Collaborative Raster Painting Protocol

CRPP

version 1.0

Contents

1 Introduction

1.1 License

1.2 Authors and contact

Author is Martin Indra from Moon Games group. You can write us on mg at mgn cz. Project web page is `http://crpp.mgn.cz/`.

1.3 Document isn't final

This document is under development. It can contain mistakes, antagonism or incomplete information. Take into account document development and call attention to authors for mistakes.

Text or his parts can be now in Czech language. Please be patient and wait to translation. We are sorry. If you can help us with translation we will be pleasure with us cooperation. Please call attention to language mistakes.

1.4 About document

This document is part of *CRPP* project, see `http://crpp.mgn.cz/`for more informations.

1.5 Introduction

This document describes network protocol for collaborative painting (CRPP - Collaborative Raster Painting Protocol). Description of term "collaborative painting" is in first part of the document (see 2). Next part is about server and his behavior 3. Most of the rest of document is about own protocol (see I). Final part **??** contains instances of CRPP communication.

2 Collaborative painting

Collaborative painting in this document means shared painting among many users in real-time. It is using computer network.

2.1 Concise list of features

- There are servers and each can contain opened and/or locked rooms.
- Rooms have canvases with itself names. Canvases can be created and deleted dynamically.
- Canvas is compounded by layers. Layers can be dynamically created, deleted and sorted.
- Each layer can contain any raster graphics (image).
- It is possible to either partially and fully erase layer content.

3 Server

Server associate all connected clients and it is enabling mutual communication. Server is able to request authentication by name and password or by token.

3.1 Client

Each connected client is virtually represented on server by unique ID on interval 0 to $2^{32} - 1$ and it's user name (nick). User name can be duplicate on server (only ID has to be unique).

3.2 Rooms

Server contains rooms, this rooms cam be used to mutual communication among joined users. Server is able to maintain zero to 2^{32} rooms (limitation of 4 bytes length ID). Each room has own ID and name. Room can be opened or locked by password.

Clients can communicate together just in case that there are in same room. Every client is outside of all rooms after connection to server. Single client can be only in one room simultaneously therefore client can join room just in case that it is not in other room in that time. Clients can joint (connect to), leave and create rooms. When client create new room it is automatically connected to this new room. Room exists just if there is at least one client, otherwise room is destroyed. See figure 3.1 for illustration.

3.2.1 Canvases and layers

The main target of the protocol is shared drawing. It is maintained (in rooms) by canvases and layers. Each room can contain several canvases and each canvas can contain layers. Canvas is composed by layers which user can draw into. Structure of room is illustrated by figure 3.2.

Every canvas has own dimension (in pixels), name and ID. There is no specified order of canvases it is depends just on client. Layers have particular order specified by server.

Room can contain 0 to 2^{32} (4 bytes ID) canvases and canvas is composed of 0 to 2^{32} layers.

Canvas has owns ID, order.

