
RazorSharp Technologies

Hockedu
Protocole de communication

Version 1.1

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

Historique des révisions

Date	Version	Description	Auteur
2013-02-06	1.0	Ébauche initiale	Mathieu Parent
2013-02-08	1.1	Révisions	Mathieu M-Gosselin

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

Table des matières

1.	Introduction	4
2.	Communication entre les éléments du réseau	5
2.1	Serveur maître	5
2.1.1	Communication avec la base de données	5
2.1.2	Communication avec le serveur de jeu	5
2.1.3	Communication avec le client lourd	5
2.2	Serveur web	6
2.2.1	Communication avec la base de données	6
2.2.2	Communication avec le client léger	6
2.2.3	Communication avec le client lourd	6
2.2.4	Communication avec l'utilisateur	6
2.3	Serveur de jeu	7
2.3.1	Communication avec le serveur maître	7
2.3.2	Communication avec le client lourd	7
3.	Description des paquets	8
3.1	Structure de base des paquets	8
3.2	Paquets du serveur maître	8
3.2.1	Accomplissements (« achievements »)	8
3.2.2	Événements	8
3.2.3	Requête de connexion	8
3.2.4	Réponse à une requête de connexion	8
3.2.5	Envoi des parties en attente d'un joueur	8
3.2.6	Envoi des terrains	9
3.2.7	Enregistrement d'une partie	9
3.2.8	Résultats de la partie	9
3.2.9	Message de clavardage	9
3.2.10	Statut utilisateur	9
3.3	Paquets du serveur de jeu	10
3.3.1	Maillet	10
3.3.2	Rondelle	10
3.3.3	Rejoindre une partie	10
3.3.4	Paquets déjà définis pour le serveur maître	10
3.4	Paquets du client lourd	11
3.5	Serveur web	11
3.6	Client léger	11

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

Protocole de communication

1. Introduction

Le présent document vise à décrire la façon dont chacun des éléments du réseau communiquent ensemble pour ensuite décrire les différentes structures de paquets qui seront nécessaires au bon fonctionnement de ces communications.

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

2. Communication entre les éléments du réseau

2.1 Serveur maître

2.1.1 Communication avec la base de données

Le serveur maître interroge la base de données afin d'obtenir:

- La validité de la combinaison du nom d'utilisateur et du mot de passe fourni par chaque utilisateur (pour la connexion)
- La liste des terrains publics
- La liste des terrains associés à un utilisateur

De plus, il enregistre les données suivantes dans la base de données :

- Les accomplissements d'un utilisateur
- Les statistiques d'un utilisateur
- Le classement des utilisateurs

2.1.2 Communication avec le serveur de jeu

Le serveur maître obtient les informations suivantes du serveur de jeu :

- Vainqueur et score d'une partie
- Informations pour l'enregistrement du serveur

De plus, il lui envoie les données suivantes :

- Liste des terrains accessibles à un utilisateur

2.1.3 Communication avec le client lourd

Le serveur maître reçoit les informations suivantes du client lourd :

- Nom d'utilisateur et mot de passe haché pour la connexion
- Informations sur les accomplissements (« *achievements* »)
- Requête pour *matchmaking*.

De plus, il lui envoie les données suivantes :

- Si la combinaison nom d'utilisateur/mot de passe reçue est valide ou non
- La liste des terrains publics
- La liste des terrains associés à un utilisateur
- La liste des serveurs disponibles

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

2.2 Serveur web

2.2.1 Communication avec la base de données

Le serveur web va interroger la base de données afin d'obtenir :

- Le classement des joueurs
- Validité de la combinaison du nom d'utilisateur et du mot de passe de chaque utilisateur (pour la connexion)
- Les accomplissements (« *achievements* ») d'un utilisateur
- La liste des terrains publics
- La liste des terrains associés à un utilisateur
- Les statistiques d'un utilisateur

De plus, il communique avec la base de données pour :

- Changer le mot de passe d'un utilisateur
- Changer le niveau d'accès d'un terrain (public/privé) de l'utilisateur courant
- Supprimer un terrain de l'utilisateur courant

2.2.2 Communication avec le client léger

Le serveur web reçoit les informations suivantes du client léger :

- Les terrains créés par le client léger
- Nom d'utilisateur et mot de passe haché pour la connexion

De plus, il lui envoie les données suivantes :

- Si la combinaison nom d'utilisateur/mot de passe est valide ou non
- La liste des terrains publics
- La liste des terrains associés à un utilisateur

2.2.3 Communication avec le client lourd

La communication entre le serveur web et le client lourd se limite au téléchargement et au téléversement (« *upload* ») de terrains.

