
Protocole de communication Client-Serveur

Spécification

Les applications FriendPaint tourneront sur les ports suivants, par ordre de préférence : 63360 à 63369.

Ce protocole est basé sur les paquets SFML 2.2, qui fonctionnent comme suit :

Remplissage : packet << foo << bar;

Extraction : packet >> foo >> bar;

Avec foo et bar étant de n'importe quel type, y compris des objets de nos classes.

Ainsi sera le contenu minimal des paquets :

{<indicator>, [<payload>]}

indicator> est un entier sur 32 bits, qui suffira à l'application pour extraire correctement <payload>.

Les valeurs possibles de <indicator> seront indiquées par des constantes.

Protocole

User est un utilisateur.

Serv est le socket du serveur qui communique avec User.

Premier contact avec le serveur

User -> {FP_INIT, <user_name>, <version>}

Si <user_name> est trop long ou <version> non supportée :

Serv -> {FP_DENIED, <max_pseudonym_length>, <supported_version>}

Sinon :

Serv -> {FP_OK, <infos>}

Où <infos> est une structure contenant toutes les données que le client a besoin de connaître.

- **Cas 1 : L'utilisateur veut créer un salon.**

User -> {FP_CREATE_LOBBY, <passwd>, <title>, <size> <description>, <slots>, <ask_before_accept>, <layers>}

Si Serv ne peut pas:

Serv -> {FP_DENIED}

Sinon:

Serv -> {FP_OK, <user_color>}

- **Cas 2 : L'utilisateur veut rejoindre un salon.**

User -> {FP_JOIN_LOBBY, <lobby_id>, <passwd>}

Si <passwd> est incorrect

Serv -> {FP_DENIED}

Sinon :

Serv envoie le canevas dans son état actuel.

Serv -> {FP_OK, <user_color>, <layers>}

Communication au sein du salon

NB : Les utilisateurs sont identifiés par leur couleur.

On l'envoie à chaque message pour que le serveur ET les autres clients sachent qui est l'originaire (ou la cible) du message.

Généralement, les messages comportant <user_color> concernent tout le monde, et ceux comportant ne concernent que l'utilisateur ciblé.

- **Communication par tchat**

User -> {FP_SAY, <text>}

- **Gestion des utilisateurs**

Activer les privilèges administrateur :

User -> {FP_GRANT_ADMIN, <controlled_user_color>}

Donner le statut d'hôte :

User -> {FP_GRANT_HOST, <controlled_user_color>}

S'expulser :

User -> {FP_EXIT, <controlled_user_color>}

Verrou de création de calque :

User -> {FP_LOCK_CREATE_LAYER, <controlled_user_color>}

User -> {FP_UNLOCK_CREATE_LAYER, <controlled_user_color>}

- **Signaux du moteur de peinture**

- **Modifier l'outil :**

User -> {FP_SET_TOOL_TYPE, <user_color>, <type>}

User -> {FP_SET_TOOL_COLOR, <user_color>, <color>}

User -> {FP_SET_TOOL_DIAMETER, <user_color>, <diameter>}

- **Appliquer l'outil :**

User -> {FP_BEGIN_TOOL_DRAG, <user_color>, <x>, <y>}

User -> {FP_STEP_TOOL_DRAG, <user_color>, <x>, <y>}

User -> {FP_END_TOOL_DRAG, <user_color>, <x>, <y>}

- **Créer un nouveau calque :**

User -> {FP_CREATE_LAYER, <user_color>, <layer_name>}

- **Sélectionner un calque :**

User -> {FP_SELECT_LAYER, <user_color>, <layer_index>}

- **Renommer un calque :**

User -> {FP_RENAME_LAYER, <user_color>, <layer_index>, <name>}

- **Supprimer un calque :**

User -> {FP_DELETE_LAYER, <user_color>, <layer_index>}

- **Déplacer un calque dans la pile (haut ou bas, de <delta> crans) :**

User -> {FP_MOVE_LAYER, <user_color>, <layer_index>, <delta>}

- **Fusionner vers le bas un calque :**

User -> {FP_MERGE_DOWN_LAYER, <user_color>, <layer_index>}

- **Modifier verrous d'un calque :**

User -> {FP_LOCK_LAYER, <user_color>, <layer_index>, <locks>}

Fermeture

- **Fermeture normale**

User -> {FP_EXIT, <controlled_user_color>}

- **Fermeture anormale**

Serv peut détecter lorsque l'application de l'utilisateur ne s'est pas terminée proprement et entreprendre les actions nécessaires.