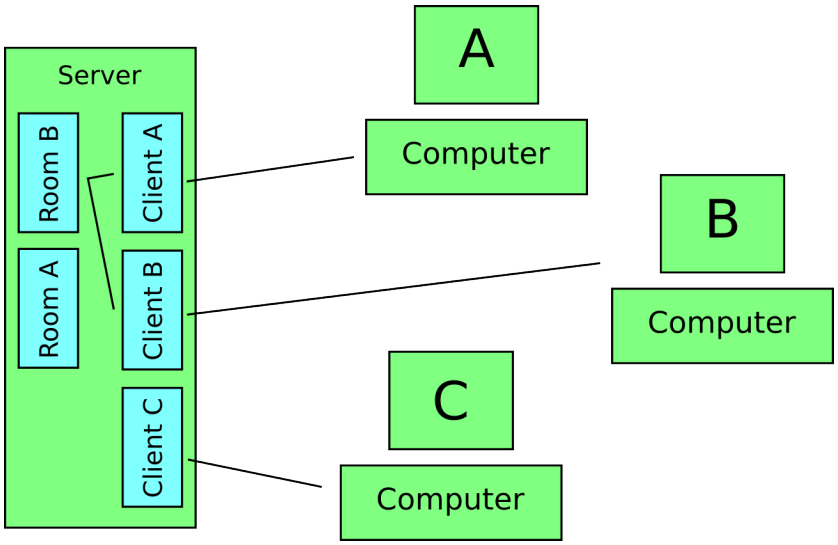

Figure 3.1: Clients, server and rooms

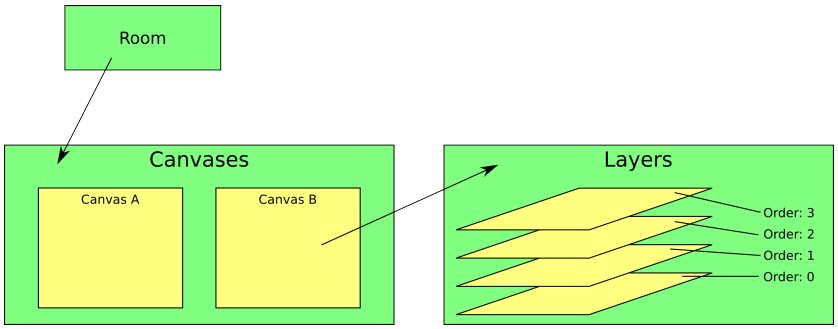

Figure 3.2: Room

Layer is picture with 32 bits color depth – 1 B is used for alpha channel (opacity).

Order of layers starts with 0 and ends with $p - 1$ where p is count of layers in canvas. Layer with order 0 is under all other layers, controversially layer with list $p - 1$ is painted over rest of layers. General rule is that order is number of layers painted under this particular layer. See figure 3.2.

3.2.2 Chat

Chat is cherished inside of rooms. Clients outside of all rooms can't chat with anybody!

3.2.3 Another services

Part I

Connection

The protocol is divided to two mutually independent layers. That structure separates connection and data transfer from protocol sense and data representation. Protocol layers are illustrated on figure 3.3.

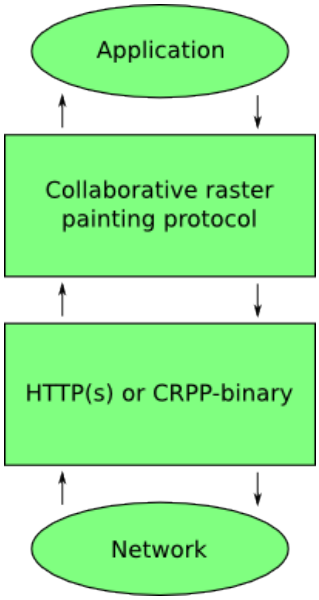

Figure 3.3: Protocol layers diagram

Top layer interchanges monolithic data chunks with bottom layer. Bottom layer deliver this data to the up layer on other side of connection. Top layer is described on III and bottom layer is described on II.

4 Data types

If it is not say otherwise all integers are unsigned.

4.1 Unsigned integers

Every bit of value 1 represents number 2^p, where p is bit position in big-endian notation. 0 bits represent 0. The value of integer is than sum of all bits. Equation **??** is specifying number value.

$$\sum_{p=0}^{n-1} b_p 2^p \tag{4.1}$$

p is bit position (the last has value 0, than value is increasing by step 1), n is count of bits representing the number, b_p is bit value(0 or 1) on position p.

For example sequence 00001010 has value 10 ($2^1 + 2^3$).

4.2 Signed integers

Hodnota znaménkového celého čísla je součet čísel a and b (it is $value = a+b$).

Hodnotu proměnné a uručje vzorec $a = -2^{n-1} \cdot f$, kde n je počet bitů reprezentujících číslo a f první bit (0 or 1).

Hodnotu promměné b určuje vzorec 4.1 přičemž první bit není do vzorce vložen. Například posloupnost 10011 by měla hodnotu $b = 2^0 + 2^1$ (není zahrnut první bit 1).

Příklad: číslo reprezentované bity 1011 má hodnotu -5 ($a = -2^3 \cdot 1 = -8$, $b = 2^1 + 2^0 = 3$, $a + b = -8 + 3 = -5$).

4.3 Floating point numbers

Real number are used in meaning of standard IEEE 754 with 32 bites.

4.4 Logical value (booleans)

Logical values (boolean) are represented by one byte, where full zeros (00000000) means *false* and value one (00000001) represents *true*. Every other value is considered as invalid.

