

# Collaborative raster painting protocol

CRPP

version 1.0

# Contents

<b>I</b>	<b>Introduction</b>	<b>4</b>
0.1	Licence . . . . .	5
0.2	Contact . . . . .	5
0.3	Document isn't final . . . . .	5
0.4	About document . . . . .	5
0.5	Introduction . . . . .	5
<b>II</b>	<b>Collaborative painting</b>	<b>6</b>
0.6	Stručný popis požadavků . . . . .	7
<b>III</b>	<b>Server</b>	<b>8</b>
<b>1</b>	<b>Client</b>	<b>10</b>
<b>2</b>	<b>Rooms</b>	<b>11</b>
2.1	Canvases and layers . . . . .	11
2.2	Chat . . . . .	12
2.3	Another services . . . . .	12
<b>IV</b>	<b>Connection</b>	<b>13</b>
<b>3</b>	<b>Data types</b>	<b>15</b>
3.1	Unsigned integers . . . . .	15
3.2	Signed integers . . . . .	15
3.3	Floating point numbers . . . . .	15
3.4	Logical value (booleans) . . . . .	15
3.5	Strings . . . . .	16

---

<b>V</b>	<b>Down layer</b>	<b>17</b>
<b>4</b>	<b>CRPP-binary</b>	<b>19</b>
<b>5</b>	<b>HTTP(s)</b>	<b>20</b>
<b>VI</b>	<b>Up layer</b>	<b>21</b>
<b>6</b>	<b>Collab message</b>	<b>23</b>
6.1	Message . . . . .	23
6.2	Parameter . . . . .	24
<b>VII</b>	<b>Commands</b>	<b>25</b>
<b>7</b>	<b>Commands</b>	<b>26</b>
7.1	Příchozí . . . . .	26
7.2	Odchozí . . . . .	26
7.2.1	Add layer . . . . .	26
7.3	Obousměrné . . . . .	27
7.3.1	Paint . . . . .	27

# Part I

## Introduction

## 0.1 Licence

All text of this document is licenced by GNU FDL version 1.3 or greater. Current version of licence text you can find on web address <http://www.gnu.org/copyleft/fdl.html>.

## 0.2 Contact

Author of this text is Martin Indra from Moon Games group. Write us on [mg@mgn.cz](mailto:mg@mgn.cz). Project web page <http://crpp.mgn.cz/>.

## 0.3 Document isn't final

This document is in development. It can contains 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.

## 0.4 About document

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

## 0.5 Introduction

Document describes network protocol for collaborative painting (CRPP - Collaborative Raster Painting Protocol). We describing what we mean by “collaborative painting” in first part of docuemnt (see II). What its include and what not. Next part is about server and his behaviour III. And rest of document is about own protocol (see IV).

# Part II

## Collaborative painting

Kolaborativním kreslením míníme sdílené kreslení mezi mnoha uživateli v reálném čase. Sdílení by mělo probíhat v počítačové síti.

## 0.6 Stručný popis požadavků

- Je možné zakládat otevřené i zamčené místnosti.
- V každé místnosti je možné dynamicky přidávat a/i odebírat plátna. Plátna mají své jméno.
- U každého plátna lze dynamicky přidávat a/i odebírat vrstvy. Vrstvy lze pojmenovávat a měnit jejich pořadí.
- Do vrstev lze zakreslovat libovolný bitmapový obsah.
- Z vrstev je možné zcela, nebo částečně odmazávat (například pomocí gumy libovolného tvaru a krytí).

# Part III

## Server



Server sdružuje všechny klienty, kteří se na něj připojí a umožňuje případnou vzájemnou komunikaci. Server může po připojení vyžadovat autentikaci uživatele pomocí jména a hesla.

# 1 Client

Každý připojený fyzický klient má na serveru svou virtuální reprezentaci. Je reprezentován pomocí unikátního ID v rozmezí 0 až  $2^{32} - 1$  a svého uživatelského jména. Uživatelské jméno může být na serveru duplicitní, záleží na implementaci serveru, jestli tuto možnost spřístupňuje.

