**Protocol Abalone**

# **Core Protocol: Client**

These are messages which are sent by the client to the server. The server should handle these incoming messages. The overall format of messaging the server is the following:

**COMMAND;**Argument1**;**Argument2**;** …**;**Argumentn

If the command is MOVE, the format will be following:

**MOVE;**direction**;**marblePos1**,**marblePos2**,**marblePos3

Command: The command/method name to communicate to the server.

Argument: Arguments will be the Objects you with certain type which you want to send to the server and back. All arguments will be split up by String.split(), so you will have the Command and the arguments split into a String-array, so if the server or client is expecting integer numbers, the String number has to be converted.(int i = Integer.parseInt(myString))

Every command has its own numbers of arguments (so the array isn’t static and the number of arguments can change).

If **command** is **MOVE**:

The arguments are now specific set for moving marbles. The first argument of the protocol will be the direction-String. The following argument will select the marbles separated by commas instead of semicolons, so the marbles are saved in an own array. Each number chosen by the position of the marble. **IMPORTANT:** The array isn’t static, the numbers of the marbles can change, depending on how many marbles the player wants to move. Therefore if a player wants to move marbles, every server needs a communication tool like the switch statement in the MOVE command, which checks how many marbles are entered by the player.

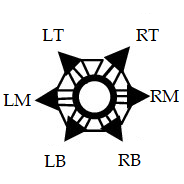
For Example:

Player wants to only move two marbles into the direction Right-Top, marble1 on field 42 and marble2 on field 50. So the message to the server should be the following:

**MOVE**;RT;42,50

The Server would have to distinguish how many marbles are selected by calculating the Marble.length = numberOfMarbles.

The delimiter in this case is the space between the words. The Arguments are listed in an Array. The command can also have none arguments.

**How to read a move and create a direction-String:**

We implemented directions of type String to distinguish where to place a marble next. The first char in a direction-string is always the horizontal direction: **L** for **left**, **R** for **right**.

The second char in the String should indicate the vertical direction: **T** for **top**, **M** for **middle**(no vertical movement), **B** for **bottom**.

For example, if you want to move to the left bottom you’d have the direction-String ‘LB’.

## 

## 

## 

## **Client-to-server commands**

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Arguments** | **Description** | **Example** |
| 1.CREATE | **String** playerName  **int** numPlayers | A player creates a lobby of 2-4 players and automatically joins that lobby himself/herself. | CREATE;Thomas;2 |
| 2.JOIN | **String** playerName  **int** numPlayers | A player indicates his/her name and how many players his/her intended lobby will consist of. If there’s no lobby with the corresponding number of players, the player will be assigned to a random lobby. | JOIN;Jorge;3 |
| 3.READY | **String** playerName | A player signals the server he’s ready to start the game. If every player is ready, the game can begin. | READY;Thomas |
| 4.MOVE | **String** direction  **int** marble1  **int** marble2  **int** marble3 | The players direction is the first index, the next indexes are the position of the marbles he wants to move. (Notice that the number will be saved in an inner String Array) | MOVE;RT;53,54,55 |
| 5.EXIT | **String** playerName | A player has finished the game and wishes to exit the lobby. | EXIT;Thomas |
| \*RECONNECT | **String** playerName | A player has been disconnected during the game and now wishes to reconnect to continue playing | RECONNECT;Ben |
| CHAT | String playerName  String message | A player sends a message to the server, which then sends it to all other players in the lobby. The playerName is the player who sent the message | CHAT;Ben;Hey guys! |

