *Users* value from the Room-Data table and update the *Previous-Room* property in the User-Data table of each user.
* When a user purposefully leaves a game (i.e. not merely disconnect), he is either a host or guest. First, the user will send a “LeaveGame” message. Upon receiving this message, the server will (1) call handleWSClosure and (2) use the disconnected user’s uuid to check whether the user was a host or guest by accessing the host property in the Room-Data table. If the user was a host, update the *Host* value to the next user in the *Users* array in the Room-Data table and delete the user from the *Users* array. If the user was a guest, just delete the guest from the *Users* array. Ideally these tasks should be put in a separate function perhaps called handleLeaveGame.
* When a game ends and the room is deleted, delete the room id and all its properties from the Room-Data table. Also delete the previous-room attribute for all the users.

Client-Side