2.2.4 Communication avec l'utilisateur

Le serveur web supporte les actions suivantes de l'utilisateur :

- Se connecter
- Supprimer un terrain lui appartenant
- Modifier le niveau de visibilité d'un terrain lui appartenant
- Modifier son mot de passe
- Consulter ses statistiques
- Consulter le classement
- Modifier son nom d'utilisateur

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

2.3 Serveur de jeu

2.3.1 Communication avec le serveur maître

Voir la section 2.1.2 pour la description de la communication entre le serveur de jeu et le serveur maître.

2.3.2 Communication avec le client lourd

Le serveur de jeu obtient les données suivantes du client lourd :

- Déplacements du maillet

De plus, il lui envoie les informations suivantes :

- Déplacements de la rondelle
- Si un but a été marqué
- Début et fin de partie
- Temps absolu de la partie

Bien que la physique soit calculée sur le serveur ET sur le client, les résultats envoyés par le serveur sont prioritaires en tout temps.

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

3. Description des paquets

3.1 Structure de base des paquets

Tous les paquets possèdent la même structure de base. Cela est requis afin de pouvoir ignorer les paquets qui arrivent d'une application autre que la nôtre. Cette structure de base possède les attributs suivants :

- Une chaîne de caractères (« LindseyStirling ») commence chaque paquet. Cette chaîne est constante, ce qui signifie qu'elle représente l'identifiant de nos paquets.
- Une chaîne de 20 caractères qui représente le type du paquet. Elle est utilisée pour pouvoir effectuer les actions adéquates pour chaque paquet.
- Un nombre entier contenant un numéro de séquence qui sera utilisé pour détecter la perte de paquets UDP.
- Un entier contenant la taille totale du paquet.

Ainsi donc, toutes les données des paquets décrits ci-dessous vont implicitement suivre la forme décrite ci-dessus.

3.2 Paquets du serveur maître

3.2.1 Accomplissements (« achievements »)

Le paquet qui envoie les informations pour les accomplissements contient :

- Un entier qui définit le type de la réalisation
- Une chaîne de caractères qui contient le message à associer à la réalisation, précédée de sa taille.
- L'identifiant du joueur auquel la réalisation est rattachée.

3.2.2 Événements

Ce paquet contient tout ce qui a trait à des événements spécifiques et contient :

- Un entier qui définit le type de l'événement
- Une chaîne de caractères contenant le message à associer à l'événement, précédée de sa taille

3.2.3 Requête de connexion

Ce paquet informe le serveur maître d'une tentative de connexion et contient :

- Une chaîne de caractères contenant le nom de l'utilisateur, précédée de sa taille
- Une chaîne de caractères contenant le mot de passe haché, précédée de sa taille

3.2.4 Réponse à une requête de connexion

Ce paquet est émis en réponse à une requête de connexion et contient :

- Un caractère indiquant si la connexion a été acceptée
- Un entier représentant l'identifiant unique associé au nom d'utilisateur envoyé si la connexion s'est effectuée avec succès

3.2.5 Envoi des parties en attente d'un joueur

Le paquet qui envoie la liste des parties disponibles contient, pour chaque partie:

- Une chaîne de caractères contenant le nom de la partie, précédée de sa taille
- Une chaîne de caractères contenant l'adresse IP du serveur hébergeant la partie, précédée de sa taille

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

- Un entier représentant l'identifiant unique de la partie sur le serveur
- Une chaîne de caractères contenant le nom du joueur qui a créé la partie, précédée de sa taille
- Une chaîne de caractères contenant le nom du terrain associé à la partie, précédée de sa taille

De plus, le paquet commence avec le nombre de parties non-commencées enregistrées sur le serveur maître.

3.2.6 *Envoi des terrains*

Le paquet qui envoie la liste des terrains contient, pour chaque terrain :

- Une chaîne de caractères contenant le nom du terrain, précédée de sa taille

De plus, le paquet commence avec le nombre de terrains contenus dans la liste envoyée.