4.5 Strings

If it is not specified otherwise text (characters string) is coded by UTF-8 (Unicode). In several cases ASCII is use instead. Protocol is case sensitive.

4.6 Image

Every image is represented as binary data of PNG `http://www.w3.org/Graphics/PNG/`. PNG image is specified by standard ISO/IEC 15948:2003 (E), see `http://www.w3.org/TR/PNG/`.

In all cased ARGB PNG is used (four 8 bites channels: alpha – transparency, red, green, blue), that is 32 deep.

Part II

Down layer

Spodní vrstva se stará o spojení, jeho udržování, obnovování a fyzickou podobu přenášní dat. Je možné použít jeden ze dvou aplikačních protokolů (ve spodní vrstvě), a to HTTP(s) (viz 6), nebo CRPP-binary (viz 5).

Spodní vrstva komunikuje s vrchní tím, že jí předává respektive dostává monolitické bloky dat. Tyto bloky jsou stejné na obou stranách spojení, viz diagram 4.1. Od okamžiku kdy bloky dat opustí vrstvu CRPP do jejich předání do této vrstvy na druhé straně spojení nesmí být změněno jejich pořadí (blok, který byl odeslán první, musí být jako první doručen). Může nastat situace, ve které je pořadí dat mezi protokoly spodní vrstvy jakkoli permutováno, ale mezi vrchními vrstvamy musí komunikace probíhat tak, jako by k žádnému prohození pořadí nedošlo.

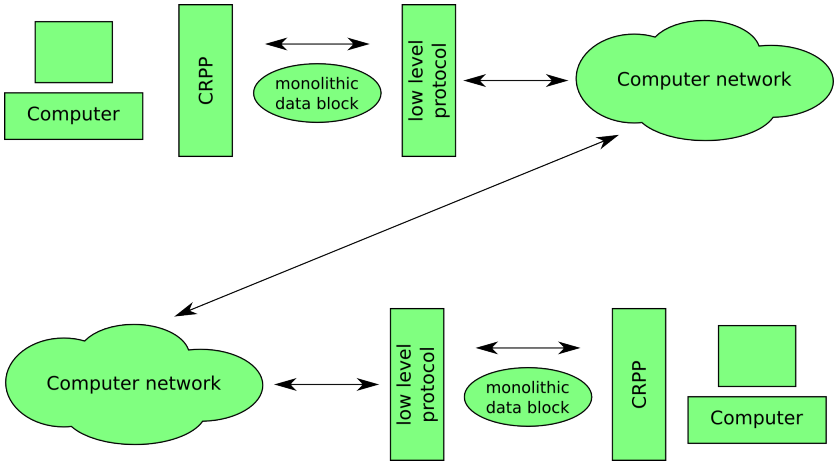

Figure 4.1: Monolithic data blocks

Veškerý úkol spodní vrstvy je brát posloupnosti bytů a ty doručovat na druhou stranu spojení ve stejním pořadí.

5 CRPP-binary

Fyzické spojení protokolu CRPP-binary je zprostředkováno pomocí protokolu *TCP/IP*.

CRPP-binary je jednoduchý protokol, který umožnuje balit a odesílat data z vrchní vrstvy. Data z vrchní vrstvy se zabalí do packů a ty se na druhé straně opět rozbalují. Strukturu packu ilustruje obrázek 5.1.

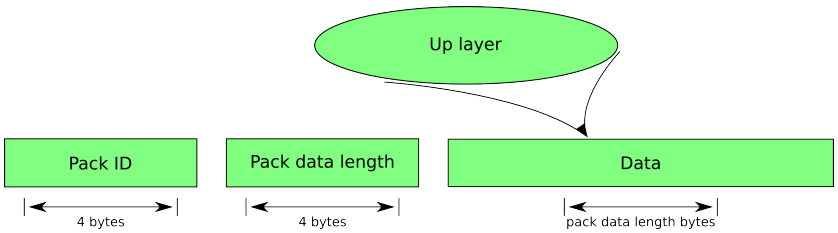

Figure 5.1: CRPP-binary pack

Pack se dělí na tři části, první dvě obsahuje hlavička packu a mají konstatní velikost. Třetí část obsahuje data packu.

