

# PuckNet

Protocole — Document de travail — version 3



# Sommaire

<b>Sommaire</b>	<b>2</b>
<b>Informations générales</b>	<b>3</b>
<b>Exemple de communication</b>	<b>4</b>
<b>Commandes</b>	<b>5</b>
Serveur	5
Connexion	5
Jeu	5
Client	6
Recherche d'un serveur	6
Jeu	6
<b>Propriétés</b>	<b>6</b>
Envoyées par le client	7
Envoyées par le serveur	7

# Informations générales

Le serveur écoute sur le port 7825 (puck sur un téléphone)

le plateau est divisé en case. Chaque case a une largeur de  $60/4 = 15$  unités.

Un Pacman se déplace de quatre cases par seconde (soit  $15*4=60$  unités). Son déplacement est continu (il ne peut pas s'arrêter à moins de rencontrer un mur).

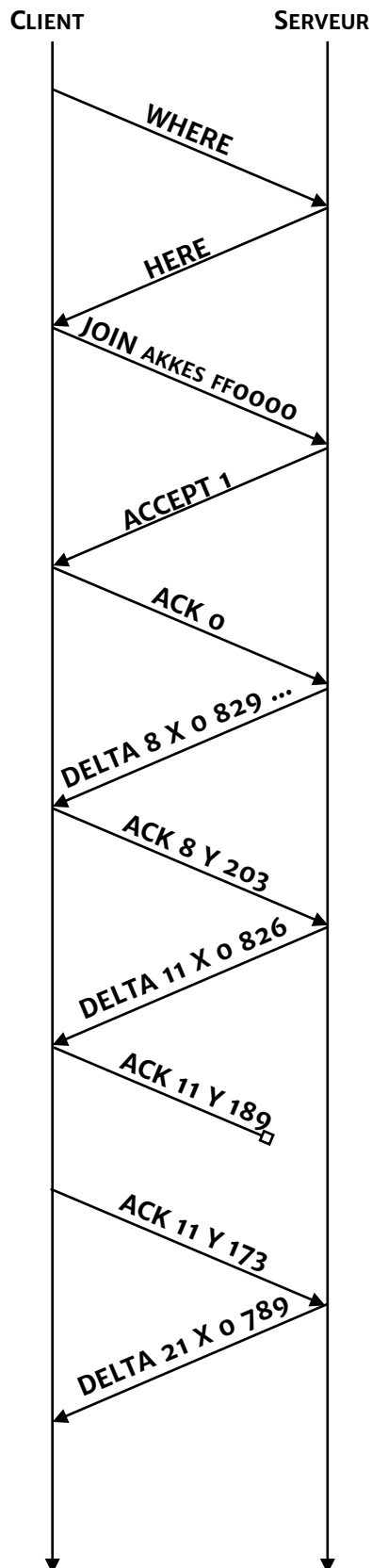
Chaque Dot mangé rapporte 10 points

Manger une Gum transforme en SuperPacman qui peut manger les autres pacmans pendant 10 secondes. Manger un autre pacman rapporte 200 points pour le premier mangé d'une transformation puis le double du précédent pour tout les suivants. Chaque Gum mangée rapporte 50 points.

Les éléments grisés ne sont pas nécessaires pour l'utilisation du protocole.

Bon code! (Ou netcat si vous voulez jouer en slow-motion...)

# Exemple de communication



Le client recherche un serveur en broadcastant un WHERE. Le serveur lui répond avec un HERE

Le client indique son nom et sa couleur. Le serveur lui indique s'il est accepté ou refusé et le cas échéant son ID de joueur.

Le client indique qu'il a bien reçu l'information et déclare qu'il connaît l'état 0 (aucune information connue). Le serveur lui répond avec l'état actuel (8).

Le client indique qu'il a bien reçu l'état 8 et donne ses informations qui ont changé. Le serveur lui répond avec le delta entre l'état 8 (ack) et l'état 11 (état actuel).

Le client ACK avec son nouvel état mais le paquet se perd.

Le client renvoi l'ack et son état mit à jour. Le serveur lui répond avec un nouveau DELTA

# Commandes

## Serveur

### Connexion

HERE

**En réponse à:** WHERE

**Réponse:** JOIN ou DENY

ACCEPT player\_nb

Informe de l'acceptation du client

player\_nb: l'ID du client lors du jeu

**En réponse à:** JOIN

**Réponse:** ACK

DENY

Informe du refus du client

**En réponse à:** JOIN

### Jeu

DELTA state Propriétés

Donne les différences entre le dernier état reçu (ack'ed\_state) et l'état actuel.

**En réponse à:** ACK

**Réponse:** ACK

## Client

### Recherche d'un serveur

WHERE

**Réponse:** HERE

JOIN name color

Demande à rejoindre une partie:

name: le nom du joueur

color: la couleur du joueur

**Réponse:** ACCEPT ou DENY

### Jeu

ACK ack'ed\_state Propriétés

Informe le serveur de la bonne réception de l'état ack'ed\_state

**En réponse à:** JOIN et DELTA

**Réponse:** DELTA

RESET

Remet la totalité des Dots et les Gums sur le terrain

LEAVE

Indique au serveur que le joueur se déconnecte

## Propriétés

## Envoyées par le client

X position

La nouvelle position horizontale du joueur  
position est en unité

Y position

La nouvelle position verticale du joueur  
position est en unités

Dot dotID

Le joueur a mangé le Dot dotID

Gum gumID

Le joueur a mangé la Gum gumID

Eat playerID

Le joueur a mangé le joueur playerID

## Envoyées par le serveur

NewDot dotID

Le Dot dotID est revenu sur le terrain

NewGum gumID

La Gum gumID est revenue sur le terrain

NewPlayer playerID X Y name color

Un joueur s'est connecté. L'identifiant playerID lui a été attribué. Il se situe en position (X,Y). A le nom name et la couleur color.

X et Y sont en unités

X playerID position

le joueur playerID est en position horizontale position

Y playerID position

le joueur playerID est en position verticale position

Dot dotID

Le Dot dotID a été mangé.

Gum gumID

La Gum gumID a été mangée.

Super playerID

Le joueur playerID est un SuperPacman.

Normal playerID

Le joueur playerID est redevenu un pacman normal.

Leave playerID

Le joueur playerID est s'est déconnecté.

Score playerID score

Le joueur playerID est a comme score score;