**Network Working Group M. Klein**

**Request for Comments: 2138 Epibros**

**Category: Informational R-type**

**January 18**

**RFC’S memo for r-type project**

**Status of this memo**

This memo provides information for the RFC protocol

of r-type project.

**Copyright Notice**

Copyright © The EpiBros Society (2018). All rights reserved.

**Abstract**

This document describes the RFC’s operating. The r-type project will use this

RFC protocol.

**Table of Contents**

1. Overview…………………………………………………………………………………………………………………………………………….2
2. Codes…………………………………………………………………………………………………………………………………………………….3 – 5
3. Examples…………………………………………………………………………………………………………………………………………….5 - 6

**Klein Informational [Page 2]**

**RFC EPIBROS RFC’S memo for r-type project January 18**

1. **Overview**

This document describes the RFC’s operating. This RFC is for the r-type

project.

The communication will be between a client and a server. All the data will be

used for different states. This RFC provides five states. These five states

will be defined by five codes. Codes range from 100 to 500 and will be defined

by each part.

**Klein Informational [Page 3]**

**RFC EPIBROS RFC’S memo for r-type project January 18**

1. **Codes**

The codes 100 to 199 will define the connections between server and client.

Here are all the codes:

* 101 is sent from client to server and requests a connection to this server.
* 102 is sent from server to client and allows the client to connect to this

server. The message’s format is 102 IP:PORT:ID\_PLAYER. The connection

will authorize the TCP connection.

* 103 is sent from server to client and allows the client to connect to this

server. The message’s format is 103 IP:PORT. The connection will authorize

the UDP connection.

* 104 is send from server to client and informing the client to wait for the

game to start.

* 105 is sent from server to client and is informing the client that there is

a new player in the game. The message’s format is 105 PLAYER\_ID.

* 106 is sent from server to client and informs all the clients in the same

game’s room that the game is starting.

* 107 is sent from server to client and lists all the rooms. The message’s f

format is 107 **ID\_****ROOM, PLACE\_LEFT:ID\_ROOM, PLACE\_LEFT:ID\_ROOM, PLACE\_LEFT**

**Klein Informational [Page 4]**

**RFC EPIBROS RFC’S memo for r-type project January 18**

* 108 is sent from client to server and informs the server to create a new

Room.

* 109 is sent from client to server and informs the server that the client

will joining a room. The message’s format is: **109 ID\_ROOM**

* 110 is sent from client to server and informs the server that the client

has joined a room. The message’s format is: **110 UUID (UUID is user id)**

* 111 is sent from client to server and informs the server that the client

want to launch game.

**Klein Informational [Page 5]**

**RFC EPIBROS RFC’S memo for r-type project January 18**

The codes 200 to 299 will define the positive answer’s server.

Here are all these codes:

* 201 is sent from server to client. This code informs the client that the

server has accepted the request connection.

The codes 300 to 399 will inform if there is an error, to the client.

Here are all these codes:

* 301 is sent from the server to the client. This code informs the client

that his command is unknown.

* 302 is sent from the server to the client. This code informs the client that

the room he’s tried to join doesn’t exist.

* 303 is sent from the server to the client. This code informs the client that

his command is badly formatted command.

* 304 is sent from the server to the client. This code informs the client that

his command targeting another room is invalid as he doesn’t belong to it.

**Klein Informational [Page 6]**

**RFC EPIBROS RFC’S memo for r-type project January 18**

The codes 400 to 499 will give information about the Game event to the server

and the server will reply a code to inform to the client about his new state.

Here are all these codes:

* 401 is sent from client to server. This code informs the server that the

client has moved to the top in the game.

* 402 is sent from client to server. This code informs the server that the

client has moved to the bottom in the game.

* 403 is sent from client to server. This code informs the server that the

client has moved to the left in the game.

* 404 is sent from client to server. This code informs the server that the

client has moved to the right in game.

* 405 is sent from server to client. This code informs to the client of

its new position in the game. The message’s format is: 405

**ID\_PLAYER:X:Y:SPRITE\_ID**

**Klein Informational [Page 6]**

**RFC EPIBROS RFC’S memo for r-type project January 18**

The codes 500 to 599 inform of a player event to the client.

Here are all these codes:

* 501 is sent from server to client. This code informs these client that a

client has been disconnected from the game.

1. **Examples**

The following examples are between a server and a client. The “=>” are for the

Client’s message and “<=” are for the server’s message.

Here come the examples:

=> 101 (Connect to server)

<= 201 (Answer for the new client connection)

<= 102 IP:PORT:ID (TCP)

<= 103 IP:PORT (UDP CONNECTION FOR GAME MESSAGE)

<= 104 (Waiting Game)

<= 105 ID\_PLAYER (NEW PLAYER IN ROOM)

<= 106 (GAME START)

=> 401 (PLAYER MOVE TOP)

=> 402 (PLAYER MOVE BOTTOM)

=> 403 (PLAYER MOVE LEFT)

=> 404 (PLAYER MOVE RIGHT)

<= 405 ID:X:Y:SPRITE\_ID

<= 501 ID (DECONNEXION PLAYER)