O.G Working Group C. Benjamin
Internet-Draft H. Adrian
Expires: May 5, 2021 B. Arthur
V. Matthias
Epitech, Inc.
November 1, 2020

R-Type Protocol

Abstract

This memo presents the way of using the R-Type Protocol as the interaction between the client side and the server side for commands and reply codes. Request for Comments (RFC) series.

Status of This Memo

Copyright Notice

1. TABLE OF CONTENT

1. Introduction...2

2. Format...2

   1. Model...3

3. RTP Commands...4

   1. RTP Commands Arguments Types And Format......................3

   2. Access Control Commands.......................................4

   3. RTP Ingame Commands...5

4. RTP Menu Replies...5

5. RTP Ingame Replies...5

6. Format..5

2. INTRODUCTION

The mid ’80s was a renaissance for shooters. In 1985, Konami released Gradius. Not long later, in 1987, Irem released R-Type. These are two of the most important games in the genre. It’s completely stunning that, in its relative infancy, Irem could shape a game as ingenious as R-Type. One of the premiere side-scrolling shooters, R-Type moved a bit slower than your typical twitch games, with a meticulous pace and almost leisurely scrolling. The idea wasn’t to simply blast everything on the screen, although you could certainly try. The idea was, quite simply, to stay alive. Naturally, given the claustrophobic nature of the levels, swarming with enemies from every angle, this was never an easy task. In this project, we had to make our own version of R-Type.

3. FORMAT

```
                                           ------ client
                                       ---
                           -- game ----        -- client
                        ---                ----
                     ---               ---------- client
                   --      -- game ---
                  -     ---           ----------- client
           server ------                   ----
                  -     --                     -- client
                   --     -
                     ---   -- game -------------- client
                        ---
                           -- game -------------- client
                                     --
                                       ---------- client
                                        --
                                         -------- client
                                            --
                                               -- client
```

3.1. MODEL

Here is the diagram of the MTP model

```
                           /----------\              /----------\
                           |          |  [Commands]  |          |
                           |  Client  | <----------> |  Server  |
                           |   RTP    |   [Reply]    |   RTP    |
                           \----------/              \----------/
```

The RTP works on the principe of request/reply which is pretty simple. As we can see on the diagram, the client sends request to the server that will send back a reply depending on the request and on what he found.

3.2. RTP COMMAND ARGUMENTS TYPES AND FORMAT

The commands sent by the client sometimes need an argument. Here will be described the different types accepted:

string: <"string">

The String type is a character pointer of not defined size. Its content depends on the command before this argument type.

4. RTP COMMANDS

4.1. ACCESS CONTROL COMMANDS

These commands will be used to access or leave the server and launch a game.

These are the states which in a user can be:

```
+===========+
| - NONE    |
|           |
| - INLOBBY |
|           |
| - READY   |
|           |
| - INGAME  |
|           |
| - OFF     |
+===========+
```

NONE :

When the User is in NONE state, he can either create a game by sending code
"201 <NAME>" to the server. the <NAME> will be the name of the Lobby.
or join an existing game by sending code
"202 <NAME>" to the server.
The <NAME> is the name of the Lobby that the user wants to join.
The User can of course leave the program by sending code
"200" to the server.:
An unexpected close on the control connection will cause the server to close the user.

INLOBBY :

When the User is in INLOBBY state, he can either says that he’s ready by sending code
"203" to the server.
Or he can leaves the lobby by sending code
"204" to the server.
The User can receive the menu routine by sending code
"210" to the server.
The User can of course leave the program by sending code
"200" to the server.:

An unexpected close on the control connection will cause the server to close the user.

READY :

When the User is in READY state, he can either says that he’s not ready anymore by sending code
"205" to the server.
Or he can launch the game by sending code
"206" to the server.
Or he can leaves the lobby by sending code
"204" to the server.
The User can receive the menu routine by sending code
"210" to the server.
The User can of course leave the program by sending code
"200" to the server.:
An unexpected close on the control connection will cause the server to close the user.

INGAME :

When the User is in INGAME state, all he can do is leave by sending code
"207" to the server.
An unexpected close on the control connection will cause the server to close the user.

OFF :

When the User is in OFF state, his firts interaction with the server will set his state to "NONE".

4.2. RTP INGAME COMMANDS

/right (right arrow key) :

The /right command gives information that the user is moving to the right. This will be sent to the Server and it will change player’s position. The player’s position will be update when the Server will send back refreshed games informations to the Client. The code "101" will be sent to the Server.

/left (right arrow key) :

The /left command gives information that the user is moving to the left. This will be sent to the Server and it will change player’s position. The player’s position will be update when the Server will send back refreshed games informations to the Client.

The code "102" will be sent to the Server.

/up (up arrow key) :

The /up command gives information that the user is moving to the top. This will be sent to the Server and it will change player's position. The player's position will be update when the Server will send back refreshed games informations to the Client.
The code "103" will be sent to the Server.

/down (down arrow key) :

The /down command gives information that the user is moving to the bottom. This will be sent to the Server and it will change player's position. The player's position will be update when the Server will send back refreshed games informations to the Client.
The code "104" will be sent to the Server.

/shoot (space bar) :

The /shoot command gives information that the user is shooting. This will be sent to the Server and it will create a new ally bullet starting at player's position. The player's bullet will be update when the Server will send back refreshed games informations to the Client. The code "105" will be sent to the Server.

5. RTP MENU REPLIES

5.1. FORMAT

111
When the server will receive something from a client, it will send back the code "111" to validate that everything is OK.

121
The code "121" is the code that the Server will send to the Client when he creates or joins a lobby successfully.

131
The code "131" is the code that the Server will send to the Client when he starts a game successfully.

222
The code "222" is the basic error code for the server. When a command is invalid, the server will send back this code.

333

The server will send code "333" if a player is trying to take a room name that already exists.

444
The server will send code "444" if someone is trying to join a room which is already full.

555
The server will send code "555" is someone is trying to lauch a game when all the players aren’t ready.

777
The server will send code "777" if someone is trying to join a room which doesn’t exist.

As a routine menu, the Server will send updates of menu informations to the client. To activate the routine the client must send the code "210" to the Server. Each time this code is received, a menu routine will be send to the client.

We will use a specific format of replies that will give informations precisely. The message will be preceded by the code "110".(the reply is sent as a string with an Enum for player, an Enum for state):
110 [(enum)PLAYER] [(enum)STATE]

6. RTP INGAME REPLIES

6.1. FORMAT

111
When the server will receive something from a client, it will send back the code "111" to validate that everything is OK.

222
The code "222" is the basic error code for the server. When a command is invalid, the server will send back this code.

666
The server will send code "666" if a player is trying to do menu commands while its last game isn’t finished yet.

135
The server will send code "135" if the game is finish, in order to put the players back in the lobby.

As a routine, the Server will constantly send updates of everything in the game to the client. Every time the client sends something, the server will check the Request and treat it to execute the action. He will then return to the client a reply that will give information about every element of the game.

We will use a specific format of replies that will give informations precisely. The message will be preceded by the code "100".(the reply is sent as a string with an Enum for type, an Enum for state and a position x and y):

```
100 [(enum)TYPE] (int)Id|[(enum)ANIMATION_STATE]|[(bool)TO_UPDATE].[(
int)posX].[(int)posY]
```

Authors’ Addresses

Coulon Benjamin
Epitech, Inc.

Hoarau Adrian
Epitech, Inc.

Bertaud Arthur
Epitech, Inc.

Virassamy Matthias
Epitech, Inc.