## 2 Rooms

Na serveru se nacházejí místnosti, v rámci nichž mezi sebou uživatelé komunikují. Na serveru nemusí být žádná a nebo až  $2^{32}$  místností (4 bytové ID). Místnosti mají své ID a jméno. Místnost může být otevřená všem, nebo zamčená heslem.

Klienti spolu na serveru mohou komunikovat jen v případě, že se nacházejí ve stejné místnosti. Každý klient se po připojení nenachází v žádné místnosti, do té se však může následně připojit. Klient se může do místností připojovat jen v případě, že se nenachází v žádné jiné. Logický důsledek je, že se klient může nacházet jen v jedné místnosti v jeden okamžik. Klienti se mohou do místností připojovat, odpojovat a místnosti vytvářet. V případě, že klient vytvoří novou místnost je do ní automaticky připojen. Místnost existuje jen po dobu co se v ní nachází alespoň jeden klient, prázdné místnosti jsou automaticky mazány. Místnosti na serveru ilustruje obrázek 2.1.

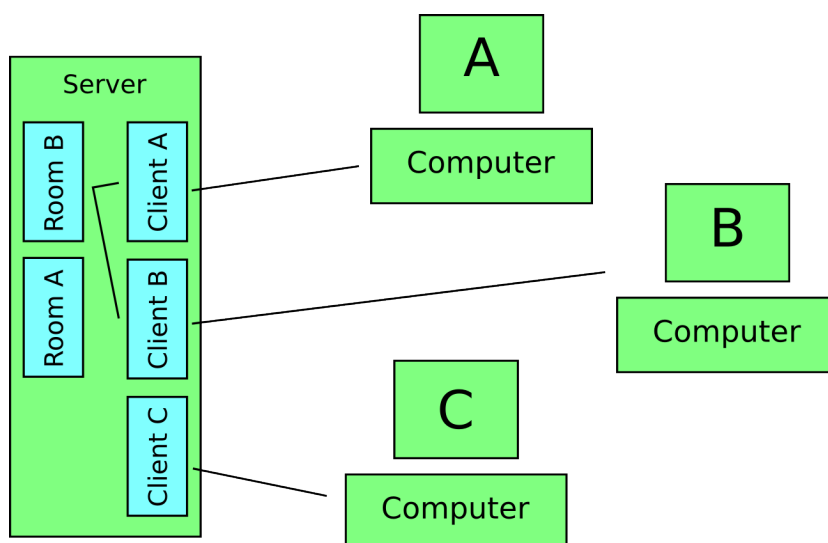


Figure 2.1: Clients, server and rooms

### 2.1 Canvases and layers

Hlavním cílem protokolu je umožnit uživatelům sdílené kreslení. To je zajišťováno v rámci místností pomocí jednotlivých plánek a vrstev. V rámci místnosti může existovat několik pláten (samostatných obrázků) v nichž jsou vrstvy

(které se překrývají) do nichž klienti kreslí. Strukturu místnosti ilustruje obrázek 2.2.

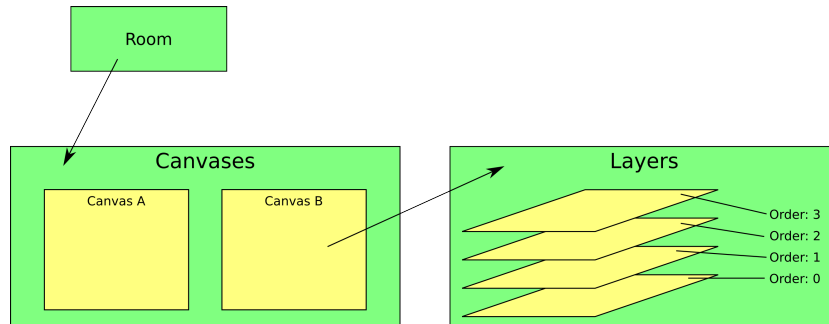


Figure 2.2: Room

Všechny plátna mají své rozměry (v pixelech), jméno a ID. Server se nestará o pořadí pláten, je plně na klientovy jakým způsobem je zobrazuje. Dále obsahují vrstvy a poskytují jejich pořadí.

