R-Type

Request for comment : Draft revision 1

Ludovic Alarcon

Epitech

November 2014

R-Type Communication

1. Introduction

This document describes the communication protocol between clients and server in our R-Type.

The communication uses UDP protocol communication described in [RFC0768] and TCP protocol communication described in [RFC0793].

1.1. Conventions used in this document

The key words « MUST », « MUST NOT », « REQUIERED », « SHALL », « SHALL NOT »,

« SHOULD », « SHOULD NOT », « RECOMMENDED », « MAY » and « OPTIONNAL » in this document are to be interpreted as described in [RFC2119].

2. Start session

Server and clients SHOULD use this RFC to communicate.

First, a TCP connection will be created to start a game and then an UDP connection will be created to manage the game.

When a client connects to the server, it SHALL send a handshake.

TCP Protocol SHOULD be used to send.

The client MUST send 6 bytes to the handshake.

- 4 bytes Protocol Version (1000)

- 2 bytes : if reconnection id client / else 0

The server MUST send 3 bytes after receiving the client's handshake.

- 1 byte : 0 false / 1 true

- 2 bytes : id client / if false id = 0

As soon as the server has four clients, it starts a party. It SHALL send StartGame.

TCP Protocol SHOULD be use to send.

The server MUST notify all clients and send 11 bytes.

- 1 byte : Command : 20
- 6 bytes : 2 bytes per id client
- 4 bytes : 2 bytes id munition * 2

If the server has less than four client connected, it SHALL NOT start a party.

If a client disconnected in a party, the server SHALL send ClientCrash.

TCP Protocol SHOULD be used to send.

The server MUST notify all clients except the client who has crashed and send 3 bytes.

- 1 byte : Command : 21
- 2 bytes : id client

The server SHALL NOT forget the id of disconnected client in case it tries to reconnect.

3. Client

When a client does an action, it SHALL send this action to the server. We will see below the various activities it can do.

SendMove

UDP Protocol SHOULD be used to send.

The client MUST send 9 bytes.

- 1 byte : Command : 1
- 4 bytes : 2 bytes posX, 2 bytes posY
- 4 bytes : 2 bytes directionX, 2 bytes directionY

The server MUST answer to all clients and send 11 bytes.

- 1 byte : Command : 1
- 2 bytes : id player
- 4 bytes : 2 bytes posX, 2 bytes posY
- 4 bytes : 2 bytes directionX, 2 bytes directionY

SendShoot

UDP Protocol SHOULD be use to send.

The client MUST send 12 bytes.

- 1 byte : Command : 2
- 1 byte : Enum weapon
- 2 bytes : id munition
- 4 bytes : 2 bytes posX, 2 bytes posY
- 4 bytes : 2 bytes directionX, 2 bytes directionY

The server MUST answer to all clients and send 11 bytes.

- 1 byte : Command : 2
- 1 byte : Enum weapon
- 4 bytes : 2 bytes posX, 2 bytes posY
- 4 bytes : 2 bytes directionX, 2 bytes directionY
- 2 bytes : id munition

The server MUST answer to the client who shoots with 3 bytes.

- 1 byte : Command : 11
- 2 bytes : id new munition

SendCollision

UDP Protocol SHOULD be used to send.

The client MUST send 9 bytes.

- 1 byte : Command : 3
- 4 bytes : 2 bytes id, 2 bytes id
- 4 bytes : 2 bytes posX, 2 bytes posY

The server MUST answer to all clients and send 9 bytes.

- 1 byte : Command : 3
- 4 bytes : 2 bytes id, 2 bytes id
- 4 bytes : 2 bytes posX, 2 bytes posY

If the client and the server disagree, the server is always right.

SendHitMonster

UDP Protocol SHOULD be used to send.

The server MUST notify all clients and send 7 bytes.

- 1 byte : Command : 4
- 2 bytes : id monster
- 4 bytes : 2 bytes posX, 2 bytes posY

SendKillMonster

UDP Protocol SHOULD be used to send.

The server MUST notify all clients and send 7 bytes.

- 1 byte : Command : 5
- 2 bytes : id monster
- 4 bytes : 2 bytes posX, 2 bytes posY

If the client has no answer from a monster after 5 second, the client MUST destroy it.

The client SHOULD NOT cheat because the server calculates for the monsters. If the client and the server disagree, the server is always right.

4. Server

The server manages all the monsters. He MUST notify all clients when a monster does an action. We will see below the various activities it can do.

SendMonsterSpawn

UDP Protocol SHOULD be used to send.

The server MUST notify all clients and send 8 bytes.

- 1 byte : Command : 6
- 1 byte : enum monster
- 2 bytes : id monster
- 4 bytes : 2 bytes posX, 2 bytes posY

SendMonsterMove

UDP Protocol SHOULD be used to send.

The server MUST notify all clients and send 13 bytes.

If a client doesn't know a monster's id, it SHOULD create and display it. It MUST NOT ignore this id.

- 1 byte : Command : 7

- 2 bytes : id monster

- 4 bytes : 2 bytes posX, 2 bytes posY

- 4 bytes : 2 bytes directionX, 2 bytes directionY - 2 bytes : orientation
SendMonsterDestroy

UDP Protocol SHOULD be used to send.

The server MUST notify all clients and send 7 bytes.

- 1 byte : Command : 8

- 2 bytes : id monster

- 4 bytes : 2 bytes posX, 2 bytes posY

SendMonsterFire

UDP Protocol SHOULD be used to send.

The server MUST notify all clients and send 12 bytes.

- 1 byte : Command : 9

- 1 byte : Enum weapon

- 2 bytes : id monster

- 4 bytes : 2 bytes posX, 2 bytes posY

- 4 bytes : 2 bytes directionX, 2 bytes directionY

SendMonsterKillPlayer

TCP Protocol MAY be used to send. Otherwise UDP Protocol SHOULD be used.

The server MUST notify all clients and send 7 bytes.

- 1 byte : Command : 10

- 2 bytes : id client

- 4 bytes : 2 bytes posX, 2 bytes posY