První čtyři byty packu obsahují ID packu, které by se mělo inkrementálně zvětšovat a to zvlášť u každého spojení každým směrem. Takže například spojení klienta A se serverem směrem od klienta A k serveru počítá zvlášť než směr od serveru ke klientovi A. ID je uvedeno jako bezznaménkové celé číslo, které je pospáno v části 4.1.

Další čtyři byty (tedy pátý až osmý v pořadí) udávají celkovou délku přenášených dat bez hlavičky. Číslo je udáno opět podle specifikace v části 4.1.

Zbytek zprávy jsou vlastní data, která zpracovává vrchní vrstva.

6 HTTP(s)

Part III

Up layer

Vrchní vrstva přenášená data (ta která si vyměnuje se spodní vrstvou) zpracovává a to vždy po celých kusech. V kontextu vrchní vrstvy (vrstvy CRPP) se monolitické bloky dat označují jako “message”. Každá message je ucelený soubor dat (posloupnost bytů), který je možné jednotlivě zpracovat. Datová struktura a způsob zpracování message je popsáno v části 7, použité datový typy v části 4.

7 Collab message

CRPP message je posloupnost bytů. Každá message má význam příkazu (požadavku na vykonání nějaké funkce) a může obsahovat parametry.

7.1 Message

Message se skládá ze dvou částí a to z jejího příkazu a nepovinných parametrů. Příkaz reprezentují 4 ASCII znaky (4 byty). Parametry nemusí message vůbec obsahovat a může jich být neomezené množství. Jejich struktura je popsána v části 7.2.

Obsah message je také ilustrován obrázkem 7.1.

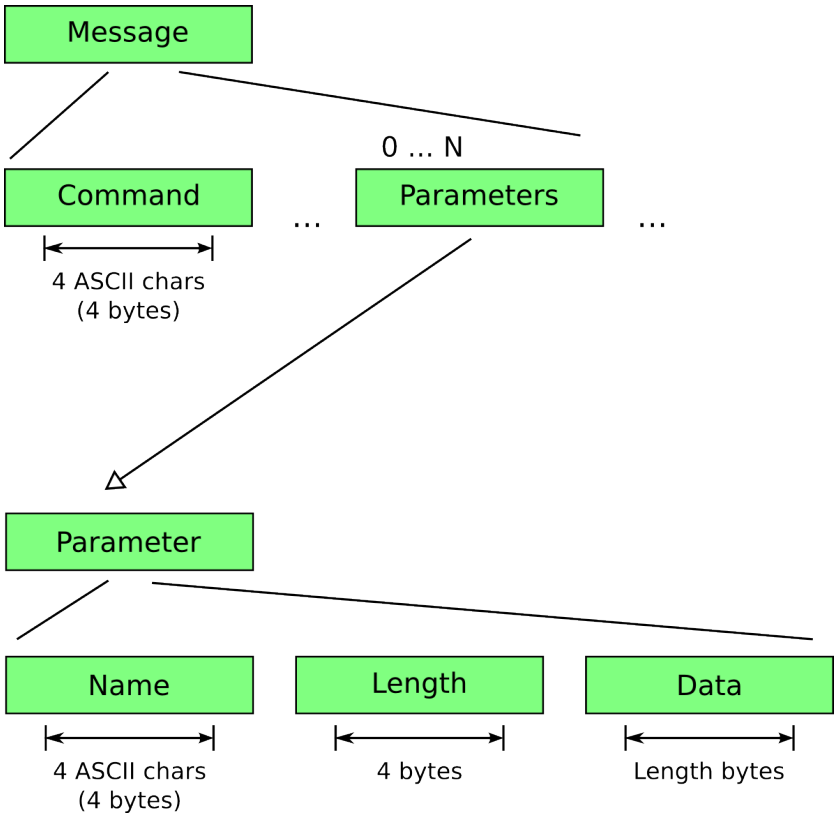

Figure 7.1: Message structure diagram

7.2 Parameter

Každý paremetr je reprezentován jako blok dat. Obsahuje tyto části:

1. paremeter name (4 B)
2. data length (4 B)
3. parameter data (data length B)

První část (paremeter name) jsou 4 ASCII znaky (4 byty), které jednoznačně identifikují parametr. Následující 4 byty udávají jako neznaménkové celé číslo délku dat (n). Dalších n bytů jsou data parametru. Data jsou interpretována různě podle toho v jakém příkazu je parametr obsažen a v jakém parametru jsou obsaženy data. Základní používané datové typy jsou posány v části 4.