Každá místnost může obsahovat 0 až  $2^{32}$  (4 bytové ID) pláten a každé plátno se skládá z 0 až  $2^{32}$  vrstev.

Plátno má své ID, pořadí a vlastní obsah (obrázek s 32 bitovou hloubkou – 1 B je vyhrazen jako alpha channel).

Pořadí vrstev začíná číslem 0 a končí  $p - 1$ , kde  $p$  je celkový počet vrstev. Při vykreslování se nejvíce dospod (takže jako první) kreslí vrstva s pořadím 0. Takže čím větší má vrstva pořadí, tím více navrchu je vykreslena. Viz obrázek 2.2.

## 2.2 Chat

## 2.3 Another services

# **Part IV**

## **Connection**

Vlastní protokol je rozdělen do dvou vzájemně nezávislých vrstev. To umožňuje oddělení funkcionality aplikačního síťového protokolu, přes který se posílají data, a vlastní datové struktury. Na obrázku 2.3 jsou ilustrovány vrstvy protokolu.

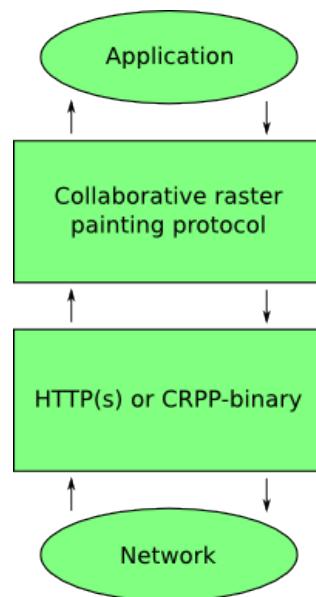


Figure 2.3: Protocol layers diagram

Vrchní vrstva předává spodní vrstvě monolitické bloky dat, které spodní vrstva dopraví vrchní vrstvě na druhé straně spojení. Vrchní vrstva je popsána v části VI a spodní v části V;

## 3 Data types

Pokud není uvedeno jinak, tak všechny celočíselné hodnoty jsou bezeznaménkové.

### 3.1 Unsigned integers

Každý bit s hodnotou 1 reprezentuje číslo  $2^p$ . Kde  $p$  je pozice bitu (poslední bit – v textu nejpravější – posloupnosti má pozici 0). Bity s hodnotou 0 jsou rovny 0. Rovnice ?? udává hodnotu čísla.

$$\sum_{p=0}^{n-1} b_p 2^p \quad (3.1)$$

kde  $p$  je pozice bitu (poslední má hodnotu 0, pak roste směrem k předcházejícím),  $n$  je celkový počet bitů reprezentujících číslo,  $b_p$  je hodnota bitu (0 nebo 1) na pozici  $p$ .

Například posloupnost 00001010 má hodnotu 10 ( $2^1 + 2^3$ ).

### 3.2 Signed integers

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

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

Hodnotu proměnné  $b$  určuje vzorec 3.1 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  $-5$  ( $a = -2^3 \cdot 1 = -8$ ,  $b = 2^1 + 2^0 = 3$ ,  $a + b = -8 + 3 = -5$ ).

### 3.3 Floating point numbers

Reálná čísla jsou reprezentována podle standardu IEEE 754 ve 32 bitech.

### 3.4 Logical value (booleans)

Logické hodnoty (boolean) jsou reprezentovány jedním bytem, kde samé nuly (00000000) reprezentují hodnotu *false* a hodnota jedna (00000001) reprezentuje hodnotu *true*. Ostatní hodnoty jsou považovány za nevalidní.

## 3.5 Strings

Veškerý text (řetězce znaků) je kódován jako UTF-8. V některých případech jako posloupnost ASCII znaků, to je ovšem vždy uvedeno. Všude je rozlišováno mezi malými a velkými písmeny (case sensitive).



# Part V

## Down layer

Spodní vrstva se stará o spojení, jeho udržování, obnovování a fyzickou podobu přenášení dat. Je možné použít jeden ze dvou aplikačních protokolů (ve spodní vrstvě), a to HTTP(s) (viz 5), nebo CRPP-binary (viz 4).

