

1 The protocol

1.1 Introduction

This protocol concerns the communication between the server and the clients. We use socket communication in the formats defined below.

All communication is sent either by the controllermodule (on behalf of either the boardmodule or the controllermodule itself) to a player- or viewermodule, or from a player- or viewermodule to the controllermodule (addressing either the boardmodule or the controllermodule itself).

1.2 Message format

All messages have the same format: a string consisting of a header and an optional data part, separated by a single space.

The header consists of three characters. The first represents the sender of the message, the second contains the addressee of the message, and the third the type of the message.

The characters used for the sender and the addressee are defined in section 1.3, and the message types are defined in section 1.4

1.3 Codes for the addressee and sender

- B for the board unit
- C for the controller unit
- D for the dolphin player
- E for the dolphin viewer
- F for the fox player
- G for the fox viewer
- P for a new player
- Q for a new viewer

1.4 Message types

1.4.1 B

Type: Board update — server to viewer

Data: The board, see section 1.5 for details of the encoding.

Sent on behalf of: B
Sent to: E or G

Example: “BEB WWWDWLFGWLLGDWDW”, if board size would be 4
Description: The board module sends a board update to the dolphin viewer.

1.4.2 F

Type: Board update — server to viewer
Data: The move, sent by the player, has failed.
Sent on behalf of: B
Sent to: E or G

Example: “BEF”
Description: The board module tells the dolphin viewer to display a “fail!” message.

1.4.3 J

Type: Request to join the game — player to server
Data: None
Sent by: P
Addressee: C

Example: “PCJ”
Description: This message is sent right after a new client connects to the controller.

1.4.4 M

Type: Move unit — player to server
Data: Two coordinates, both written as two integers separated by a single space.
Sent by: D or F
Addressee: B

Example “DBM 4 1 4 2”
Description: The dolphin player asks the board to move the unit at the first coordinate [(4,1) in the example] to the second coordinate [(4,2) in the example], if such a unit exists.

1.4.5 N

Type: Next turn — server to player
Data: None
Sent on behalf of: B

Sent to: (D and E) or (F and G)

Example: “BDN”

Description: The board announces that the dolphin player’s previous move has been processed and that the dolphin player may make another step.

1.4.6 R

Type: View refreshment request — player to server

Data: None

Sent by: D or F

Addressee: B

Example: “DBR”

Description: The dolphin player unit asks the board module to send an updated board to the dolphin viewer.

1.4.7 S

Type: Game end — server to player

Data: None

Sent on behalf of: C

Addressee: D or

Example: “FBS”

Description: The controller tells the fox player that the game is over.

1.4.8 T

Type: Announce the team — server to (player and viewer)

Data: The team the player will play (“D” for dolphin or “F” for fox), or “R” for rejected if the player can not join the game.

Sent on behalf of: C

Sent to: P and Q

Example: “CPT D”

Description: This message assigns the dolphin team to a joining player.

1.5 Message format of the board

In the board update message, the complete board is transmitted to the viewers. In this message the board is encoded as a string where every character represents a tile. The string is of length n^2 , where n is the board size. The following characters are allowed:

- W: a water tile
- L: a land tile
- D: a water tile with a dolphin
- E: a land tile with a dolphin
- F: a land tile with a fox
- G: a water tile with a fox

1.6 Role of the controller

The controller is the part in the server responsible for the communication with the clients. It receives the messages from the client and forward them to the board module if necessary. The controller also sends the messages from the board to the clients.

When the game is starting, the controller opens the sockets for the client to connect to and it handles the connecting clients by assigning them a team, or rejecting them if for example more then two players try to join.