

# Protocoles des services Internet

## Protocole Client/Serveur (TCP)

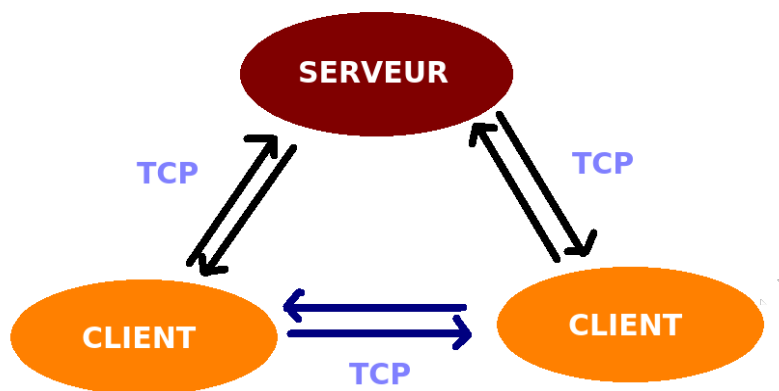
### I – Participants

---

Karim YAHIAOUI - Kamilia AHMED ALI  
Youness BOUSBIH - Mohammad NAZIR  
Florian LALLIER - Phuonsavath BOUTAVETH  
Valens IRAGUHA - Manoela Tsizehena

### II – Architecture

---



### III – Spécification

---

#### A ) Messages

Type de message : **chaîne de caractères (java.lang.String)** suivi d'un **saut de ligne (\n)**.

Composition des messages :

CODI AdresseClient PortClient

DISP AdresseClient PortClient

QUIT AdresseClient PortClient

OCCU AdresseClient1 PortClient1 AdresseClient2 PortClient2

WIN0 AdresseClient1 PortClient1 AdresseClient2 PortClient2

WIN1 AdresseClient1 PortClient1 AdresseClient2 PortClient2

WIN2 AdresseClient1 PortClient1 AdresseClient2 PortClient2

## B ) COMMUNICATIONS

- 1) Le serveur doit être démarré en premier. Il attend des connexions sur le **port 1027**.
- 2) Chaque client qui se connecte au serveur lui envoie un message CODI.
- 3) À chaque message CODI ou DISP reçu, le serveur envoie :
  - À tous les clients un message DISP avec les informations du nouveau client.  
*Précision : un seul message.*
  - Au nouveau client la liste de tous les clients disponibles.  
*Précision : autant de messages DISP que de clients connectés.*

---

### Pré-protocole de jeu

---

- 4) Si une session de jeu est acceptée durant le pré-protocole de jeu, le client initiateur de la demande doit envoyer un message OCCU au serveur contenant ses propres informations ainsi que les informations de son adversaire.
- 5) À chaque message OCCU reçu par le serveur, ce dernier renverra ce même message OCCU à l'ensemble des autres clients.

*Précision : autant de messages OCCU que de clients connectés.*

---

### Protocole de jeu

---

- 6) Après la partie, le client qui a initié la partie envoie le message résultat au serveur :
  - WIN0 indique un match nul.
  - WIN1 indique que Client1 a gagné.
  - WIN2 indique que Client2 a gagné.
- 7) Une fois la partie terminée, le client peut envoyer un message DISP au serveur (voir étape 3) ou envoyer un message QUIT. À chaque message QUIT reçu, le serveur renverra ce même message QUIT à l'ensemble des clients.

*Précision : autant de messages QUIT que de clients connectés.*

# Protocole Client/Client (TCP)

## A ) Jeu

**Le jeu choisi est le morpion (tic-tac-toe).**

## B ) Messages

Type de message : **chaîne de caractères (java.lang.String)** suivi d'un **saut de ligne (\n)**.

Composition des messages :

GAME

OK

NO

COOR <entier[0-2]> <entier[0-2]>

Précision : le premier entier est l'indice des lignes, le deuxième l'indice des colonnes.

## I – Pré-protocole de jeu

---

- 1) Le Client1 souhaite jouer avec Client2, il lui envoie un message GAME.
- 2) Si le Client2 accepte, il envoie au Client1 un message OK.  
Si le Client2 refuse, il envoie au Client1 un message NO.

## II – Protocole de jeu

---

- 1) Par convention, le client qui a initié la demande commence la partie et envoie un message COOR.
- 2) Le Client2, à la réception d'un message COOR, renvoie un message COOR. Et ainsi de suite.
- 3) La partie se termine lorsque qu'un joueur a gagné ou bien que toutes des cases ont été remplies (match nul).