

SAÉ 3.01 C3 C6

Communications
client / serveur par sockets
TCP/IP et/ou UDP

Sujet : PN

Rappel : la SAÉ du semestre 3 doit couvrir les 6 compétences et produire une évaluation dont la note finale est reportée dans les UE 1 à 6 du semestre selon la compétence évaluée.

Cette SAÉ a pour point central la compétence 3 : Administrer des systèmes informatiques communicants complexes.

Elle donnera lieu à 2 notes appelée C3-PN et C6-PN dans l'unique SAÉ S3.01 du semestre 3.

Critères de notation :

- C3 : manipulation des sockets en tcp/ip et/ou udp
 - AC23.01 : Concevoir et développer des applications communicantes
- C6 : travailler dans une équipe informatique
 - AC26.04 : Rendre compte de son activité professionnelle

Cette SAÉ permet de vous initier à la communication client-serveur et vous demande d'avoir recours aux sockets réseau pour établir des communications en tcp/ip et/ou udp.

PN signifie Pendu Net : vous allez donc développer un jeu du pendu en réseau !

Le choix de ce jeu très simple se justifie par le fait que vous allez surtout vous concentrer sur les communications avec plusieurs versions à développer, et pas sur le jeu en lui-même qui ne vous demandera que peu de travail.

Contraintes :

- tous les développements seront impérativement effectués en **langage C**
- vous pourrez utiliser indifféremment le système Linux, Windows ou macOS
- vous utiliserez les **sockets TCP/IP et/ou udp** pour les communications, sans bibliothèque supplémentaire de haut niveau

Quelles sont les productions de cette SAÉ ?

Il y a 5 versions du jeu à développer de la V0 à la V4, avec de plus en plus de fonctionnalités, toutes décrites dans les pages suivantes. La V4 n'est pas obligatoire pour les APP (moins d'heures à l'edt). Vous produirez pour chaque version un programme client et un programme serveur ou un programme unique permettant de se comporter en client ou en serveur.

Les livrables sont :

- un fichier pdf présentant les organigrammes de communication client/serveur de chaque version ainsi que la manière de lancer le client et le serveur (options de ligne de commande)
- une archive de votre code contenant les fichiers PN_client_Vx.c + PN_serveur_Vx.c ou alors PN_client_serveur_Vx.c (x la version de 0 à 4) + autres fichiers de code complémentaires

Comment se fait le travail ?

- Projet sur 10 jours : soutenance du résultat mercredi 17 décembre 13h
- travail en groupes de 4 mélangeant parcours A et parcours B
- groupe FI/APP possible
- en présentiel

Dépôt sur Moodle des livrables avant la soutenance finale :

- Présentation orale de 10mn maximum
- Pas de powerpoint mais présentation (organigramme + code + démo) de la version la plus élevée aboutie (V4 au mieux, ou sinon V3, ou V2, ...)

Énoncé de la SAE : Lire intégralement cet énoncé avant de commencer

Tout d'abord relire les règles du [jeu du pendu sur Wikipedia](#). Vous pourrez éventuellement choisir le nombre d'erreurs autorisées.

PN version 0 :

Voici la description succincte du rôle du client et du serveur.

Les explications n'indiquent volontairement pas le nombre ni le contenu des messages échangés afin que vous ayez tous des versions différentes !

Lisez les explications ci-dessous puis dessinez tout d'abord l'organigramme des communications avant de coder !!!

Sauvegardez cet organigramme dans un fichier pdf.

Le client incarne le joueur qui doit deviner le mot :

- se connecte au serveur
- attend le message "start x" du serveur qui indique que la partie peut commencer avec un mot comportant x lettres
- affiche le mot (x tirets pour commencer)
- permet à l'utilisateur de choisir une lettre, l'envoie au serveur
- attend le message du serveur qui indique si la lettre fait partie du mot ou non ou alors si il faut rejouer dans le cas où la lettre a déjà été choisie. Ce message doit aussi permettre d'indiquer si la partie est gagnée ou perdue.
- En fonction des cas, la partie se termine ou alors le client boucle sur une nouvelle lettre

Le serveur incarne le joueur qui fait deviner le mot :

- accepte la connexion d'un client
- envoie le message "start x" (x = nombre de lettres dans le mot)
- attend le message indiquant la lettre du client et vérifie par rapport au mot à deviner, met à jour les données pertinentes pour la gestion de la partie.
- répond au client pour lui indiquer ce qu'il doit faire ensuite
- remarque : une seule connexion est attendue pour ensuite jouer la partie qui, une fois terminée, permet au serveur de boucler en attente d'une nouvelle connexion car il ne quitte jamais, ... c'est un serveur
- il fait deviner toujours le même mot dans cette version car il n'y a pas d'interaction avec la console (en dehors d'informations de log dans la console ou dans un fichier)

PN version 1 (après avoir terminé et testé V0) :

Dans cette version, 2 joueurs vont se connecter pour trouver le même mot.
Attention à séparer vos fichiers sources de la V0 et de la V1 pour continuer !

Lisez les explications ci-dessous puis dessinez tout d'abord l'organigramme des communications avant de coder !!!

Sauvegardez cet organigramme dans un fichier pdf.

Explications succinctes (à vous d'en déduire les organigrammes de communication avant de coder et produire le pdf) :

- Le premier client se connecte et attend l'accusé de réception du serveur qui lui indique de patienter pour trouver le mot de x lettres
- Le deuxième client se connecte et attend l'accusé de réception du serveur qui lui indique de commencer pour trouver un mot de x lettres
- le serveur fait donc jouer en premier le 2^e client connecté puis alterne avec le 1^{er} client
- le premier des 2 clients a avoir trouver le mot à gagné
- le serveur ne gère qu'une partie à la fois

PN version 2 (après avoir terminé et testé V0) :

Dans cette version, les deux joueurs font une partie « classique » ensemble. Le serveur ne joue donc pas mais fait transiter les messages entre les 2 clients.

Lisez les explications ci-dessous puis dessinez tout d'abord l'organigramme des communications avant de coder !!!

Sauvegardez cet organigramme dans un fichier pdf.

Explications succinctes (à vous d'en déduire les organigrammes de communication avant de coder et produire le pdf) :

- Le premier client qui se connecte au serveur incarne le joueur qui fait deviner le mot
- Le deuxième client qui se connecte doit deviner le mot
- Le serveur gère les communications à tour de rôle entre les 2 joueurs mais sans intervenir dans la partie
- le serveur ne gère qu'une partie à la fois

PN version 3 (après avoir terminé et testé V2) :

Même principe que la version 2 mais le serveur doit gérer plusieurs parties donc il :

- est perpétuellement en attente de connexion pour de nouvelles parties
- gère les échanges des parties en cours

OBLIGATOIRE UNIQUEMENT POUR LES FI (car plus d'heures à l'edt) :**PN version 4 (après avoir terminé et testé V2) :**

Même principe que la version 2 mais le serveur doit mettre directement les 2 joueurs connectés en communication pour qu'ils échangent leurs messages sans passer par le serveur.

Ce mécanisme doit du coup permettre au serveur d'être perpétuellement en attente de nouvelles parties.