Space Game Documentation:

Game server location: **wss://spacegame.io:443**

All game communication is performed via TLS websockets. There are many libraries for websockets in many different languages, pick your favorite!

The game consists of players on a map. The map is divided into sectors in the following way:

Y\X	0	1	2	3	5
0					
1	A				B
2					
3					
4			C		

Player A is in sector (0,1)
Player B is in sector (5,1)
Player C is in sector (2,4)

Each sector is of size 1000 x 1000, with 0,0 in the upper left corner:

0,0					
				800,200	
	200,800				
					1000,1000

All game protocol messages are byte level messages and have the following header structure (all integers in big endian):

Header:

Name	Type / Length	Notes
Sig	char[4]	Protocol signature, always “SPGM” for SPACE GAME
type	uint16_t	Type of message (see below)
padding	char[2]	Padding for byte alignment, set to 0
length	uint32_t	Length of the rest of the message

There are 7 different message types:

Name	Number
SIGN_UP	0
SIGN_UP_RESP	1
CONTROL	2
PLAYER_LOC	3
PING	4
PONG	6
ERROR	7

After opening a websocket connection with the server, you must send a SIGN_UP message, after which you will receive a SIGN_UP_RESP message, and the server will begin to send you PLAYER_LOC messages at a rate of ~20 Hz. You can send CONTROL messages to the server at a rate of ~20 Hz.

SIGN_UP message format:

Name	Type / Length	Notes
r	uint8_t	Red part of your player color
g	uint8_t	Green part of your player color

b	uint8_t	Blue part of your player color
padding	uint8_t	Padding for message length, set to 0
name_size	uint16_t	Length of your name
name	char[name_size]	You selected name (ASCII only)

SIGN_UP_RESP message format:

Name	Type / Length	Notes
id	uint32_t	Your given player id
map_size	uint16_t	Size of the map, in sectors, square

CONTROL message format:

Name	Type / Length	Notes
num_pressed	uint16_t	Length of pressed keys array
keys	uint16_t[num_pressed]	List of pressed keys. Currently protocol only processes SDL_SCANCODE_RIGHT, SDL_SCANCODE_LEFT, SDL_SCANCODE_UP, and SDL_SCANCODE_DOWN

PLAYER_LOC message format:

Name	Type / Length	Notes
num_players	uint16_t	The number of player structures in this message
players	player[num_players]	Array of player structures (see below)

PLAYER structure:

Name	Type / Length	Notes
id	uint32_t	This player’s id
name_part	char[17]	The first up to 17 characters of the player's name, including a null

		terminator. If player name is longer than 17 characters, extra character are stored after ENTIRE player array
r	uint8_t	Red part of player color
g	uint8_t	Green part of player color
b	uint8_t	Blue part of player color
sec_x	uint16_t	What sector x coord this player is in
padding	char[2]	padding, set to 0
sec_y	uint16_t	What sector y coord this player is in
padding	char[2]	padding, set to 0
map_x	double	Player x coord within their sector
map_y	double	Player y coord within their sector
map_vx	double	Player x velocity within their sector
map_vy	double	Player y velocity within their sector