

Projet de programmation: TP

1 BAC Informatique à horaire décalé

Année académique 2014-2015

Ce projet est individuel et divisé en plusieurs itérations (aux dates du 11/04/2015, 02/05/2015 et 23/05/2015). Celles-ci vous permettront de progresser pas à pas et également de vous donner du feedback suite aux corrections. Il vous faudra donc remettre:

1. les programmes C de chacune des itérations ;
2. un rapport final lors de la dernière itération au format PDF expliquant l'architecture de votre code et de vos algorithmes;
3. le cas échéant, des fichiers de données initiales nécessaires au bon fonctionnement de votre programme.

Votre travail doit nous être remis en version électronique pour chaque itération avant 18h. Chaque itération sera cotée sur 10 (la première itération comptant pour 40% de la note finale, la seconde 20% et la dernière 40%, le rapport étant compris dans la note de celle-ci). Tout retard sera sanctionné de 1 point sur la note finale et ce par jour de retard entamé. Vous déposerez la version électronique sur Webcampus.

Ayez bien conscience dès le départ que la qualité du rapport final est tout aussi importante que la résolution du problème dans la note obtenue. Le résultat de la dernière itération sera donc coté de manière équivalente à la fois pour le programme mais aussi pour la clarté et la complétude de vos explications présentées dans votre rapport.

Votre programme sera testé en utilisant la version du compilateur C disponible au pool linux (au minimum gcc 4.7.2), votre code devra respecter le standard C90. Tous les programmes qui ne seront pas compatibles avec cette version seront considérés comme incorrects. Tout plagiat, copiage ou fraude impliquera une note de 0 sur 20.

Il vous est demandé d'utiliser les structures de données qui vous semblent les plus appropriées aux problématiques que vous rencontrerez et, si ceux-ci conviennent, d'utiliser des types de données abstraits. N'oubliez pas de motiver ces choix dans votre rapport. Détaillez également les structures de données utilisées lors du stockage sur disque.

Enfin, nous vous demandons de bien découper votre code en unités rassemblant les fonctionnalités liées à une même tâche. Cette découpe vous permettra également de rendre votre code plus lisible en l'éclatant dans plusieurs fichiers sources. Un bon découpage devrait limiter le nombre d'unités modifiées entre chaque itération.

Quelques conseils :

- Découpez votre programme en fonctions et/ou procédures lisibles. Expliquez votre découpe dans le rapport.
- Évitez de commenter à outrance votre code et utilisez l'indentation pour rendre votre code lisible.
- Prévoyez toujours des procédures de gestion d'erreur. La personne qui utilise votre programme pourrait le faire à mauvais escient. N'oubliez pas d'inclure une section dans votre rapport où vous précisez la gestion des erreurs d'utilisation de votre programme.
- Votre rapport doit contenir une analyse du problème qui a mené à votre algorithme et ensuite à votre implémentation. Pour mettre en évidence celle-ci, par exemple, liez votre analyse et le code en référant aux lignes de code. Aidez-vous d'exemples clairs et parlants pour illustrer votre propos. Présentez éventuellement des fils d'exécution graphiquement.

Énoncé

Nous vous demandons de réaliser un programme permettant à l'utilisateur de jouer une partie d'OXO. Dans un souci de simplicité, l'intelligence artificielle (IA) de l'ordinateur se limitera à un choix aléatoire du placement des lettres à jouer. Afin de vous permettre de vous concentrer sur les algorithmes et les structures de données, nous ne vous demandons pas d'interface utilisateur complexe. L'usage simple de la console suffit.

Règles du jeu

- Un damier carré de 10 cases de longueur.
- 2 joueurs: l'un prends les "X", l'autre les "O".
- Le but du jeu étant d'aligner trois X ou trois O (tous les sens sont bons même diagonalement).
- Chaque joueur doit, à tour de rôle, remplir l'une des cases soit avec un X, soit avec un O suivant ce qui a été décidé.
- Chaque combinaison établie sur le damier rapporte un point au joueur.

Voici un exemple illustratif dans lequel le joueur ayant les X a alors 6 points :

X O

X X X X

O X O

X X X

Fonctionnalités

A terme, le programme à réaliser comprendra:

- Un menu pour reprendre une partie en cours, jouer à deux ou jouer contre l'ordinateur. (En bonus, l'ordinateur pourra également jouer contre lui-même).
- En début de partie, les joueurs humains rentrent leur nom.
- Un écran permettant de jouer et reprenant le damier, les points des deux joueurs et le joueur en cours.
- A la fin de la partie, le programme affiche le gagnant, son score ainsi que la durée du jeu.
- Un système de sauvegarde automatique et de reprise d'une partie en cours ayant été interrompue.

Itérations

Afin de répartir au mieux la tâche à effectuer, le développement sera découpé en plusieurs itérations:

- 11/04/2015: Le programme permet de lancer une partie contre un joueur humain. Aucun menu de lancement n'est présent, on entre le nom des joueurs humain et on arrive directement sur l'écran permettant de jouer avec le damier, les points des deux joueurs et le joueur en cours. A la fin de la partie, le programme affiche le gagnant, son score ainsi que la durée du jeu.
- 02/05/2015: Le programme permet de lancer une partie contre l'ordinateur. Un menu se lance permettant de choisir entre jouer à deux ou jouer contre l'ordinateur. (Et ordinateur contre lui-même en bonus)
- 23/05/2015: Le programme sauve automatique la partie en cours (après chaque placement de lettre sur le damier) et le menu permet à présent de choisir de reprendre celle-ci. Ainsi que le rapport final comme détaillé auparavant.

Bon travail !