V části IV jsou popsány jednotlivé příkazy, jejich parametry, význam a způsob zpracování.

Part IV

Commands

This chapter describes particular commands, their meaning and structure.

Commands are divided to three groups, it is outgoing commands (from client to server), incoming commands (from server to client) and two-way (the same command can be send either from and to server). In 8 is list of commands with brief description, detailed description is further.

8 List of commands

8.1 Outgoing commands

- GCIN – get connection info – request for connection info
- AUTN – authenticate – authenticates client
- GRLI – get rooms list – request for list of rooms on server
- CROM – create room – request for new room
- OJRO – join to room outgoing – request for connect client to particular room
- ODRO – disconnect from room outgoing – request for disconnecting client from room
- ALAY – add layer – request for new layer in a canvas
- RLAY – remove layer – request for removing layer
- SLAL – set layer location – request for moving layer to another position (affects order of layers)
- ACAN – add canvas – request for new canvas in room
- RCAN – remove canvas – request for deleting canvas in room
- HTIM – make HTTP image – request for HTTP accessible snapshot of a canvas
- OCHA – chat message outgoing – sends a chat message

8.2 Incomming commands

- SINF – server info – informations about server
- CINF – connection info – info about connection state
- SSUC – client connection success – information that some request affecting connection (e.g. authentication) has been successful

- SERR – client connection problem – information about problem with connection (e.g. with authentication)
- RLIS – rooms list – list of rooms on the server
- IJRO – join to room incoming – client has been connected to room
- IDRO – disconnect from room incoming – client has been disconnected from room
- ULIS – users list – list of clients (users) in room
- LORD – layers order – new order of layers
- SRES – set resolution – new resolution of a canvas
- ICHA – chat message incoming

8.3 Duplex commands

- SNIC – set nick – set nick of client (user)
- PANT – paint – information about particular layer update
- SLAN – set layer name

9 Paint

This commands informs both server and client about update in particular layer.

9.0.1 List of parameters

- UDTY – update type
- UDID – update ID
- LYID – layer ID
- CNID – canvas ID
- XCOR – X coordinate
- YCOR – Y coordinate
- UIMG – update image

9.0.2 Detailed description of parameters

Update type

Update type is four bytes unsigned integer determining how this update should be applied. It could has value 0 and 1, where 0 means *add* and 1 *remove*.

In case of *add* update "source over" compositing algorithm should be used (update is paint over/on current layer).

In case of *remove* update it is the alpha channel (of pixels) which is affected. Resulting alpha of every pixel D_a is counted as $D_a = S_a \cdot P_a$, where S_a is alpha in update picture (received update data) and P_a is original alpha (in layer before update). Mentioned equation considers 0 as no transparency and 1 as full transparency with linear transition (0.5 is half-transparency).

Update ID

Four bytes unsigned integer representing ID of the update.

Layer ID

Four bytes, unsigned integer identifying layer to update.

Canvas ID

Four bytes, unsigned integer identifying canvas to update (there is possibility to use several canvases simultaneously).

X coordinate

Unsigned, four bytes integer specifying X coordinate (horizontal) of left side of the update (position of rectangle).

Y coordinate

Unsigned, four bytes integer specifying Y coordinate (vertical) of the top of the update (position of rectangle).

Update image

See specification in 4.6.

9.1 Add layer

Přidá vrstvu do plátna. Obsahuje tyto bloky:

- LPOS – layer position
- LNAM – layer name
- CNID – canvas ID

Server action

Server vytvoří novou prázdnou vrstvu a zařadí jí do správné pozice. Poté odešle všem připojeným klientům nové layers order a pak name of this layer (set layer name).

Blocks

Layer position Obsahuje celočíselné čtyřbytové bezznaménkové číslo, které udává kolikátá v pořadí bude vrstva od spoda po přidání. Vrstva na pozici 0 bude pod všemi ostatními. V případě, že je číslo větší než počet všech vrstev, bude vrstva přidána na vrch.

Layer name Obsahuje text s názvem vrstvy.

Canvas ID Obsahuje celočíslené bezznaménkové čtyřbytové číslo s ID plátna, do kterého má být vrstva přidána.

Part V

Examples