

Documents et conception - Tic-Tac-Toe

Objectifs - Concevoir et développer une application console client-serveur en utilisant la méthode de communication de votre choix

- Votre application devra permettre de jouer au jeu de tic-tac-toe en réseau (2 joueurs).
- Vous devrez pouvoir jouer sur 2 ordinateurs différents (*application serveur sur un, le client sur un autre*).
- Vous devrez utiliser les mécanismes appropriés afin d'effectuer les communications entre le client et le serveur.
- Vous devrez permettre la saisie des informations afin d'établir correctement la connexion entre le client et le serveur

Date de remise (Projet final)

- À remettre au plus tard **Mercredi le 20 Septembre à 8h09**

Dates de remise intermédiaires (Livrables partiels)

- Plan de développement **Lundi 11 septembre à 10h00**
 - Analyse des besoins, calendrier, planification des tests
 - Diagramme de cas d'utilisation, de séquences, UML
- Maquette **Lundi 11 septembre à 10h00**
- Tests unitaires **Lundi 11 septembre à 10h00**

Remise

- L'ensemble des documents de conception et d'analyse
- Remettre sur LÉA le lien GIT de votre projet avec le nom des coéquipiers
- Il est **fortement suggéré d'utiliser GIT** (*adéquatement*) tout au long de votre développement

Directives générales

- Ce travail doit être réalisé en **équipe de 2 ou 3 personnes**
 - Votre équipe vous sera dévoilée dans quelques minutes
- La solution sera compilée sur un environnement standard VS2022
 - **Une solution qui ne compile pas = programme non fonctionnel = ZÉRO**

Critères d'évaluation

- Fonctionnement du programme (3)
- Utilisation adéquate des concepts de la POO, validations, gestion d'exceptions et tests (1)
- Modularité, structure et architecture du code (1)

Consignes / Contraintes

Vous devrez respecter les consignes et contraintes suivantes :

- Vous pouvez utiliser la méthode de communication désirée.
- Votre programme doit gérer adéquatement (*autant que possible*) les exceptions et effectuer les validations nécessaires au bon fonctionnement des traitements du programme.
 - Perte de connexion réseau
 - Client qui ferme son application sans demander/valider avec le serveur
 - *Pas besoin pour l'instant de gérer du côté client le serveur qui fermerait*
- Le serveur est en attente continue d'une connexion client tant qu'une partie n'a pas été débutée (*suite à la connexion d'un client*).
- Une fois la connexion établie entre un client et le serveur, les 2 joueurs devraient pouvoir effectuer autant de partie qu'ils le désirent et ce **sans rétablir une nouvelle connexion**.
- Votre programme devra permettre la connexion client/serveur peu importe où les programmes client et serveur sont exécutés. (*Paramétrer les informations « hôte » avant d'établir la connexion*).
- **Aucun des 2 joueurs (client ou serveur) ne peut prendre unilatéralement une décision.**
 - Vous devez donc valider que l'autre joueur n'a pas « triché »
- Lorsque le client décide de cesser de jouer (*après une partie*) le serveur se remet en mode attente.
- Il ne peut y avoir qu'une seule connexion client concourante sur le serveur.

Informations importantes en lien avec les consignes

- Ne pas confondre « Gestion d'erreur » et validations.
- Éviter autant que faire se peut, l'utilisation de break et continue.
 - ***Toute utilisation devra être justifiée***
- Minimiser et standardiser les informations transmises entre le client et le serveur.
 - Vous devez utiliser le principe de sérialisation vu la/les sessions précédentes

Livrable final du mini-projet #2

- Vous devez livrer tout le nécessaire afin d'avoir une application fonctionnelle client-serveur afin de jouer au tic-tac-toe.
- Après le projet, vous devrez présenter à l'enseignant votre projet et les différents documents de conception.
- Votre projet devra contenir un projet de test afin de tester différents éléments de votre projet
 - Vous ne devriez pas utiliser seulement 1 seul des projets individuels afin de compléter cette phase, il faudra donc adapter votre projet de test en conséquence