## 

## 

## 

## 

## 

## **Server-to-client commands**

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Arguments** | **Description** | **Example** |
| ERROR | **String** playerName  **String** errorMessage | This is sent to the client when the client has sent an invalid command to the server. | ERROR;Jorge;Out of Bounce |
| 1.LOBBY | **String** playerNamesn  **Int** playersAmount | The server confirms the client has successfully entered the lobby, and lists the other players that are in the lobby. Note this command will be sent every time a client enters or leaves the lobby. (There are multiple names possible) | LOBBY;Jorge,Thomas;3  (In this Case, the two players are still waiting for a third person to play with.) |
| 2.START | **String**[] playerName | The game has started with the players listed. Players take turns in the order listed. Note: a game should only be started when the lobby is full and all players in the lobby have sent the READY command. |  |
| \*WARNING | **String** playerName  **String** warningMessage | A player makes an invalid moved, a warning is displayed that discourages that player from making more invalid move attempts. A player is given another chance of making a valid move. | WARNING;Thomas;Warning! |
| \*RANDOM | **String** playerName  **String** direction  **int** marblePosition | A player has made 2 invalid moves in a row which results in a computer making a random move on the player’s behalf. | RANDOM;Thomas;RT;30 |
| \*UPDATE | **String** playerName  **String** direction  **int** marble1  **int** marble2  **int** marble3 | This indicates the player has finished his turn. When the player made a move, this will be specified by the *marble* and *field* arguments. The *token* will be passed to the next player. | UPDATE;Thomas;LB;2,3; |
| 3.SCORE | **String** winnerName  If 4 Players:  String winnerName2 | The game is finished, displays results | SCORE;Thomas |
| \*DISCONNECTED | **String** playerName | A player has disconnected. If it happened in the middle of a game | DISCONNECTED;BEN |
| CHAT | String playerName  String message | Send a message coming from playerName to the client | CHAT;Ben;Hey guys! |

## 

## **Order of commands**

1. Client sends CREATE to server to create lobby
2. Other clients send JOIN to the server to join the lobby
3. Everytime a client joins, the server sends LOBBY to all clients
4. Every client sends READY to the server
5. If every client is ready, the server sends START to every client and the game begins
6. Every client has to send MOVE throughout the hole game when he wants to set Marbles
7. After each move, the server UPDATEs every client
8. If a client won, the server sends SCORE to every client to let everyone know who won
9. If a client decides to leave the lobby, he sends EXIT to the server. Afterwards the server has to update every client with sending LOBBY

The missing commands are “optional” and depending on what happens ingame, but **must** be included!

## **Command arguments**

playerName

The display name of the player. Allowed characters: [A-Z,a-z,0-9], so:

* letters and capital letters
* digits

The definition of a field, consisting of a number.

errorCode

An integer value describing the error type. This must be a value described in the *error codes* section. A value of 0 must be used when this function is not implemented.

errorMessage

A textual representation of the error, intended to be displayed to the user.

points

An integer value greater than or equal to 0.

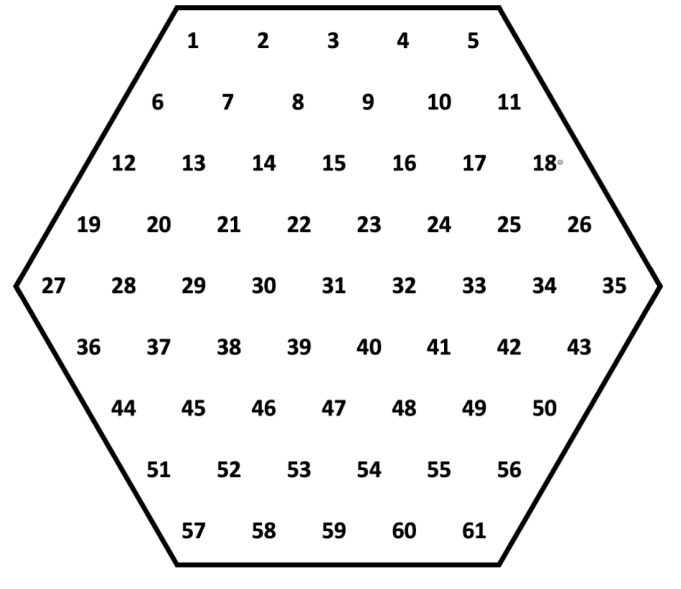
## **Marbles (identification, positioning)**

There are a total of 4 letters available to diversify the tiles of different (up to 4) players in the game: a, b, c, d. Depending on the marble’s position, there may be up to 6 possible directions available to move the marble. These positions are: (Top left (LT), Top right (RT), Left middle (LM), Right middle (RM), Bottom left (LB), Bottom right (RB)).

If a player is idle and does not make a move within 60 seconds, a random move is made for him.

Board: consists of 61 positions marked by numbers from 0 to 60.

**In the picture below, change number of 1-61 to 0-60.**



In case you wish to use extentions, contacts us.

## **Error codes**

* 0: Generic / unknown error.
* 1: Unrecognized command.
* 2: Invalid arguments.
* 3: Invalid player name.
* 4: Player name already taken.
* 5: Not your turn.
* 6: Invalid move.
* 7: Not in a lobby.
* 8: Not in a game.
* 9: Unsupported extension