## MYSQL ChineseCheckers Stored Procedures

ALL STORED PROCEDURES FOLLOW THE FORM <heading>.<procedure name>

For example: GU.userJoinGame(1, 3);

IF A PROCEDURE IS RETURNING A VALUE, INCLUDE A @ SYMBOL WITH A VARIABLE NAME AFTER THE PARAMETERS FOR A RETURN STATEMENT.

For example: users.createUser(@id); SELECT @id;

## Users

createUser () returns userID of new user created

createAI () returns userID of new AI created

getAi (userID) returns isAi field of users

getUserID (username) returns userID of username

getUsername (userID) returns username of userID

setUsername (String newName, userID) updates userID with new username

deleteUser Designed to be used on human users, replaces user with AI in every game they are currently in

## Games

createGame(numberOfPlayers, createdTime) creates a new game

getCreated (gameID) returns DATETIME game was created at

getCurrentTurn (gameID) returns INT what current turn of the game is

getNotReadyList () returns list of all games where isReady is 0

getNumPlayer (gameID) returns number of players allowed for a game

getReady (gameID) returns isReady for a gameID

getWinner (gameID) returns winner of a game, NULL if no winner set

checkWinner (userID, gameID) returns a BOOL representing whether a player has the win condition for the game. 1 = true, 0 = false.

setWinner (userID, gameID) sets winner of a game

nextTurn (gameID) iterates currentTurn according to number of players

## GamesUsers (GU)

getGameID (guID) returns gameID of some gamesusers record

getGuID (userID, gameID) returns gamesusersID

getPlayerNumber (guID) returns player number of that gamesusers session

getUsername (guID) returns username of that gamesusers session

getUserID (guID) returns userID of that gamesusers session

userJoinGame (userID, gameID) creates a new relation between user and game, automatically fills playerNumber appropriately according to how many players that game is for

userLeaveGame (userID, gameID) replaces the user with an dynamically created AI, checks to see if all other players are AI. If so the game is deleted, also automatically clears TABLE users of any unassociated AI. Once a game is deleted, pieces of the associated game are also deleted.

getUsersGames (userID) returns a list of gameIDs the userID belongs to

## Pieces

createPiece(guID, Row, Index) returns pieceID of the piece created

getPieceID (guID, Row, Index) returns pieceID

updatePiece (guID, newRow, newIndex) updates the row and index of piece in question

getGamePieces(gameID) returns list of playerNumber, Row, Index of pieces belonging to the gameID.