R-Type Group
Request for Comments: 4242
V-G : 1
Category: Video Games

U. LEVI--CESCUTTI
L. RENARD
A. BENARD
G. BRASSEUR
November 2020

Protocol documentation for R-TYPE

Status of This Memo

This document specified how the R-Type Protocol works.
Distribution of this memo is unlimited.

Abstract

R-Type use a UDP connection between a server and some clients.
This protocol is using binary communication

Table of Contents

1. Introduction

The R-Type protocol has been developed on systems UDP network protocol.
IP is described in [RFC0791]. TCP is described in [RFC0793].
UDP is described in [RFC0768].

2. Conventions Used in This Document

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 [RFC2119].

3. Message Exchange

R-Type server MUST listen for UDP connection requests on a standard given in parameters.
Client send request to the server, then a response is coming from the server.

You can send a request using the struct Response::ResponseStruct, fill the attributes with data by following the protocol below.

4. Network Protocol

4.1 Types

RoomCode: enum
```
Connection = 100,
ConnectionOk = 101,
CreateRoom = 102,
RoomCreated = 103,
JoinRoom = 104,
JoindedRoom = 105,
LeaveRoom = 106,
RoomLeaved = 107,
ClientLeave = 108,
DeleteRoom = 109,
Ready = 110,
PlayerReady = 111,
GameStart = 112,
Ping = 113,
WaitList = 114,
AddRoom = 115
```

GameCode: enum
```
Input = 200,
CreateEntity = 210,
SetComponent = 211,
DeleteEntity = 220,
DeleteComponent = 221
```

4.2 Packet

Connection:
int code = RoomCode::Connection

Client ping server to know is port

The code MUST be provided.
The others parameters SHOULD NOT be provided.

ConnectionOk:
int code = RoomCode::ConnectionOk
int port;

Response of Connection code with attribute port with client port

The code MUST be provided.
The port MUST be provided.
The others parameters SHOULD NOT be provided.

CreateRoom:
int code = RoomCode::CreateRoom

Client ping server create a Room

The code MUST be provided.
The others parameters SHOULD NOT be provided.

RoomCreated:
int code = RoomCode::RoomCreated
int roomId;

Response of CreateRoom code with attribute roomId set with a value

The code MUST be provided.
The roomId MUST be provided.
The others parameters SHOULD NOT be provided.

JoinRoom:
int code = RoomCode::JoinRoom
int roomId;

Client ping server to join a room

The code MUST be provided.
The roomId MUST be provided.
The others parameters SHOULD NOT be provided.

JoindedRoom:
int code = RoomCode::JoindedRoom

Response of JoinRoom code

The code MUST be provided.
The others parameters SHOULD NOT be provided.

LeaveRoom:
int code = RoomCode::LeaveRoom
int roomId;

Client ping server to leave a room

The code MUST be provided.
The roomId MUST be provided.
The others parameters SHOULD NOT be provided.

RoomLeaved:
int code = RoomCode::RoomLeaved

Response of LeaveRoom code

The code MUST be provided.
The others parameters SHOULD NOT be provided.

ClientLeave:
int code = RoomCode::ClientLeave
int port;

Server send to the others client that someone leave

The code MUST be provided.
The port MUST be provided.
The others parameters SHOULD NOT be provided.

DeleteRoom:
int code = RoomCode::DeleteRoom

Server delete send response to the client to delete his local room

The code MUST be provided.
The others parameters SHOULD NOT be provided.

Ready:
int code = RoomCode::Ready
int roomId;

Client ping server to say he is ready to play

The code MUST be provided.
The roomId MUST be provided.
The others parameters SHOULD NOT be provided.

PlayerReady:
```
int code = RoomCode::PlayerReady
int port;

Inform other client that someone is Ready

The code MUST be provided.
The port MUST be provided.
The others parameters SHOULD NOT be provided.
```

GameStart:
```
int code = RoomCode::GameStart

Inform client that the game is starting

The code MUST be provided.
The others parameters SHOULD NOT be provided.
```

Ping:
```
int code = RoomCode::Ping

Ping server to say that the client is running

The code MUST be provided.
The others parameters SHOULD NOT be provided.
```

WaitList:
```
int code = RoomCode::WaitList

Client ping server to request room List

The code MUST be provided.
The others parameters SHOULD NOT be provided.
```

AddRoom:
```
int code = RoomCode::AddRoom
int roomId;
int nbP;

Server send room to be add by the client

The code MUST be provided.
The others parameters SHOULD NOT be provided.
```

Input:
```
int code = RoomCode::Input
char event[64];

Client send event to the server

The code MUST be provided.
The event MAY be provided.
The others parameters SHOULD NOT be provided.
```

CreateEntity:
int code = RoomCode::CreateEntity

Server request creation of an entity in the client

The code MUST be provided.
The others parameters SHOULD NOT be provided.

SetComponent:
int code = RoomCode::SetComponent

Server request modification of a component in the client

The code MUST be provided.
The others parameters SHOULD NOT be provided.

DeleteEntity:
int code = RoomCode::DeleteEntity

Server request deletion of an entity in the client

The code MUST be provided.
The others parameters SHOULD NOT be provided.

DeleteComponent:
int code = RoomCode::DeleteComponent

Server request deletion of a component in the client

The code MUST be provided.
The others parameters SHOULD NOT be provided.

5. References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981

[RFC0768] Postel, J., "User Datagram Protocol", RFC 793, 28 August 1980

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6. Author's Address

EPITECH PARIS
24 Rue Pasteur
Le Kremlin-Bicêtre
FRANCE

email: ugo.levi-cescutti@epitech.eu
lucas.renard@epitech.eu
arthur.benard@epitech.eu
gregoire.brasseur@epitech.eu