

Network Protocol Documentation

Team Members: Matthew Ramos, Jordan Walker

Table of Contents

[How the Game Protocol Works](#)

[Pre-Game Protocols](#)

[Client Request Table](#)

[Server Responses Table](#)

[In-Game Protocols](#)

[Client Requests Table](#)

[Server Responses Table](#)

How the Game Protocol Works

The basic protocol used for this for this game is divided into two sections named “Pre-Game” and “In-Game”. A client is in Pre-Game when it is connected to the server but has not joined a game. A client is In-Game when they have selected a game from the game list and joined that game. All communications between the client and the game server use a single TCP socket and packets which end in “\r\n\r\n”.

The full range of possible communications between the client and server are outlined below, but a typical conversation between our client and server pregame might look something like this:

Actor	Packet Contents
Client	USER CHECKNAME\r\n ExampleUsername\r\n \r\n
Server	1.0 USER CHECKNAME, OK\r\n AVAILABLE\r\n \r\n
Client	SETUP RSA\r\n \r\n
Server	1.0 SETUP RSA, OK\r\n [PublicKeyKere]\r\n \r\n
Client	USER CREATE\r\n ExampleUsername\r\n [RSAEncryptedPassword]\r\n \r\n
Server	1.0 SETUP USER CREATE, OK\r\n SUCCESS\r\n \r\n

Pre-Game Protocols

Client Request Table

Requests clients can make before they join a game

Command Name	Client Commands	Description
Version Check	UTILITIES VERSIONCHECK\r\n\r\n	Asks the server which version of the protocol is being used
RSA Check	UTILITIES RSA\r\n\r\n	Asks the server for an RSA public key so the client can send sensitive data to the server.
Username Check	USER\r\n[Username]\r\n\r\n	Asks the server if a given username is available for the taking.
Create User (Register)	USER\r\n[Username]\r\n[RSAPassword]\r\n\r\n	Asks the server to create a new user on its database with a given username and password. Password is encrypted with RSA public key provided by server.
Authenticate User	USER AUTHENTICATE\r\n[Username]\r\n[RSAPassword]\r\n\r\n	Asks the server to authenticate (sign in) a user account with a given username and password. Password is encrypted with RSA public key provided by server.
Delete User (Unregister)	USER DELETE\r\n[Username]\r\n[RSAPassword]\r\n\r\n	Ask the server to delete a user account with a given username and password. Password is encrypted with RSA public key provided by server.
Display (List Games)	GAME LIST\r\n\r\n	Asks the server for a list of available games
Create Game	GAME CREATE\r\n[Name]\r\n[RSAPassword]\r\n\r\n	Asks the server to create a new game with a given name and optional password. Password is encrypted with RSA public key provided by server.
Join Game	GAME JOIN\r\n[GameID]\r\n[Password]\r\n\r\n	Asks the server to join the client to a game with a given ID and optional password. Password is encrypted with RSA public key provided by server.

Server Responses Table

Responses the server can have to client requests. All responses from the server in this table start with:

[Server Version Number] [Client Request], [Request Status]\r\n

Response Name	Server Response	Description
Username Action	USERNAME TAKEN\r\n\r\n USERNAME AVAILABLE\r\n\r\n USERNAME INVALID\r\n\r\n SUCCESS\r\n\r\n AUTHENTICATION FAILED\r\n\r\n	TAKEN: Tells the client that a the requested username has already been taken. AVAILABLE: Tells the client that the requested username is available. Invalid: Lets the client know that the requested username is not valid. SUCCESS: Lets the client know that either a new account has successfully been created or user has been successfully logged in the server AUTHENTICATION FAILED: Lets the client know that a user does not exist or that a password is incorrect.
RSA Reply	PUBLIC KEY\r\n\r\n	Lets the client know that a requested utility is RSA, and setups a RSA for the current client and gives them a public key
Game List	GAME LIST\r\n[ListOfGames]\r\n\r\n	Lets the client the list of games available and what to display.
Game Create	GAME CREATE\r\n\r\n SUCCESS\r\n\r\n	Lets the client know that a game has been created
Game Join	JOIN AUTHENTICATION REQUIRED\r\n\r\n JOIN INVALID GAMEID\r\n\r\n JOIN INVALID GAME\r\n\r\n JOIN GAME FULL\r\n\r\n JOIN INCORRECT PASSWORD\r\n\r\n SUCCESS\r\n\r\n\r\n JOIN OK\r\n\r\n\r\n JOIN BAD REQUEST\r\n\r\n\r\n	AUTHENTICATION REQUIRED: Lets the client know that authentication is required to join a game. INVALID GAMEID: Lets the client know that the provided game ID is not valid. INVALID GAME: Lets the client know that the game does not exist. GAME FULL: Lets the client know that a game is full and no more players can join. INCORRECT PASSWORD: Lets the client know that the password given is incorrect. SUCCESS: Lets the client know that the user has successfully joined a game. OK: Lets the client know that the request is okay. BAD REQUEST: Lets the client know that the response is a bogus request.

In-Game Protocols

Client Requests Table

Requests clients can make once they are in a lobby or are playing a game.

Command Name	Client Commands	Description
Ping	PING\r\n\r\n	Pings the game server. The server responds with the first thing in the queue of messages to send to the client
Game Vote	GAME VOTE\r\n\r\n	Vote to start the current game while user is waiting in the game lobby
Game Exit	GAME EXIT\r\n\r\n	Exits the current lobby or game. Server will inform other users in the game or lobby and gracefully inform them
Game Move	GAME MOVE\r\n[TileNumber]\r\n\r\n\r\n	Asks the server to make a requested move

Server Responses Table

Responses the server can send the client once the client is in a lobby or playing a game. All of these are responses can be added to a client's personal queue on the server and send to the client when the client makes a ping request.

Response Name	Server Response	Description
Player Leave	PLAYER LEAVE\r\n[PlayerName]\r\n\r\n	Lets the other clients know that a player has left the game
Start Game	GAME START\r\n\r\n	Lets the clients know that a New game has started
List Players	PLAYER LIST\r\n[PlayerName]\r\n[PlayerName]\r\n...\r\n\r\n	Displays a list of all player names
Add Player	PLAYER ADDED\r\n[PlayerName]\r\n\r\n	Adds a player to the selected game lobby room
Player X	PLAYERX\r\n[PlayerName]\r\n\r\n	Sent at the start of a game. Let's all clients know which player has been selected to play 'X', From that it can be assumed which client is going to play 'O'
Start Vote	START VOTE\r\n\r\n	Player has casted their vote to start the game
Pong	PONG\r\n\r\n	The server has nothing to say to the client. Only sent when a player sends "Ping"
Game Move	MOVE [Tile]\r\n\r\n	Lets the client know another client has made a move
Game Finish	GAME FINISH\r\n[FinishState]\r\n\r\n	Lets the client know the game is finished and tells the client the state the game finished in (X win, O win, cats).