Spodní vrstva komunikuje s vrchní tím, že jí předává respektive dostává monolitické bloky dat. Tyto bloky jsou stejné na obou stranách spojení, viz diagram 3.1. 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 vrstvami musí komunikace probíhat tak, jako by k žádnému prohození pořadí nedošlo.

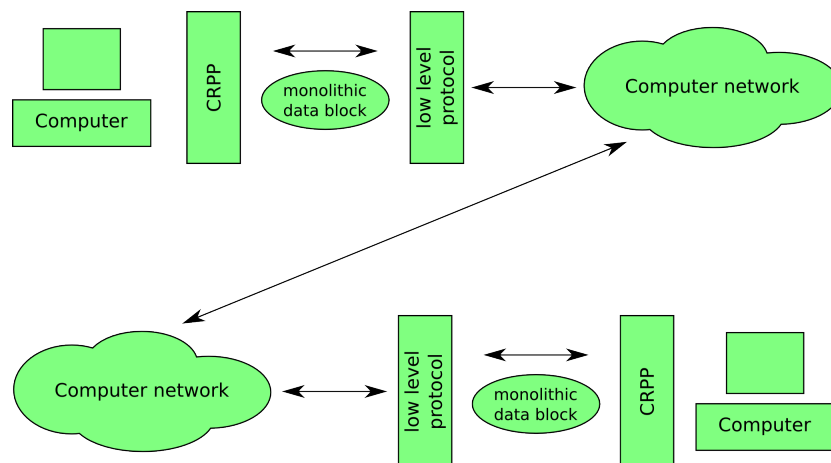


Figure 3.1: Monolithic data blocks

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

## 4 CRPP-binary

CRPP-binary je jednoduchý protokol, který umožňuje 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 4.1.

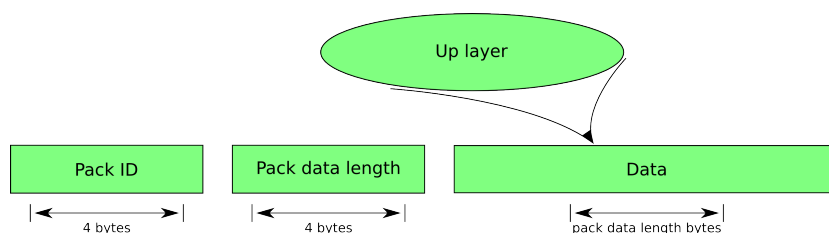


Figure 4.1: CRPP-binary pack

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. ID je uvedeno jako bezznaménkové celé číslo, které je popsáno v části 3.1.

Další čtyři byty (tedy pátý až osmý v pořadí) udávají celkovou délku přenášených dat bez hlavičky. Číslo je udáno opět podle specifikace v části 3.1.

Zbytek zprávy jsou vlastní data, která zpracovává vrchní vrstva.

## 5 HTTP(s)

# Part VI

## Up layer

Vrchní vrstva přenášená data (ta která si vyměňuje 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 6, použité datový typy v části 3.

## 6 Collab message

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

### 6.1 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 6.2.

Obsah message je také ilustrován obrázkem 6.1.

