

**SAÉ 3.01 C3**

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

**Sujet : T3N**

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-Sockets et C6-Sockets 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.

T3N signifie Tic-Tac-Toe Net. Vous allez développer un jeu du morpion en réseau !

Le choix de ce jeu très simple se justifie par le fait que vous allez surtout vous concentrer sur les communications, et peu 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 **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.

Vous produirez un programme client et un programme serveur pour chaque version.

Par exemple pour la version V0 vous produirez les fichiers T3N\_client\_V0.c et T3N\_serveur\_V0.c

Les livrables pour chaque version sont :

- un fichier pdf présentant l'organigramme de communication client/serveur
- le fichier T3N\_client\_Vx.c (x la version de 0 à 4)
- le fichier T3N\_serveur\_Vx.c (x la version de 0 à 4)
- le code devra impérativement être commenté

Dépôt sur Moodle des livrables avant la soutenance finale le vendredi 10 janvier 2025 matin :

- 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, ...)

Comment se fait le travail ?

- Projet de 27h pendant une semaine (quasi à temps plein)
- travail en groupes de 3/4
- en présentiel

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

Tout d'abord relire les règles du Tic-tac-toe sur Wikipedia : <https://fr.wikipedia.org/wiki/Tic-tac-toe>  
Les versions V0, V1 et V2 doivent être faites impérativement dans l'ordre et il faut avoir fini une version avant de passer à la suivante.

### T3N version 0 :

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

Dans cette version, les règles du jeu ne sont pas du tout respectées et il n'y a pas de fin de partie mais les communications sont mises en place dans une boucle de jeu.

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 X :

- se connecte au serveur
- attend le message "start" du serveur qui indique que la partie peut commencer
- affiche la grille de jeu vide
- permet à l'utilisateur de choisir une case, l'envoie au serveur et affiche la grille mise-à-jour
- attend le message du serveur qui contient la case choisie par le serveur -> le client affiche la grille mise-à-jour puis boucle pour rejouer.

Le serveur incarne le joueur O :

- accepte la connexion d'un client
- envoie le message "start"
- attend le message indiquant la case jouée par le client et met à jour sa grille
- choisit une nouvelle case au hasard et l'envoie au client
- remarque : il boucle toujours en attente de connexion et ne quitte jamais ... c'est un serveur.

### T3N version 1 (après avoir terminé et testé V0) :

Dans cette version, les règles du jeu sont respectées, ce qui complique le code et les communications.

Attention à dupliquer vos fichiers sources V0 dans des fichiers 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.

Le client incarne toujours le joueur X, et, **en plus de la version 0** il :

- attend le message du serveur qui peut-être :
  - "continue" + la case choisie par le serveur, le client met à jour ses données et affiche la grille mise-à-jour et boucle pour rejouer.
  - "Owins" + une case choisie par le serveur, le client met à jour ses données et affiche la grille mise-à-jour -> Le serveur, donc le joueur O, gagne grâce à ce dernier coup, donc fin de la partie, le programme quitte
  - "Oend" + une case choisie par le serveur, le client met à jour ses données et affiche la grille mise-à-jour -> La grille est pleine suite à ce dernier coup, pas de gagnant, fin de la partie, le programme quitte
  - "Xend" -> fin de la partie (grille pleine suite au dernier coup joué par le client), le programme quitte
  - "Xwins" -> Le client, donc le joueur X gagne, fin de la partie, le programme quitte,

Le serveur incarne le joueur O, et, **en plus de la version 0** il :

- vérifie la fin du jeu lorsqu'il reçoit la case jouée par le client et dans ce cas :
  - envoie « Xwins » si X gagne ou « Xend » si la grille est pleine
- choisit une case à jouer (au hasard parmi les cases vides, ne faites pas d'IA ce n'est pas le moment ! Vous la ferez si vous avez terminé la V4 !) :
  - si grille pleine envoie "Oend"
  - si gagné envoie "Owins" + la case jouée
  - sinon envoie "continue" + la case jouée

### **T3N version 2 (après avoir terminé et testé V1) :**

Dans cette version, les deux joueurs sont des clients. Le serveur ne joue donc pas mais fait respecter les règles du jeu (c'est le juge ou l'arbitre).

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 X et le deuxième, le joueur O
- Le serveur gère la partie en communiquant avec les clients et en les faisant jouer à tour de rôle comme dans la V1 et avec les mêmes messages

### **T3N version 3 et 4 (après avoir terminé et testé V2) :**

Il n'y a pas d'ordre pour ces 2 dernières améliorations.

Produisez encore les organigrammes de communication et les pdf associés avant de coder

Faites celle que vous voulez en premier.

Si vous réussissez à faire les 2 alors appelez la dernière version 34 pour qu'on le sache facilement.

### **V3 :**

Comme pour la V2, les 2 premiers clients qui se connectent jouent.

Possibilité de connexion de clients supplémentaires qui observent la partie mais ne jouent pas.

### **V4 :**

Le serveur est capable de gérer plusieurs parties de Tic-tac-toe simultanées