3.2.7 *Enregistrement d'une partie*

Le paquet servant à enregistrer une partie contient :

- Une chaîne de caractères contenant le nom de la partie, précédée de sa taille
- Une chaîne de caractères contenant l'adresse IP du serveur hébergeant la partie, précédée de sa taille
- Une chaîne de caractères contenant le nom du joueur qui a créé la partie, précédée de sa taille
- Une chaîne de caractères contenant le nom du terrain associé à la partie, précédée de sa taille

3.2.8 *Résultats de la partie*

Le paquet envoyant le résultat de la partie contient :

- Une chaîne de caractères contenant le nom d'un des joueurs, précédée de sa taille
- Un entier représentant le pointage du joueur
- Une chaîne de caractères contenant le nom de l'autre joueur, précédée de sa taille
- Un entier représentant le pointage du joueur

3.2.9 *Message de clavardage*

Le paquet pour transmettre un message de clavardage contient :

- Une chaîne de caractères qui contient le message à envoyer, précédée de sa taille
- Un booléen qui indique si le paquet s'adresse à un groupe ou non
- Une chaîne de caractères qui contient le nom du groupe, ou le nom du joueur à qui le message est destiné (dépend de la valeur du booléen), précédée de sa taille.
- Une chaîne de caractères qui contient le nom du joueur qui a envoyé, précédée de sa taille.

3.2.10 *Statut utilisateur*

Ce paquet contient :

- Une chaîne de caractères contenant un nom d'utilisateur, précédée de sa taille
- Un entier indiquant son statut de connexion

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

3.3 Paquets du serveur de jeu

3.3.1 *Maillet*

Le paquet permettant d'envoyer les déplacements du maillet contient :

- Un entier qui représente l'identifiant pour déterminer à quel joueur le maillet appartient
- Deux entiers qui représentent la position (x,y) actuelle du maillet
- Deux entiers qui représentent la position (x,y) désirée du maillet

3.3.2 *Rondelle*

Le paquet permettant d'envoyer les déplacements de la rondelle contient:

- Deux entiers représentant la position (x,y) de la rondelle
- Une valeur à virgule flottante représentant la vitesse angulaire de la rondelle
- Une valeur à virgule flottante représentant la vélocité de la rondelle

3.3.3 *Rejoindre une partie*

Le paquet permettant de rejoindre une partie contient :

- Une chaîne de caractères contenant le nom du joueur qui a rejoint la partie, précédée de sa taille
- Une chaîne de caractères contenant l'adresse IP du joueur qui a rejoint la partie, précédée de sa taille

3.3.4 *Paquets déjà définis pour le serveur maître*

Les paquets suivant sont réutilisés dans le serveur de jeu :

- Paquet de type événement (3.2.2)
- Paquet de type enregistrement d'une partie (3.2.7)
- Paquet de type résultat de la partie (3.2.8)
- Paquet de type message de clavardage (3.2.9)
- Paquet de type statut d'utilisateur (3.2.10)

Hockedu	Version: 1.1
Protocole de communication	Date: 2013-02-08

3.4 Paquets du client lourd

Le client lourd se sert de paquets qui ont déjà été définis plus haut dans le document, soit :

- Paquet de type accomplissement (3.2.1)
- Paquet de type événement (3.2.2)
- Paquet de type requête de connexion (3.2.3)
- Paquet de type réponse à une requête de connexion (3.2.4)
- Paquet de type envoi des parties en attente d'un joueur (3.2.5)
- Paquet de type envoi d'un terrain (3.2.6)
- Paquet de type message de clavardage (3.2.9)
- Paquet de type statut d'utilisateur (3.2.10)
- Paquet de type maillet (3.3.1)
- Paquet de type rondelle (3.3.2)
- Paquet de type rejoindre une partie (3.3.3)

3.5 Serveur web

Le serveur web reçoit des requêtes HTTP qui permettent à l'utilisateur d'effectuer les actions suivantes :

- Se connecter
- Supprimer un terrain lui appartenant
- Modifier le niveau de visibilité d'un terrain lui appartenant
- Modifier son mot de passe
- Consulter ses statistiques
- Consulter le classement
- Créer un nouveau nom d'utilisateur
- Télécharger des terrains
- Téléverser des terrains

3.6 Client léger

Le client léger envoie des requêtes HTTP au serveur web afin de se connecter et de télécharger et de téléverser des terrains.