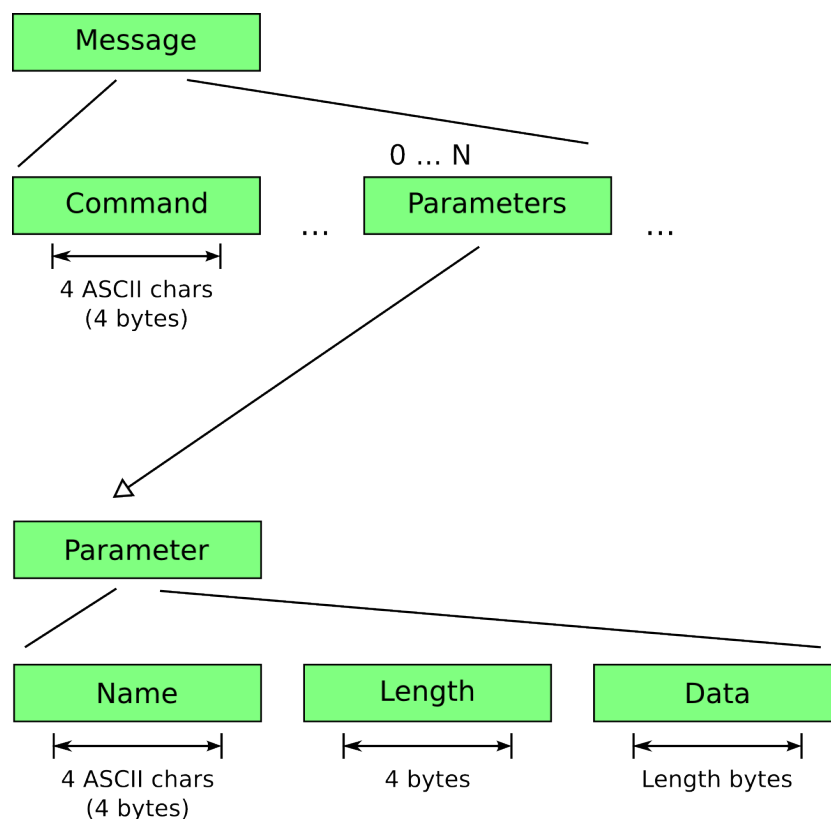


Figure 6.1: Message structure diagram

## 6.2 Parameter

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

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

První část (parameter 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 3.

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



# **Part VII**

## **Commands**

## 7 Commands

Tato kapitola popisuje jednotlivé příkazy, které mohou být jako message posílány, jejich strukturu, způsob zpracování a smysl.

Příkazy se dělí na příchozí (posílá je server na klienty), odchozí (klienti je posílají na server) a obousměrné (posílají se na server i na klienta).

Seznam příkazů:

- PANT – paint
- LORD – layers order
- ADDL – add layer
- LNAM – set layer name

### 7.1 Příchozí

### 7.2 Odchozí

#### 7.2.1 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 celočíselné bezznaménkové čtyřbytové číslo s ID plátna, do kterého má být vrstva přidána.

## 7.3 Obousměrné

### 7.3.1 Paint

Tento příkaz informuje o aktualizaci kreslicího plátna. Obsahuje tyto bloky:

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

#### Server action

Server zakreslí update na příslušné místo a poté rozešle the same update (paint) to all connected users.

#### Blocks

**Update type** Update type je typ updatu a obsahuje čtyřbytové neznaménkové celočíselné číslo udávající typ updatu. Může nabívat hodnot 0 a 1, kde 0 reprezentuje přidávací update a 1 mazací.

V případě přidávacího updatu se standartním (source over) algoritmem update nakreslí přes obrázek vrstvy.

V případě odebíracího updatu se jen mění velikost alpha kanálu jednotlivých pixelů updatované vrstvy. Výsledná alpha každého pixelu  $D_a$  se spočítá jako  $D_a = S_a \cdot P_a$  kde  $S_a$  značí alphu pixelu v updatovacím obrázku a  $P_a$  značí alphu v v obrázku vrstvy před updatem. Všechny alphy ve výpočtu nabývají hodnot 0 až 1, takže je před výpočtem potřeba je převést ze standartní podoby 0 až 255 na tyto dělením číslem 255 a po výpočtu převést zpátky násobením.

**Update ID** Obsahuje čtyřbytové neznaménkové celé číslo, které identifikuje tento update.

**Layer ID** Obsahuje čtyřbytové neznaménkové celé číslo, které identifikuje updatovanou vrstvu.

**Canvas ID** Obsahuje čtyřbytové neznaménkové celé číslo, které identifikuje updatované kreslicí plátno (je tak možné kreslit na několik pláten).

**X coordinate** Obsahuje čtyřbytové neznaménkové celé číslo, které udává X souřadnici levého horního rohu updatu (kam se aplikuje updatovací obrázek).

**Y coordinate** Obsahuje čtyřbytové neznaménkové celé číslo, které udává Y souřadnici levého horního rohu updatu (kam se aplikuje updatovací obrázek).

**Update image data** Obsahuje binární data PNG obrázku typu RGBA (32 bites). Viz <http://www.w3.org/Graphics/PNG/>.