

## RType Communication Protocol

Status of this Memo (58 lines per page)

This memo provides informations for the RType communication protocol. Disribution of this memo is unlimited.

Copyright Notice

Copyright (C) RType (2020). All Rights Reserved.

### Table of Contents

1.	Introduction.....
1.1	Purpose.....
1.2	Requierments.....
1.3	Terminologie.....
1.4	Overall Operation.....
2.	Notation Convention and Generic Grammar.
2.1	Command.....
2.2	Arguments.....
3.	Serialisation.....
4.	Message.....
4.1	Move.....
4.2	Attack.....
4.3	Die.....
4.4	Take bonus.....
4.5	Connexion.....
6.	Author Address.....

## RType Communication Protocol

### 1. Introduction

#### 1.1 Purpose

This request for Comments provides informations about the communication protocol for developping the Rtype video game.You will find a description of the protocol and an explanation of its use.

#### 1.2 Requierments

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [34].

An implementation is not compliant if it fails to satisfy one or more of the MUST or REQUIRED level requirements for the protocols it implements. An implementation that satisfies all the MUST or REQUIRED level and all the SHOULD level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the MUST level requirements but not all the SHOULD level requirements for its protocols is said to be "conditionally compliant."

#### 1.3 Terminology

This specification uses a number of terms to refer to the roles played by participants in the RType communication.

connection

A transport layer virtual circuit established between two programs for the purpose of communication.

## message

The basic unit of this communication protocol, consisting of a structure octets matching the syntax defined in section 4 and transmitted via the UDP protocol, client <-> server.

### 1.4 Overall Operation

This protocol behavior is based on message, the client and the server can send and receive message on both side. see section 4 for the message list.

RType

Category: Communication Protocol

H. Allabert

March 2020

## RType Communication Protocol

### 2. Notation Convention and Generic Grammar

Each message has an size of 512 octets

#### 2.1 Message

Message are binary suit of octet that means all message MUST be serialised before being send. Message MUST be formated like this:

OPCODE SPACE \$ARG1 \$ARG2

ex: 0x01 playerId \$touchNumm\$push&release

#### 2.2 lrgument

Arguments are OPTIONAL, some messages can have arg. see section 4 for more informations about message.

### 3. Serialisation

This communication is a binary based protocol, all message and command sent MUST be serialised before communication,

#### 3.1 Structure serialisation

```
{  
    char opcode;  
    vector<string> args  
}
```

### 4. Message

#### 4.1 Move

0x01 \$playerId \$x \$y

#### 4.2 Attack

0x02 \$playerId \$x \$y

#### 4.3 Die

0x03 \$playerId

#### 4.4 Take bonus

0x04 \$playerId \$bonus

#### 4.5 Connexion

0x05 \$data

### 6. Author Address

Hugo Allabert  
40 Boulevard de la Marquette  
31000 Toulouse  
hugo.allabert@epitech.eu