# Database structure

cards

|  |  |  |
| --- | --- | --- |
| **field name** | **remarks** | **example value** |
| num | unique id of this pair of words | 1 |
| a | first word in pair | shoes |
| b | second word in pair | socks |
| language | language of the words | english/chinese |

rooms

|  |  |  |
| --- | --- | --- |
| **field name** | **remarks** | **example value/possible values** |
| roomId | unique identifier |  |
| lastUpdated |  |  |
| totalCount | total number of users in room |  |
| antiCount | number of undercover |  |
| blankCount | number of blank |  |
| hasStarted | has game started? |  |
| firstTurn | user index. first turn of current round. will be updated every round. | 0 |
| currentTurn | user index or string | 0/voting/hostVoting/ended |
| currentCount | number of users still alive |  |
| votes | array of user index who were voted out. order is not maintained. | [0,1,2,2] |
| usersWithMostVotes | array of user index who had the most votes (tie) | [0,1] |
| winner | user index or string.  if is user index, means only 1 single person wins then that person must be a blank. | 0/norm/anti/blank |
| users | array of user objects (refer to next table) |  |

rooms -> users

|  |  |  |
| --- | --- | --- |
| **field name** | **remarks** | **example value/possible values** |
| name | unique identifier |  |
| role | role in game | norm/anti/blank |
| card | given word | shoes |
| isHost | is host of game? |  |
| isOut | has been voted out? |  |
| hasVoted | has voted for current round? |  |

# How are users arranged in rooms.users array?

* every time a game is started, the rooms.users array will be shuffled and displayed in that order in the UI.

# When to end game and who wins?

* ~~when alive users > 2 and all alive users have same role, that role wins~~
* when all alive are norm, norm wins
* when alive anti > alive norm, antis win
* when alive users = 2
  + if there is blank alive, that blank wins
  + if there is anti alive, antis win

# Rules of game

1. In each game, there will be a pair of similar words.
2. At the start of the game, each player will get either of the 2 words. Most of the players (Civilians) will get the same word, while the Undercover(s) will get the other word. If there is a Blank in the game, that player will not get any word. The Blank must hide the fact that he/she does not have any word.
3. In each round, players will take turns to describe their word, without saying what the word is. Players cannot repeat what have been said before.
4. After every player has spoken, each player will vote for the player that he/she thinks is an Undercover. The player with the most votes will be out of the game. If there is a tie, the players in the tie will describe their word again (with new description) and everyone will decide again who to vote for. The host of the game will do the voting in the system.
5. At the end of each round,
   1. If only Civilians are left in the game, Civilians win.
   2. If the number of Undercovers is more than that of Civilians, Undercovers win.
   3. If there only 2 players left in the game, if there is a Blank left, that Blank wins. If there is an Undercover left, Undercovers win.